Auditory Image Understanding for the Visually Impaired Based on a Modular Computer Vision Sonification Model

DISSERTATION
zur Erlangung des Grades eines Doktors
der Ingenieurwissenschaften

vorgelegt von
Dipl. Inf. Michael Banf

eingereicht bei der Naturwissenschaftlich-Technischen Fakultät
der Universität Siegen
Siegen 2013

1. Gutachter: Prof. Dr. Volker Blanz
2. Gutachter: Dr. Karol Myszkowski
Datum der mündl. Prüfung: 16.10.2013

Zusammenfassung

Die vorliegende Arbeit beschreibt ein System das blinden Menschen einen direkt erfahrbaren Zugang zu Bildern mit Hilfe akustischer Signale anbietet. Der Benutzer exploriert ein Bild interaktiv auf einem berührungsempfindlichen Bildschirm und erhält eine akustische Rückmeldung über den Bildinhalt an der jeweiligen Fingerposition. Die Gestaltung eines solchen Systems beinhaltet zwei größere Herausforderungen: Welche ist die relevante Bildinformation, und wie kann möglichst viel Information in einem Audiosignal untergebracht werden. Wir behandeln diese Probleme basierend auf einem modularen Computer Vision Sonifikations Modell, welches wir als grundlegendes Gerüst für die Aufnahme, Exploration und Sonifikation von visueller Information zur Unterstützung blinder Menschen vorstellen. Es werden einige Ansätze vorgestellt, welche hierzu die Information auf verschiedenen Abstraktionsebenen kombinieren. So z.B. sehr grundlegende Information wie Farbe, Kanten und Rauigkeit und komplexere Information welche durch die Verwendung von Machine Learning Algorithmen gewonnen werden kann. Diese Machine Learning Algorithmen behandeln sowohl das Erkennen von Objekten als auch die Klassifikation von Bildregionen in "künstlich" und "natürlich", basierend auf einem neu entwickelten Typs eines probabilistischen graphischen Modells. Wir zeigen, dass dieser Mehr-Ebenen Ansatz dem Benutzer direkten Zugang zum Wesen und Position von Objekten und Strukturen im Bild ermöglicht und gleichzeitig das Potential neuester Entwicklungen im Bereich Computer Vision und Machine Learning ausnutzt. Während der Exploration kann der Benutzer erkannte "künstliche" Strukturen oder bestimmte natürliche Regionen als Referenzpunkte verwenden um andere natürliche Regionen mit Hilfe deren individueller Position, Farbe und Texturen zu klassifizieren. Wir werden zeigen, dass geburtsblinde Teilnehmer diese Strategie erfolgreich einsetzen um ganze Szenen zu interpretieren und zu verstehen.

Abstract

This thesis presents a system that strives to give visually impaired people direct perceptual access to images via an acoustic signal. The user explores the image actively on a touch screen or touch pad and receives auditory feedback about the image content at the current position. The design of such a system involves two major challenges: what is the most useful and relevant image information, and how can as much information as possible be captured in an audio signal. We address those problems, based on a Modular Computer Vision Sonification Model, which we propose as a general framework for acquisition, exploration and sonification of visual information to support visually impaired people. General approaches are presented that combine low-level information, such as color, edges, and roughness, with mid- and high-level information obtained from Machine Learning algorithms. This includes object recognition and the classification of regions into the categories "man-made" versus "natural" based on a novel type of discriminative graphical model. We argue that this multi-level approach gives users direct access to the identity and location of objects and structures in the image, yet it still exploits the potential of recent developments in Computer Vision and Machine Learning. During exploration, the user can utilize detected man made structures or specific natural regions as reference points to classify other natural regions by their individual location, color and texture. We show that congenital blind participants employ that strategy successfully to interpret and understand whole scenes.

In loving memory of my grandmother Margarete

Don't only practice your Art,
But force your way into its Secrets,
For it and knowledge can raise men to the Divine.
Ludwig van Beethoven

Words of Gratitude

First of all, I would like to express my gratitude to my advisor, Prof. Volker Blanz, for his incredible support that finally led to the success of this work. I am very thankful that he gave me the freedom to develop and explore own strategies and approaches to encounter the various challenges of my project, while at the same time continuing to contribute valuable feedback, advice, and encouragement.

I am also very thankful for the excellent working conditions and the gifted colleagues I had at the Media Systems Group. I would like to mention Marc Strickert, Joanna Czajkowska, Davoud Shahlaei and Pia Breuer, whom I had many fruitful discussions with concerning several aspects of my project. Especially, I want to thank Davoud most sincerely for reading the manuscript of this thesis. Furthermore, I want to give thanks to Marcel Piotraschke, Matthias Schumacher and Björn Schiefen, whom I enjoyed many rather humorous and relaxing conversations with.

I also want to take this opportunity to express my appreciation to the Rheinischen Blindenfürsorgeverein, Düren, and especially Mrs. Gut and the students Marina, Larissa, Florian and Sascha for their interest and participation in the project. I also thank Tobi Fechner and Rainer Damerius for their highly appreciated advisory support in giving me invaluable feedback about the system design from a congenital blind person's point of view.

Finally, I am deeply grateful for all the friends I am blessed with. It is their continuous support and precious encouragement that made a major contribution to the completion of this thesis.

Contents

Introduction

Motivation

In “Critique of Pure Reason” Immanuel Kant stated that our knowledge of the outside world depends on our modes of perception[1]:

> *What is first given to us is appearance. When combined with consciousness, it is called perception.*

Leonardo da Vinci proclaimed[2]:

> *The eye encompasses the beauty of the whole world.*

Making this visual beauty of the world more accessible to visually impaired people has inspired researchers in Computer Vision for a long time. Perhaps the most ambitious software solution for the vision problem would be an algorithm that produces a semantic description of the image content which is then output of a speech synthesis device in natural language. This automated image analysis system would mimic a partner with normal vision who describes the image to the user. However, despite the fact that automated image understanding will remain a challenge to researchers for many years, it would continue to deprive the visually impaired of a direct perceptual experience, an active exploration, and an impression of where things are in the image and what visual appearance they have. The approaches, described in this thesis, therefore, are to augment the sensory capabilities of visually impaired persons by translating image content into sounds. The task of analyzing and understanding images is still up to the user, which is why we call our approach “auditory image understanding”. Very much like a blind person who explores a Braille text or a bas-relief image haptically with the tip of her finger, our users touch the image (via touch pad or touch screen) and experience the local properties of the image as auditory feedback. Due to the simplicity and directness of the sensory mapping from visual to auditory, we harness the human ability to learn, so we consider the brain of the user as a fundamental part of the system. Visually impaired persons can use the system to analyze images that they find on the internet, but also for personal photos that their friends or loved ones want to share with them. It is this application scenario that makes the direct perceptual access most valuable. The user feedback that we received for our system indicates that visually impaired persons appreciate the fact that they obtain more than an abstract verbal description and that images cease to be meaningless entities to them[3]. Expressed in the words of one adult participant:

> *What amazes me is that I start to develop some sort of a spatial imagination of the scene within my mind which really corresponds with what is shown in the image.*

[1] Kritik der reinen Vernunft, 1781 [240]

[2] Leonardo da Vinci’s contributions to neuroscience, TRENDS in Neurosciences, 2002 [355]

[3] In January 2013, preparations commenced to incorporate our system permanently at a residential school for the visually impaired (Internat des Rheinischen Blindenfürsorgeverein 1886 Düren)

Overview of Employed and Developed Technology

Part II presents our proposed modular design principle for the sonification of visual information. This image sonification model emphasizes a local exploration paradigm within the sonification process. Thus, like a blind person who explores a Braille text or a bas-relief image haptically with the tip of her finger, users touch the image and experience the local properties of the image as an auditory response. One further crucial characteristic of the model would be its high modularity which allows for modules to be altered or even exchanged, without effecting the overall system. Hence, the system can be adapted to a specific application scenario.

The implementation of this model presented in part III is mostly dedicated to scrutinize the possibilities and limitations of recognizing objects through sound. To provide a fundamental basis for such an "auditory object recognition", we extract rather fundamental image characteristics, such as color along with edges of various orientations as well as a measure of coarseness. Before feature extraction, the image is smoothed using an edge-preserving filtering approach. Edge detection is performed based on Gaussian image pyramids and Gabor wavelets. A novel Computer Vision algorithm is presented to filter single and especially repetitive sets of significant edges and discard rather distracting ones. This algorithm incorporates local variance, connected-component analysis, graph-theory and iterative case-by-case analysis approaches. Coarseness is computed for each pixel using a measure based on local entropy of image gradients.

Also in part III, we develop a novel fundamental concept of an audible representation of color space that can be used to convey the concept of colors and color mixing to congenital blind people. This audible color space is inspired by Hering's theory of Opponent colors and represents each color value within the intuitive HSL color space as a mixture of instruments, assigned to the four significant opponent colors. As congenital blind people do not have any previous visualization of colors, color sonification hence becomes a much more challenging task.

Calculating the volumes of instruments in a mixture of sounds for all intermediate colors is formulated as an interpolation problem. Thus, we define a mapping on a set of control points manually to achieve the desired volumes for specific color values and mixtures and interpolate all values in between using a non-linear interpolation approach based on Thin Plate Splines.

Despite most other image sonification approaches, we focus on the simultaneous sonification of multiple image information locally, i.e, at specific positions within the image, depending on the users finger position. Simultaneous sonification of colors, significant edges and sets of edges as well as roughness is realized based on MIDI instruments using an external MIDI synthesizer software.

The modular sonification model and the work on "auditory object recognition" has been published in [18].

In part IV of the work, we consider leveraging Computer Vision and machine learning algorithms to derive and sonify image information on many levels, ranging from low-level such as color, to high-level, as for example scene labeling and object recognition. Machine learning techniques are developed and employed to even pre-select the extraction of specific low-level features in certain areas. Still, the results of these algorithms remain tied to the image pixel where the feature occurs, so the user always knows the location of any detected image entity. Incorporating the imaginative capabilities of a blind person's brain as a fundamental element of the process proves to be a promising combination for more sophisticated tasks, such as "auditory image understanding".

First, robust scene labeling is performed to classify images into man made structures and natural regions. Therefore, a novel type of probabilistic graphical model along with a specific feature set for man made structure detection is presented. The novelty of this model would be that Support Vector Machines are incorporated in the unary as well as the pair-wise potentials. Thus, our novel model is called Dual Support Vector Field. Our feature set surpasses any existing feature sets for man made structure recognition, as it incorporates rather sophisticated features, such as, smoothed histograms of gradient orientations as well as results that are provided by algorithms we devise to capture the specific properties of man made structures, e.g., junctions, line patterns or corner point patterns. Parameter learning in the Dual Support Vector Field is reduced to training a Support Vector Machine and learning an additional scalar model parameter using gradient ascent, both based on a given set of training images and ground truth labelings. Inference using max-flow/min-cut (Graph Cut) algorithms is employed to compute the image labeling based on the Maximum A Posteriori estimate.

Based on our novel feature set, we do not have to compute additional features to sonify important information about man made structures and choose from the feature set directly. Between the sonification of man made structures in general, we will further sonify the most dominant gradient orientation as well as the number of parallel lines within each man made structured region.

Natural Regions are are further pre-processed for sonification applying a textural roughness measure based on the fractal dimension. Therefore, we implement an extension of the regular differential box-counting method for fractal dimension estimation of an image region.

Additionally, a Support Vector classifier along with a specific feature set is presented to verify true or discard false object detections, which have been found by regular object recognition approaches, before sonification to avoid confusion on the side of the blind user, who can not check for a correct detection visually. The feature set incorporates mainly relative information and graph based features that can be extracted from the results of these regular object detection approaches. Due to the rather linear separable and correlated nature of this feature set, Principal Component Analysis is employed to perform a transformation of the feature set before classification.

Within part IV, an intuitive color sonification concept, based on the one presented in part III, will be proposed which represents colors the way they are perceived visually by appropriate fundamental sound characteristics, instead of instruments. Sound generation of this advanced color sonification scheme is entirely based on an integrated additive synthesis concept and, therefore, does not require any MIDI instruments or external MIDI synthesizer software, which makes the whole program more suitable to be operated by blind people.

Feature sonification is performed based on acoustical elements that are selected to not interfere with color sonification. Thus, sonification of man made structures in general is based on a bongo drum rhythm. The number of parallel lines in each man made structured region is acoustically represented by adding reverberation to this bongo rhythm. Additionally, a hi-hat rhythm is employed that varies in speed according to the orientation of most dominant gradient in each man made structure from slow (horizontal) to fast (vertical). Roughness in natural regions is represented by an intuitive audible element based on brown noise. Verified object detections are sonified using auditory icons that avoid additional memorizing of object to sound mappings.

To allow for simultaneous exploration and sonification of an image a non-blocking audio queue is implemented. Feature sonification is performed using pre-computed wave-files with an audio engine library. It allows for post-processing of pre-computed wave-files with sound effects. We harness such possibilities to convey complex features audibly along with colors without additional computational effort to synthesize such sounds additionally. Further, the usage of external audio files allows for an easy exchange of sounds.

The work on “auditory image understanding” has in part been published in [20]. The two novel machine learning algorithms proposed, the Dual Support Vector Field as well as the object detection verification approach, have been published in [19].

The Explorative Image Sonifyer Software

The work on “auditory image understanding” in part IV has been developed into a stand-alone application that is designed to run on *Microsoft* Windows 7 and 8 based tablet PCs, as will be presented in detail in chapter 14. Tablet PCs offer the best solution in terms of computational power, screen size, touch screen usability and mobility. The final software, called *Explorative Image Sonifyer (EIS)*, is designed to allow the user to load images either from a specific folder or, captured, from the tablet’s integrated front camera. As designed to be used by visually impaired, the software can be controlled by only 3 touchscreen gestures. It further guides the user throughout the whole usage of the program via speech output. The system has been developed under considerations of portability, reducing the amount of adaptions needed to run the system on other operating systems as far as possible. To summarize, the features of our *Explorative Image Sonifyer* Software are:

- A stand-alone software to make the internet more accessible to visually impaired by making available image data for interactive audible exploration and helping blind people gaining fundamental image understanding of sceneries.

- The software contains a novel sonification scheme. The sounds associated to visual features (colors, roughness) are selected based on perceptual, semantic and aesthetic considerations. This includes an intuitive color sonification concept, representing colors acoustically the way they are perceived visually by appropriate sound elements. Further, sounds are designed to be sonified simultaneously without interferences or distractions.

- The software is controlled by only a few touch screen gestures. To load an pre-process an image from the tablet PC’s internal camera, a double tap is performed. To load an image from a specific image folder, an open gesture is performed. A closing gestures exits the program. Speech output guides the user throughout the whole usage of the program.

- Specifically developed machine learning algorithms pre-detect man made structures and object within images. Man made structures as well as important edges among them are sonified based on drum rhythms. Natural regions within images are sonified based on their grade of roughness. Detected objects are sonified in an intuitive way using auditory icons that avoid additional memorizing of object to sound mappings.

- The loaded / captured image is rendered in the middle of the tablet PC’s screen, with found man made structures and objects highlighted, to allow a blind person to discuss the image with a normal sighted friend. An interactive color to sound mapping chart is rendered on the left side of the screen to allow a blind person to quickly look up the sound of specific color mixtures.

- The software is internally designed as a finite state machine and implements a non-blocking audio queue to allow parallel real-time exploration and sonification.

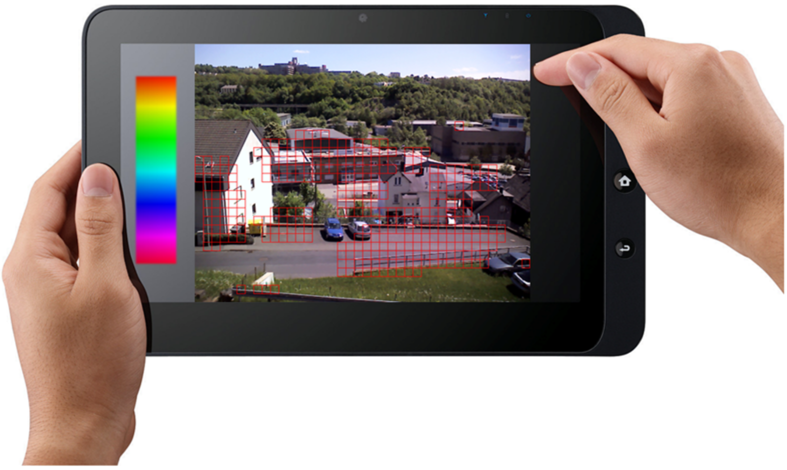


Figure 1: The *Explorative Image Sonifyer* System on a regular *Microsoft* Windows 7/8 tablet PC. Detected man made structures are highlighted (using squares). On the left side, the color to sound mapping

Part I

Sonification & Auditory Perception

Chapter 1

Introduction to Sonification

1.1 Definitions

Sonification as a field of research is a sub-type of the area of **Auditory Displays**. Auditory Displays, formalized in [203] (see figure 1.1), denote systems that use sound to communicate information to allow a human user to understand data through listening. Sonification is a rather young discipline, however, its conceptual roots can be traced back to poet Rainer Maria von Rilke. He suggested, already in 1909, to use a phonograph needle to "seek" sounds in the lines of world materials and, thus, transform experience into another field of sense [387]. Publications can be found since about 25 years. The first International Conference on Auditory Display (ICAD) was held in 1992. Sonification is defined by Kramer et al.[257] as

> *the use of non-speech audio to convey information. More specifically, sonification is the transformation of data relations into perceived relations in an acoustic signal for the purposes of facilitating communication or interpretation.*

Although this definition is the most accepted one, others exist, such as that given by Scaletti [413], which states sonification as

> *a mapping of numerically represented relations in some domain under study to relations in an acoustic domain for the purpose of interpreting, understanding, or communicating relations in the domain under study.*

In effect, a definition quite similar to the first one. Starting from Scaletti's definition, Barass [23] develops the concept of what he calls **Auditory Information Design** considering both the specific information to be sonified as well as the design of possible acoustical representations:

> *Auditory Information Design is the design of sounds to support an information processing activity, focusing on the specific task like interpreting, understanding or communicating relations in the data.*

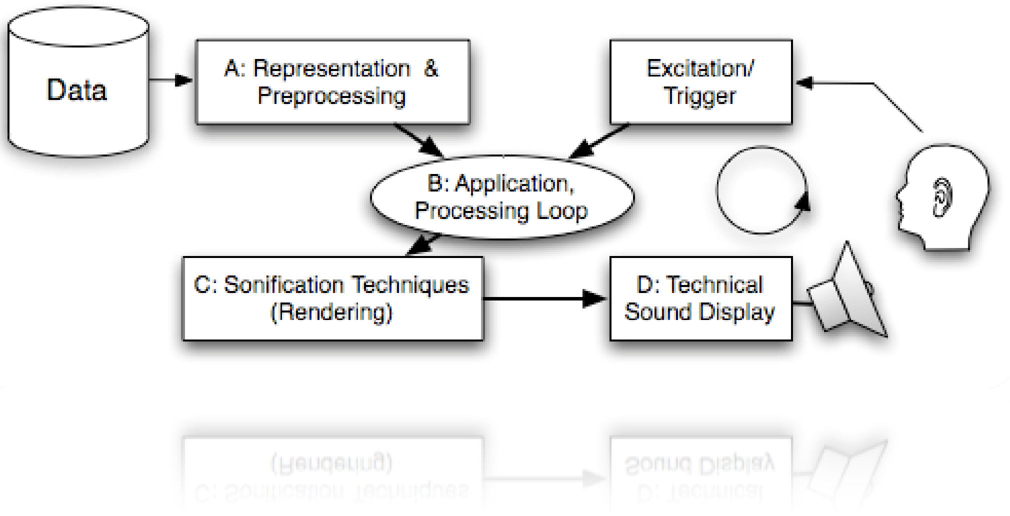

Data
A: Representation & Preprocessing
Excitation/ Trigger
B: Application, Processing Loop
C: Sonification Techniques (Rendering)
D: Technical Sound Display

Figure 1.1: Auditory display systems, as formalized in [203], include (A): information pre-processing, (B) techniques for data processing and computation, (C) a sound synthesis engine, (D) the user. Figure taken from [203]

Although the definitions given above differ slightly, one can still point out some basic requirements for a sound to be called a sonification. First, the sound synthesis depends upon the data of the domain under study, and second, the intention for synthesizing sounds is to learn something about the data by listening. Thus, the sound itself is only regarded as the medium of communication. Further, speech is by definition primarily prohibited as a sonification method. However, speech, provided in the form of a speech interface, can be a valuable element in auditory displays as it is able to provide additional annotations or explanations about the data-set.

By definition sonification differs from data driven music composition ([415]; [374]; [375]; [513]) where the intent is primarily an aesthetic. With data driven music composition, the sound itself becomes more than just the medium of communication and delivering information is of minor or no importance at all. Thus, data driven music composition is not about learning anything about the data.

The motivations for communicating information using auditory displays, rather than visual displays, have been thoroughly discussed in the literature ([65]; [201]; [256]; [333]; [352]; [407]). Briefly, auditory displays exploit the superior ability of the human auditory system to differentiate temporal changes and patterns ([50]; [149]; [150]; [159]; [65]; [257]; [310]; [317]). As a result, Auditory Displays may be a very appropriate modality to examine information containing complex temporally varying patterns.

In every day work environments auditory systems can be, and are, employed to support or warn people besides the information given through any visual display. Especially, if the person already has to focus on many visual entities ([51]; [56]; [147]; [507]) or is visually impaired ([147]; [257]; [498]; [499]; [506]).

Aesthetics

Generally, Edworthy [130] emphasized the independence of sonification as a way to convey information and aesthetics. However, it is not quite clear yet in how far the performance of an Auditory Display can be improved if sonification methods are developed under aesthetic and musical concerns [500]. Vickers and Hogg [486] argue that a more careful attention to aesthetics would facilitate ease of listening and in turn promote comprehension of information displayed through sonification.

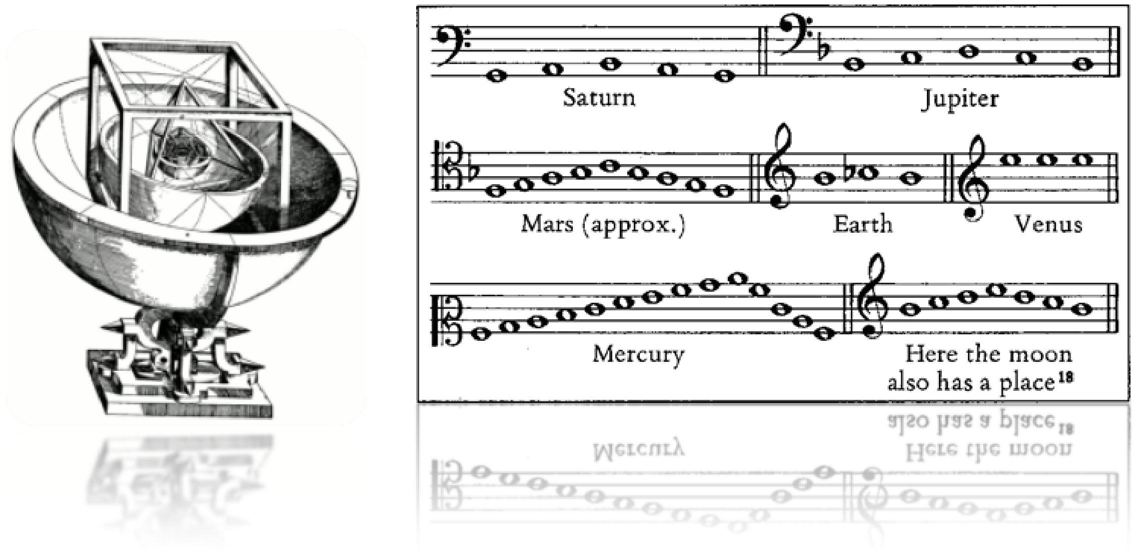


Figure 1.2: Left: Keplers Solar System, Right:“the Music of Sphere”. Illustrations taken from [245] and [246])

1.2 Research in Auditory Displays

Athough not yet an explicit field of research, sonification techniques have implicitly been employed to various applications throughout the centuries. Already Johannes Kepler proposed a kind of “scientific” sonification in his “World Harmonics” [246], where he tried to express motions and distances of planets using scales (see figure 1.2).

A detailed introduction to early researches in auditory displays and sonification can be found in [256] and [204]. Briefly, one can say that pioneering efforts were made by Pollack and Ficks [364] in 1954, who investigated the usage of abstract auditory variables for the presentation of quantitative information. Their sonification method incorporated changes in alternating tone and noise bursts, employing attributes such as pitch, volume, duration, stereo panning and others. Their studies revealed that auditory displays using multiple sound parameters generally outperformed selected uni-dimensional displays [256]. Furthermore, well founded research about auditory classification capabilities has been conducted by Bly [37] in 1982, exploring the classification of non-ordered multidimensional data sets using Parameter Mapping Sonification (see section 1.3) to represent the data. Experiments with various mappings and training methods were performed to compare displays of either only employing sound or graphics or both. Results proofed auditory displays to be as effective as the visual display, and a combined display to perform even better.

Nowadays, auditory displays and sonification are harnessed in a wide variety of fields. Applications range from desktop computer and mobile phone interfaces for visually disabled people and data mining, to chaos theory, molecular biology, particle physics and cosmology, just to name a few. Equally wide spread are the research disciplines:

- **Acoustics**: Examining the physics of sound creation can be helpful for the selection or design of sound synthesis techniques to represent data. For Model Based Sonification (see section 1.3), acoustics can supply appropriate templates for a model and its dynamics. Finally, computed sounds, given as digital representations, are transformed into acoustical sound waves by sound cards, synthesizers and loudspeakers.

- **Sound Engineering**: Sound Engineering is concerned with the technical realization of sound spatialization, from mono to multi speaker setups, as well as sound signal changes according to reflections in the listening room.

- **Statistics and Data Mining**: In case of high-dimensional data to be sonified, appropriate techniques can be employed for data pre-processing, such as data dimensionality reduction.

- **Human Computer Interaction (HCI)**: The discipline of human computer interaction in general is all about system design to provide and optimal usability for a specific system. Hence, valuable insights from this discipline can be employed to develop sonification systems.

- **Physiology and Neurobiology**: Both disciplines deal with the processing of sound signals after reaching the auditory cortex. Therefore, they are crucial to understand how signals are further processed within the ear and what kind of signals can be used for sonification from a neurobiological perspective. Other aspects include, e.g., the processing speed for auditory signals or connections between specific sounds and emotional states.

- **Musicology**: Findings in Auditory Gestalt Principles [509] or Auditory Streaming [50] provide guidelines for the usage of sound within any sonification approach. Musicology can give a methodology for the organization of the acoustic data concerning harmonies or rhythm and provides tools for the analysis of musical pieces. Hence, it also be harnessed to control and describe sonification approaches.

1.3 Sonification Techniques

We will now give a brief overview over some of the main techniques in sonification, some of which will be employed for our specific applications in part III and IV.

Audification

Audification is the most simple and direct auditory display technique for translating data into sound. Kramer [256] defines it as

> *the direct playback of data samples I refer to as "audification".*

He later updates this definition:

> *Audification is the direct translation of a data waveform into sound.*

It is a direct acoustical alternative approach to visualization, since all abstract data series might be either visualized or sonified. So [206] proposes another definition:

> *Audification is a technique of making sense of data by interpreting any kind of one-dimensional signal (or of a two-dimensional signal-like data set) as amplitude over time and playing it back on a loudspeaker for the purpose of listening.*

Since all data end up in a loudspeaker, audification is essentially a continuous, non-digital interpretation of data sets.

What makes audification difficult is that, obviously, a lot of data values are needed even for a short audification. Additionally, audification is limited to data sets which can be ordered in some reasonable way, such as e.g time series data. In some applications, however, exactly those data sets are available, such as with the analysis of dynamic systems [311] or seismic measurements [196].

Being the simplest sonification method, audification often serves as an initial approach to a new sonification task. However, it is later mostly replaced by more sophisticated methods.

Auditory Icons

Auditory Icons, first invented and employed by Gaver [160], are the acoustical equivalent to the "clickable" visual icons, nowadays common on every modern computer graphical user interface. Auditory Icons mimic everyday "real-world" non-speech sounds that we might be already familiar with. Therefore, the meaning of such sounds does not have to be learnt. As an example, the deletion of some data file might be represented by the sound of tearing a piece of paper.

Earcons

Auditory Icons, previously introduced, require an existing relationship between the sound and its meaning, which may not always exist. In such cases, Earcons can be utilized. Blattner et al.[34] defined Earcons as

> *non-verbal audio messages used in the user-computer interface to provide information to the user about some computer object, operation, or interaction.*

Brewster [52] refined such a definition. He formalized Earcons to be

> *abstract, synthetic tones that can be used in structured combinations to create auditory messages.*

Thus, Earcons can be thought of as short, structured musical messages that correspond to certain elements of the data being communicated. As Earcons are defined to not necessarily provide an already known relationship between the sound and the information that it represents. Hence, the relationship between sounds and information, synthetically created through Earcons, have to be learned initially. A well known example of an Earcon would be a specific mobile phones ringing sound associated with a certain caller or the specific and nowadays very familiar sounds representing the plugging and un-plugging of a peripheral device to or from the computer.

Parameter Mapping Sonification

Parameter Mapping Sonification [204] involves the association of information with auditory parameters for the purpose of an auditory display of the data under scrutiny. Since sound is inherently multidimensional, Parameter Mapping Sonification is considerably appropriate to display especially multivariate data. Therein lies both the power and difficulty of this method. Grond and Berger [184], point this out:

> *The enormous range of interpretive mapping decisions provides equally enormous opportunities to create an appropriate auditory display for a particular desired purpose. However, the wide variety of mapping possibilities poses a challenge in terms of consistency and comprehensibility.*

Figure 1.3 illustrated the general design process of Parameter Mapping Sonification. It involves the translation of data features (figure 1.3 (left)) into sound synthesis parameters (figure 1.3 (right)). The design involves an interplay of, and the conscious intervention in both the data and the signal domains. Thus, Grond and Berger [184] suggest an effective Parameter Mapping Sonification system to involve some compromise between intuitive, pleasant, and precise display characteristics. They state that *integrating both worlds is key in creating effective sonification.*

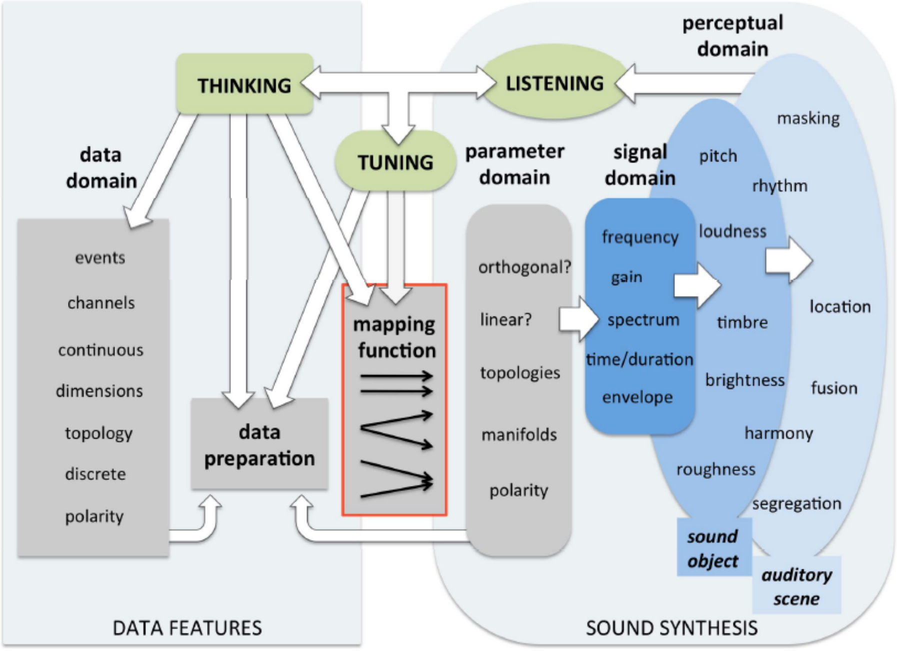


Figure 1.3: The general design process of Parameter Mapping Sonification. Figure taken from [184]

Model Based Sonification

Model Based Sonification, first proposed by Hermann [208], [204], is inspired by the observation that almost every human activity and interaction within the world is accompanied with an acoustic response and a rich feedback about the nature of the involved materials, as well as the strength and type of contact. Model Based Sonification is a sonification technique that scrutinizes how acoustic responses are generated in response to a user's interactions, and develops a framework to govern how these insights can be applied to the sonification of data. Consequentially, it is defined as a sonification approach that models a dynamic system, evolving in time, depending on the data and the users interaction to generate an acoustic signal. Hence, such a sonification model is essentially a set of instructions for the creation of what Hermann calls a "virtual sound-capable system" [204] and for how to interact with it. Such a model would remain silent in the absence of excitation, and start to change according to its dynamics only when on user interaction. The acoustic response however is directly linked to the temporal evolution of the model.

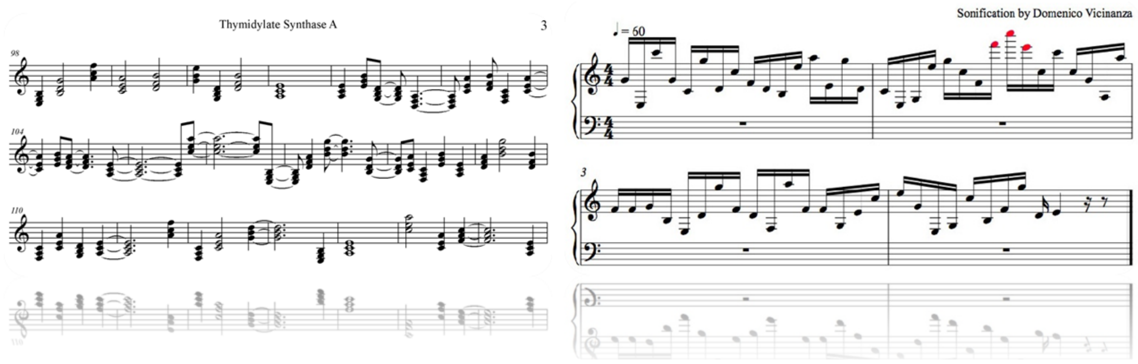


Figure 1.4: Left: Excerpt of a musical score based on the human "Thymidylate Synthase A (ThyA)" protein sequence. Amino acids are assigned to a note starting an octave below middle C with rhythm based on human codon distribution. [454]. Picture modified from [454]. Right: Different energy levels mapped to different notes on a traditional musical scale. Three, red colored, notes, an F, C, and E, represent the new found particle. Illustration by Domenico Vicinanza

1.4 Fields of Application

The various areas of applications of sonification have been thoroughly discussed and correlated in [206], so that we focus on giving only a brief impression here. One of the first successful applications is the Geiger counter, where amount and frequencies of audible clicks directly represent the radiation level in the device's immediate vicinity. Other common applications include SONAR (Sound Navigation and Ranging) [475], medical [147] and cockpit [151] auditory displays. There have been some publications about the application of sonification for the exploratory analysis of specific types of data, e.g. for the analysis or representation of certain chaotic systems ([176]; [308]) fluid dynamics, seismology [116], or the analysis of topological properties of graphs in higher dimensions [13] or network traffic. A quite entertaining example of the latter would be Tweetscapes by [207], which offers a real-time sonification of Twitter data streams.

More recent research proofs the effectiveness of sonification on motor learning, in competitive sports [414] as well as in rehabilitation [489]. Audible perception is then harnessed to support the observation and reproduction of basic movements, which are essential elements to learning a new closed skill in sports or relearning basic motor skills in rehabilitation. These processes, dominated by visual perception, can be augmented utilizing audition as another perceptual channel, suitable for gathering information about movement patterns. One can e.g. hear the rhythm of a runner, even of a swimmer, even more precisely as one can see it.

Furthermore, sonification can be used in the natural sciences to either support visual observation of data, or, as in case of visually impaired researchers, open up access to the data more readily. In molecular biology it is employed to audibly browse the RNA structure ([186]; [94]; [327]; [194]; [195]). In [454] whole musical scores are converted from genome-encoded protein sequences in order to hear auditory protein patterns (see figure 1.4 (left)). In [488] a sonification method is proposed to develop an “acoustic standard model of particle physics” and quite recently researcher at CERN utilized sonification to musically illustrate a exciting irregularity within their data. It corresponded to a particle weighing in at 126 gigaelectronvolts (GeV), consistent with the Higgs Boson that is believed to give mass to all other particles [442]. Physicist and engineer Domenico Vicinanza, a member of the team responsible for sonification at CERN, was previously involved in the creation of music from volcanic activity, facilitating to spot potential eruptions around the world due to altering a musical pitch. He appreciated the use of sonification to make this potential breakthrough in physics easier to understand by the general public. The team mapped different energy levels to different notes on a musical scale. Finally, there exist couple dozen notes, representing particle background noise, and suddenly, a spike up two octaves. In that spike, one finds three notes, an F, C, and E, representing the mysterious new particle (see figure 1.4 (right)). However, the last example of applied sonification is meant to be more than simple data analysis or a broader way to understand physical concepts. Vicinanza in his own words:

> *Both science and music are searching for harmonies, searching for regularities, ways to feel an inner peace and harmony in the universe. There is an inner beauty in the nature, in what’s around us. It’s that inner beauty that I really wanted to convert into music.*

1.5 Auditory Displays for the Visually Impaired

In recent years some research on the application of sonification to support visually impaired people has been carried out. The first technical audio system for blind people was the Optophone, a reading machine with an audible output developed 1912 by Fornier D'Albe [153]. It produced a six-tone code for letters in scanned documents. As speech synthesis was not available at that time, sonification was used as a replacement.

To enable access to personal computers, "screen readers" were eventually developed. Screen readers examine and convert the contents of the screen into sounds so that even modern graphical user interfaces (GUI) become accessible to a blind person. However, there have been attempts to create an auditory version of the GUI, called Soundtrack [128]. Soundtrack retains most of the interactions of the GUI, such as windows, scrollbars, icons, but represented them in an auditory form, using acoustical parameters such as varying pitches that represent relative spatial information. A single mouse click initiates a spoken label and a double-click then activates the current object. Soundtrack is one of only few attempts to design a non visual, mouse based, graphical user interface. Roth [399] proposed AB-Web, an active audio browser that conveys information about the structure or layout of a document while browsing websites on the internet.

Apart from the examples already presented in section 1.4, there has been further development in the application of sonification in the natural sciences. The development of so called Soundgraphs ([129]; [55]; [304]; [378]) allows a blind user to sonify a cartesian graph, getting an impression about its slope, turning points a.s.o. Grond et al. [185] go even further by acoustically displaying the first m terms of the Taylor series. To allow very direct interaction with the mathematics , a user should be able to move along a curve, sensing significant points. This could be, e.g., hearing a local maximum by a variation of pitch. Yu et al. [378] facilitated such a direct interaction with sonified graphs by adding haptic interaction via a force-feedback device. Further works ([24]; [330]; [166]) deal with printed mathematical equations that provide a significant amount of information in an highly succinct manner. As the visual representation immediately and unambiguously indicates structural information, a similarly efficient and unambiguous representation for the visually impaired is the goal of such a research.

Navigation is yet another great issue for the visually impaired. Hence, some research has been done to make use of sonification to translate useful information on different levels. Such useful information might be rather fundamental such as basic colors [40] as well as more sophisticated such as GPS ([517]; [532]) or depth data [39]. However, as acoustical awareness of his surroundings is crucial for a blind person, navigation systems often make use of rather haptic feedback, such as the Navbelt system ([42]; [434]).

Recent developments strive help blind users solve general visual search problems asking multiple helpers online in near real-time [29] or even allow a visually-impaired person to safely operate a motor vehicle [212]. The system employs an audio-tactile interface approach to convey information about the vehicle's speed and heading and the driver then uses a joystick to correct his steering and speed.

Sonification of Images

As the focus of this thesis is on sonification of images for the visually impaired, we give a separate brief overview of the developments in this area. One of the first examples to render an entire scene acoustically is TheVoice project [313]. It creates a representation of the visual picture pixel-by-pixel. The vertical positions of pixels are represented by pitch. Horizontal positions, from left to right, are represented by time, and brightness is represented by loudness. As the system scans horizontally across the image a vertical column of pixels is sonified as a single complex sound at each horizontal position. The start of a scan is marked by a "click". After the image has been scanned the procedure starts all over again. The mapping is quite simple, making the system suitable for real-time navigation.

A more recent development that follows a similar paradigm is the SeeColOr mobility aid ([40]; [39]). Other than lightness, it transforms small portions of a colored video image into sounds represented by spatialized musical instruments, which is based on the quantization of the HSL color system (see appendix B.2). The purpose is to allow for blind people to perceive elements of an environment in real time. In [39] the work is extended to sonification of depth information, employing additional rhythms. The rationale behind is to enable a blind individual, e.g., to follow a path painted on the ground in an indoor environment, such as a shopping center or a medical center. Apart from navigation the SeeColOr system was used to perform experiments on very fundamental color image interpretation tasks [40].

Various other approaches to image sonification exist without the intention to be employed in any navigation context. One of which would be SmartSight ([92]; [90]; [91]). It is a simple form of translation from visual pixel information of monochrome images to non-verbal sounds of different pitch. An auditory cursor sweeps across the graphic horizontally. As it intersects a black pixel it forms a sound, the pitch of which represents the vertical height of the pixel. The main idea is to enable blind users to detect basic object shapes. Quite similar tool would be the GUESS system [236] or the more recently proposed EdgeSonic [525]. EdgeSonic focuses on the sonification of the progression of and the distance to edges in images. As with SmartSight, the image is pre-processed using simple edge detection algorithms, such as [67], adjusted to extract only dominant edges within the image. The GUESS System on the other hand employs Blauert's approach on spatial hearing using headphones [35]. One of 3 predefined basic shapes is rendered by a moving sound that is acoustically represented within a 2D virtual sound space, as described in [400]. If, for instance, the rendered shape would be a right triangle, the user hears a tone descending vertically in the right speaker channel, then moving horizontally from the bottom right to the bottom left channel and finally, it rises from the bottom left back to the top right channel, back to its initial position.

Rather than aforementioned methods the EdgeSonic and the GUESS system both intro-

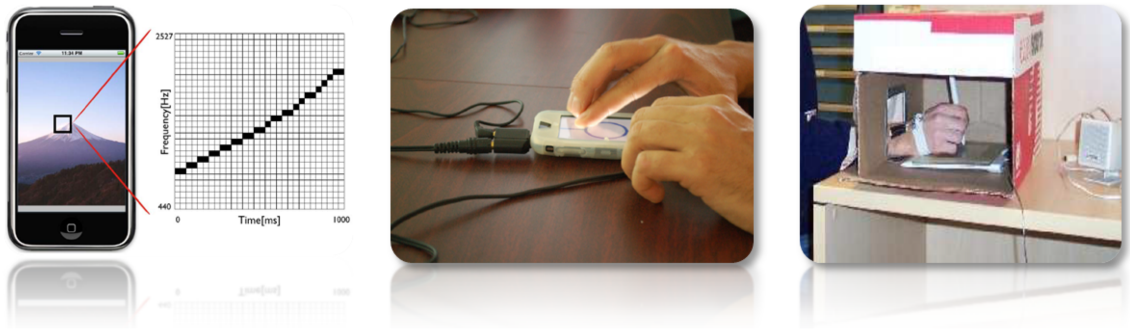


Figure 1.5: Left: Mapping from image features to sound with EdgeSonic. Figure taken from [525]. Middle: A participant using the Timbremap device during a study. The shape is displayed on the screen and the participant is completely blind and can not see the shape. Figure taken from [450]. Right: Experimental Setup of the SoundView system. A participant is wearing the buzzer on a wristband attached to his wrist. Note the occlusion of the participants view of his hand by the box. Figure taken from [481]

duce a kind of explorative interaction within the image. Although the user has to stay for 1000 milliseconds at a certain image position until the area below his finger is completely sonified from left to right (see figure 1.5 (left)), he is generally able to freely explore the image. Sonification is performed by scanning each column in a local area in the binary edge image below the user's finger position. Each of the 30 pixel positions in each column is acoustically represented by a single frequency oscillator. Each of 30 frequency oscillators is then turned on or off depending on whether its corresponding pixel lies on an edge (see figure 1.5 (left)). A horizontal line in the edge image would then yield a long lasting single frequency sine wave. A vertical line yields a single bleep sound, where all frequency oscillators are turned on simultaneously for a short duration. The GUESS system utilizes a stylus on a graphical tablet as an input device to explore an image. Another system that exploits explorative sonification is called Timbremap [450]. It sonifies local visual information based on the location of a user's finger on a map. Timbremap helps the user to navigate through a map by sonifying distances to lines on the map 1.5 (middle)). By placing his finger on a map, finger positions with respect to closest lines are transposed into audio signals. The system harnesses stereo panning to represent the location of the finger according to a particular line. The SoundView system [480], [481] is one of the first frameworks to combine color sonification with explorative interaction. Hence, an image is mapped onto a kind of virtual surface with a color-dependent roughness texture that is then explored by moving a pointer device over the image. This device acts as a kind of virtual gramophone needle 1.5 (right)). The sound produced depends on the motion as well as on the color of the area explored, as color attributes are used to filter an underlying white noise using Subtractive Synthesis. The Walk-on-the-Sun project [376] makes use of explorative image sonification using a user's feed instead of his fingers. As a user moves across images projected onto the floor, his movements are visually tracked and used to select pixels in the images which are immediately transformed in to sound, i.e. those of instruments varying in pitch. Colors are mapped to one of 9 instruments and brightness to one of 50 pitches. A user's location within an image is mapped to panning position, creating a considerable number of differentiable musical events.

More general models of image sonification and the transformation of visual information to music, without the direct intention to support the visually impaired, have been proposed over the years. [306] demonstrates a technique, using color patches within images as chromatic patterns that can be put together to form a melody. [348] mixes 8 pre-recorded musical timbres depending on the quantity of 8 hue values within an image. The Sonic-Panoramas project [235] is motivated to develop real-time interactive sound environments, such as those required in art or virtual reality and to investigate the ways in which humans perceive physical landscapes. Therefore, a goal is to enrich a participants experience of space based on acoustical interpretations of visual landscapes and to develop an interface for data exploration. Movements of users through a projection space are tracked and utilized to generate visual and auditory representation in real-time and position specific. However, these approaches rather refer to musical composition than pure sonification, which is outside the scope of this thesis. For a more in-depth review on the application of image sonification methods to musics have a look at ([523]; [522]; [162]). Giannakis and Smith [162] furthermore provide a review of auditory-visual associations that have been studied in research in computer music.

The Brain as (the) Essential Component of the Sonification System

Most of the systems designed as visual substitutions for the visually impaired, such as TheVoice [313], SmartSight [92], Timbremap [450] or EdgeSonic [525], perform rather minimal image processing and sound mapping. The rationale of those systems is to harness the brain's "plasticity", the brains ability to shift or extend processing of a specific senses such as listening to non used areas ([175]; [472]; [155]; [314]) enhancing these capabilities. In blind people such regions of the brain are primarily the areas involved in visual processing. Therefore, one fundamental intention would be that users will learn to interpret auditory scenes naturally. However, recent research reveals that visual substitution systems or **sensory substitution devices (SSD)** in general could be also harnessed in research on the effects of blindness to the brain as well as in rehabilitation programs ([380]; [448]).

In traditional neuroscience, the common view, known as the "the Sensory Division-of-Labor Principle" [530], is that the human brain is divided into separate sections, such as the visual cortex (see appendix B.1) or the auditory cortex (see section 2.1) according to each sensory modality which arouses it. From these uni-modal cortices the brain then integrates information in higher order multi-sensory areas.

However, various studies suggest this view to be not fully correct ([347]; [10]; [448]; [359]). As already mentioned, in the blind, it is well-known that the visual cortex has been plastically recruited to process other modalities, and even language and memory tasks ([155]; [314]). Lots of those changes commence within days following the onset of blindness [314], and, therefore, do not only affect congenitally blind individuals, although, probably to a different extent. Evidence demonstrates that, in both sighted and blind individuals, the **occipital visual cortex** is not purely visual and its functional specialization could be proofed to be independent of visual input [448]. This leads to the assumption that the brain is task-oriented and sensory modality-independent ([381]; [447]). Such a task selectivity could be demonstrated for various tasks and areas. For instance, a tactile Braille script, a rather simplistic form of vision-substitution is employed in [381], transforming written letters, to reveal that the **visual word form area (VWFA)**, a visual area responsible for processing written language in the sighted [104], also considerably arouse to Braille words in the congenitally blind [381] (see figure 1.6 (right)). Thus, the VWFA specializes in the perception of written words, irrespective of the sensory channel through which they are presented, and even regardless of visual experience.

Using more advanced sensory substitution devices, capable of transforming more complex visual scenes, it is possible to test the sensory modality-independence of other occipital areas dedicated to the processing of more complex visual categories. As an example, a region within the human **lateral occipital complex (LOC)** is activated by objects when either seen or touched. Hence it is named the **lateral occipital tactile-visual (LOtv)**

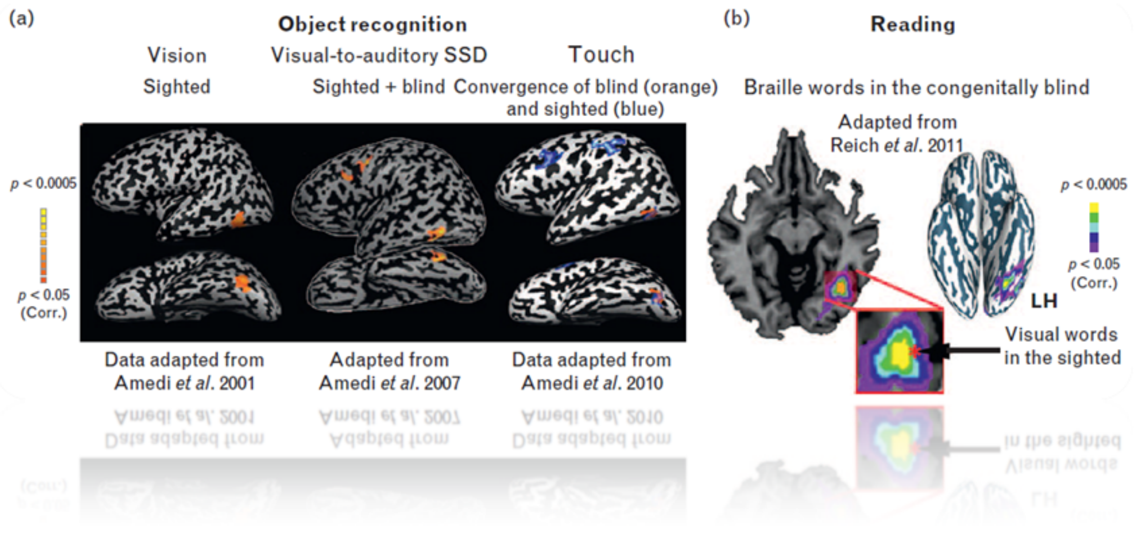

(a)
Object recognition
Vision
Sighted
Visual-to-auditory SSD
Sighted + blind
Touch
Convergence of blind (orange) and sighted (blue)
p < 0.0005
p < 0.05 (Corr.)
Data adapted from Amedi et al. 2001
Adapted from Amedi et al. 2007
Data adapted from Amedi et al. 2010
(b)
Reading
Braille words in the congenitally blind
Adapted from Reich et al. 2011
p < 0.0005
p < 0.05 (Corr.)
LH
Visual words in the sighted

Figure 1.6: *Modality-independent task-specific activations in various areas of the visual cortex. Left (a): Activation of the lateral-occipital complex (LOC) during object recognition using vision, touch and visual-to-auditory sensory substitution. Right (b): Specific activation of the visual word form area (VWFA), the site of activation to visual written words in the sighted, during tactile Braille reading in the congenitally blind.* Text and picture modified from [380]

region [9]. It could as well be activated by auditory stimuli, delivered by the TheVoice [313], that conveyed shape information (see figure 1.6 (left)). The LOtv did, however, not respond the typical sounds which regular objects might produce, which generally do not provide any shape information [9]. This strengthens the notion that the LOtv specializes in the processing of objects' shapes irrespective of the input sense.

However, the tremendous benefits, provided by such a plasticity, come with a considerable potential danger. On the one hand, it helps a blind person to better cope with blindness by harnessing compensatory capabilities, on the other hand it bears the risk to interfere with sight restoration efforts, by disturbing the visual cortex's original functions and unfortunately there exist several cases of medical sight restoration that support such an assumption ([341]; [180]). Although visual information was available to the patients' brain and some visual abilities were restored quite fast, those individuals showed very serious deficits in practical visual perception tasks such as shape and face recognition ([144]; [341]; [180]), just as if the regained visual information would be offered to a fully untrained brain in analyzing and interpreting exactly this data.

However, Reich et al. [380] infer, that

> *if the hypothesis of the highly flexible task-oriented and sensory-independent brain applies, the absence of visual experience should not limit proper task specialization of the visual system, despite its recruitment for various functions in the blind, and the visual cortex of the blind may still retain its functional properties using other sensory modalities. This is very encouraging with regards to the potential of visual rehabilitation.*

Reich et al. [380] and Striem-Amit et al. [448] propose sensory substitution devices in general and visual substitution system in particular to be potentially used as:

- *a research tool for assessing the brain's functional organization*
- *an aid for the blind in daily visual tasks*
- *to visually train the brain prior to invasive procedures, by taking advantage of the visual cortex's flexibility and task specialization even in the absence of vision*
- *to augment post-surgery functional vision using a unique SSD-prostheses hybrid*

Contributions of this Thesis

In their recent research review on the brain's plasticity and its implications for visual rehabilitation using noninvasive and invasive approaches Reich et al. [380] specifically scrutinized possibilities of TheVoice [313] system. They summarize:

> *The described high-level functional abilities using SSDs, as well as reported evidence that the adult brain retains an impressive capacity for visual learning [340], encourage the further development of advanced devices. Especially important are the use of more pleasant stimuli, the delivery of complementary color and depth information, the combination of computer vision techniques to ease the stimuli interpretation and comfortable ergonomic design that will fit daily use. It will be fascinating to see what level the users will be able to reach with further prolonged experience, with technology opening more and more doors. From a scientific perspective, it will be especially valuable to assess the level of acquired visual abilities in the congenitally blind despite what is considered to be an irreversible critical period. The delivery of color through SSDs would be of particular interest in this regard, as this feature is unique to the visual modality and thus considered as a concept that the delivery of color through SSDs would be of particular interest in this regard, as this feature is unique to the visual modality and thus considered as a concept that could not be understood or perceived by the congenital blind.*

The work presented in this thesis proposes exactly such an advanced development and first evaluations of an advanced visual sensory substitution device. Our tool is primarily designed to make the internet more accessible to visually impaired by making available image data for interactive audible exploration. An approach which helps blind people gaining fundamental image understanding of sceneries.

We develop a fundamental concept of an audible representation of color space that can be used to convey the concept of colors and color mixing to congenital blind people. As congenital blind people do not have any previous visualization of colors, color sonification hence becomes a much more challenging task. We very much endorse the concept of interactive exploration in image sonification in our application. Very much like a blind person who explores a Braille text or a bas-relief image haptically with the tip of her finger, our users touch the image (via a touch pad or touch screen) and experience the local properties of the image as auditory feedback. Despite most of the image sonification approaches given in subsection 1.5, we focus on the simultaneous sonification of multiple image information locally, i.e, at specific positions within the image, depending on the users finger position.

Our system design is highly modular, as discussed in part II of the thesis. Hence, parts of it can be altered or even exchanged to the task at hand, without effecting the overall system. Thus the implementation presented in part III is mostly dedicated to scrutinize the possibilities and limitations of recognizing objects through sound. To provide a fundamental basis for such an "auditory object recognition", we extract rather fundamental image characteristics, such as color along with edges of various orientations as well as a measure of coarseness. Simultaneous sonification is realized based on MIDI instruments. Basic recognition tests (see chapter 9) revealed that audible object recognition can be performed, though on a very limited complexity scale. Hence, in part IV of the work, we consider leveraging Computer Vision and Machine Learning algorithms to derive and sonify image information on many levels, ranging from low-level such as color, to high-level, as for example object recognition. Machine Learning techniques could be successfully employed to even pre-select the extraction of specific low-level features in certain areas. Still, the results of these algorithms remain tied to the image pixel where the feature occurs, so the user always knows the location of any detected image entity. Incorporating the imaginative capabilities of a blind person's brain as a fundamental element of the process proved to be a promising combination for more sophisticated tasks (see chapter 13), such as scene understanding. Additionally, within part IV, the color sonification is refined to allow the user at least in partial to perceive acoustically what corresponds to the visual perception of a seeing person. The advanced concept does not require any MIDI instruments and therefore no external MIDI synthesizer, which makes the whole program more suitable to be operated by blind people.

As will be discussed in chapter 14, our framework has been implemented on a *Microsoft* Windows 7 based tablet PCs to combine computational power with the most possible mobility. It is designed to allow the user to load images either from a specific folder or, captured, from the tablet's integrated front camera. Furthermore, the system has been developed under considerations of portability, reducing the amount of adaptions needed to run the system on other operating systems as far as possible. To summarize, our contribution to existing approaches to image sonification for the visually impaired are:

- A tool to make the internet more accessible to visually impaired by making available image data for interactive audible exploration.
- An approach which helps blind people gaining fundamental image understanding of sceneries, found either in images from the internet, or captured by a camera in a certain environment.
- A highly modular sonification model, especially designed for image sonification for the visually impaired.

- A novel sonification scheme. The sounds associated to visual features (colors, roughness) are selected based on perceptual, semantic and aesthetic considerations. Further, sounds are designed to be sonified simultaneously without interferences or distractions.
- An intuitive color sonification concept, representing colors acoustically the way they are perceived visually by appropriate sounds.
- A fundamental concept of an audible representation of color space that can be used to convey the concept of colors and color mixing to blind and especially congenital people.
- A Multi-level image analysis paradigm, combining low-, mid- and high-level features in each pixel. Thus, we overcome the limits of manual acoustical object recognition, employing Machine Learning techniques to pre-detect specific regions within an image, allowing a more directed and individual extraction of low-level features.
- A stand-alone application, designed to be used by visually impaired people on their personal computers. Thus, the software is easily operable and supported by speech output, guiding the user throughout the whole usage of the program.

Our visual substitution device is primarily designed as an aid for the blind to make images explorable. Especially the system in part IV that allows an "auditory image understanding" could be used by visually impaired people to analyze images that they find on the internet, making it more accessible, but also for personal photos that their friends or loved ones want to share with them.
It would, however, be fascinating if it could be employed as *a research tool for assessing the brain's functional organization* or *to visually train the brain prior to invasive procedures, by taking advantage of the visual cortex's flexibility and task specialization even in the absence of vision* as Reich et al. [380] and Striem-Amit et al. [448] propose sensory substitution devices in general to be utilized for.
As such, the system presented in part III could be also utilized to help congenital blind persons to train shape or orientation recognition of basic objects. Part IV could serve as a means to develop spacial understanding of spatial relations of objects within whole scenes. In this context, the direct perceptual access becomes most valuable. The user feedback that we received for the system, especially in part IV, indicates that visually impaired people appreciate the fact that they obtain more than an abstract verbal description and that images cease to be meaningless entities to them.

Moreover, our system can be employed to analyze photographs which blind people have taken by themselves. Interestingly, recent studies ([4]; [225]; [270]) indicate that there is a significant interest in being able to take and organize their own pictures in the blind and visually impaired community. Adams et al. [4] recently developed a mobile app to help blind persons take and organize pictures using non-visual cues. Interestingly, Adams et al. [4] anticipate utilizing our principles to sonify images that we propose for “auditory image understanding” in [20] to extend their work and enable users to interpret the content of a photo by interacting with the photo through a touch-screen interface.

Chapter 2

Auditory Perception

2.1 The Ear & the Auditory Cortex

Auditory perception in general is a very complex task and the auditory system has some remarkable abilities to deal with it. Thus, especially in the context of designing an auditory display, it is crucial to understand and incorporate these into the system design appropriately. Hence, we will give just a brief overview of some of the main characteristics of the human auditory system, so we can refer to them later in the work.

On the Transduction of Sound from Mechanical Energy into Bio-electrical Signals and Beyond

Sounds of particular interest typically interfere with other rather distracting sounds. The auditory system has the remarkable ability to disentangle various simultaneous sounds and selectively focus on a single, particular sound [50]. This phenomenon and significant signal processing challenge is known as the "cocktail party problem" [179]. Sound, in general, is defined in [1] as:

> *a mechanical wave that is an oscillation of pressure transmitted through a solid, liquid, or gas (or plasma), composed of frequencies within the range of hearing.*

Therefore, a sound is a vibration of pressure in a medium, usually generated by a vibrating object. If this medium is in direct contact with the eardrum, such vibrations might cause acoustic sensations. The mechanical structure of the inner ear maps sound frequencies onto different positions of the **basilar membrane** within the **cochlea** [27]. The flexibility and width of the membrane increases with distance from the **oval window**, which is the entry point of sound. As a result, it produces a peak of vibration near the oval window for high frequencies. For a low frequency, such a peak of vibration will be closer to the far end of the cochlea [490] (see figure 2.1). Hence, the cochlea, in many ways similar to the Fourier transform [222], decomposes mixtures of acoustical frequencies into their components, mapping them onto different spatial locations within itself. The thousands

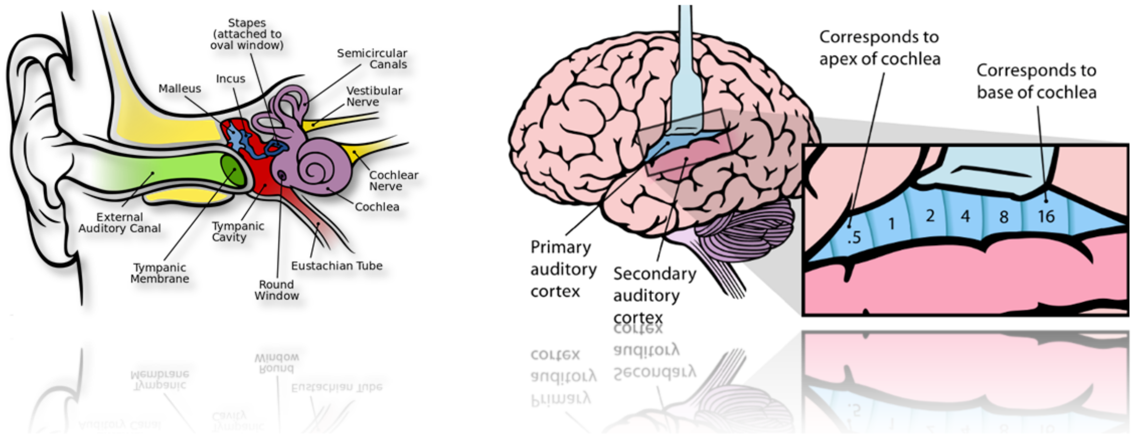


Figure 2.1: Left: The mechanical structure of the inner ear maps sound frequencies onto different positions of the basilar membrane within the cochlea. Right: Each receptor cell in the cochlea has a neuronal "cable" to the cochlear nuclei and even the auditory cortex, where one can find a complete topographical map of the audible frequency spectrum. Pictures modified from [77]

of so called **mechano-receptors** that are distributed along the basilar membrane do not need be tuned to different frequencies, other than in color vision, where each receptor type responds in a particular wavelength range (see appendix B.1). The position of each mechano-receptor determines which sound frequency will arouse it ([490]; [287]).

On a higher processing level, all fibers in the **auditory cranial nerve** send the information, in parallel, from the receptor cells to the brain. Thus, each receptor cell has its own neuronal "cable" to the **cochlear nuclei** and even the auditory cortex, where one can find a complete topographical map of the audible frequency spectrum (see figure 2.1). This topographical map of audible frequencies mirrors the mapping of frequencies in the cochlea [371].

Pitch

The perception of pitch is arranged along a single dimension, just as on a piano keyboard. Due to the parallel processing of receptor information from the cochlea, mixtures of different frequencies can be analyzed accurately, unless mixtures become too complex. Hence, the tones that a chord is made up of can be identified [460]. In case where mixtures have unique properties, such as in the case of the strengths of harmonic overtones over some fundamental frequency that distinguish an "A" tone produced by a piano from that of a guitar one can still identify the fundamental, whose perceived pitch is not effected by the overtones. Thus, a mixture of 440 Hz and 880 Hz would not be perceived as an intermediate frequency, e.g. 660 Hz.

Loudness

Beside pitch, the auditory system is sensitive to a very large range of sound levels, measured based on loudness. Loudness is the characteristic of a sound that psychologically correlates with the physical strength or amplitude of this sound. More formally, it has been defined in [220] as

> *the attribute of auditory sensation in terms of which sounds can be ordered on a scale extending from quiet to loud.*

Thus, it is more of a subjective measure and is often confused with more objective measures such as **sound pressure level (SPL)**. The perception of loudness of a sound depends on both, the pressure (i.e. its sound pressure level) and its frequency spectrum. For sounds such as a pure tone, equal-loudness-level curves can be defined that represent the sound pressure levels of a sound (see figure 2.2 (left)). These curves reveal that perception of loudness is not equal for all frequencies even when sound pressure level is constant and are considered to reveal the frequency characteristics of the human auditory system [452]. The perception of loudness is further dependent on duration of a certain sound, at least below durations of about one second. Up to a few hundred milliseconds, the longer the sound, the louder it is perceived ([360]; [455]).

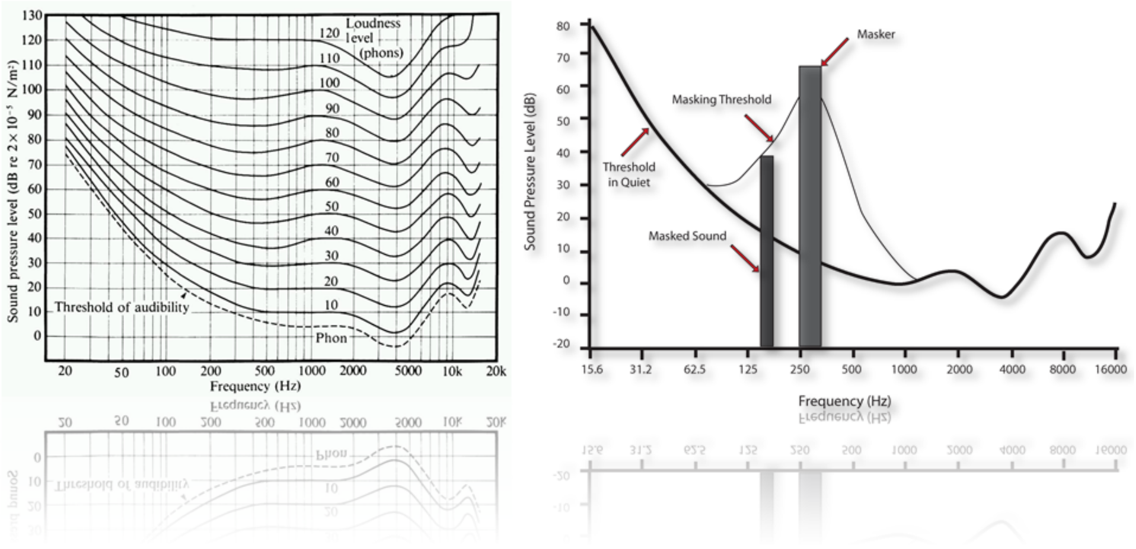


Figure 2.2: Left: The equal loudness curves reveal that perception of loudness is not equal for all frequencies even when sound pressure level is constant. Picture modified from [452]. Right: An illustration of one sound masking another

Auditory Masking

Additionally, auditory masking can occur when the perception of one sound is affected by the presence of another sound, as illustrated in figure 2.2 (right). Due to the properties of the basilar membrane, it is, however, asymmetrical, meaning that low-frequency sounds mask high-frequency sounds much more efficiently than the reverse [179].

Temporal and Spectral Variation

There has been numerous research on the remarkable auditory sensitivity of the brain that allow developing stable representations for auditory events, given the varied, and often ambiguous, temporal patterning of acoustic spectral content received by the ears ([89]; [231]; [232]; [224]; [356]; [529]; [363]). As an example, the auditory system is capable of recognizing gaps within broadband noise stimuli as short as only 2 to 3 milliseconds [363]. Research on the response of human auditory cortex to spectral and temporal variation indicates that the core auditory cortex in both hemispheres respond to temporal variation, whereas the anterior superior temporal areas bilaterally respond to the spectral variation ([529]; [418]). It further revealed that responses to temporal varying test stimuli are weighted towards the left, while responses to spectral variations are weighted towards the right, as illustrated in figure 2.3. Zatorre and Belin [529] thus infer that those findings confirm a specialization of the left-hemisphere auditory cortex for rapid temporal processing and also indicate a complementary hemispheric specialization of the right-hemisphere cortical areas for spectral processing.

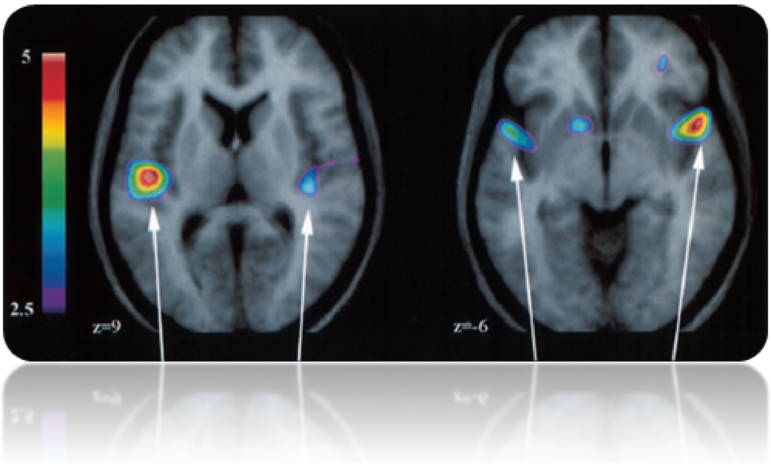


Figure 2.3: Left: A horizontal section taken through the region of **Heschl's gyri** that shows significantly greater activity to stimuli varying temporally (left) than spectrally (right). Right: A section taken through the **anterior superior temporal region**, which showed a greater response to the spectral variations than to temporal. Picture modified from [529]

Sound Localization

The brain harnesses subtle differences in sound characteristics, such as pressure and spectral characteristics to determine the location of a specific sound source [35]. It makes use of **binaural** cues, such as to measure the difference in arrival times between the ears or the relative amplitude of high-frequency sounds [388]. On the other hand it uses **monoaural** methods that depend on the asymmetrical spectral reflections from and filtering effects of various parts of our bodies, including torso, shoulders, and pinnae, summarized as the **head-related transfer function**. Thus, depending on the angle from which they strike those “filters”, sounds are frequency filtered individually. The most significant filtering cue for sound localization is the **pinna notch**. The filtering effect results from destructive interference of waves reflected from the outer ear. Depending on the angle from which the sound strikes the outer ear, a specific frequency is selectively filtered [388].

Chapter 3

Sound Computation

This chapter briefly surveys the main algorithms and techniques for digital sound synthesis, especially those that are employed throughout our own work, as related to auditory display as well as how sound is represented in digital computers. For a more in-depth analysis of sonification methods see ([388]; [403]; [316]).

3.1 Sound Synthesis

Additive Synthesis

In Additive Synthesis [403], being the oldest sound synthesis technique, each partial is modeled by a separate sinusoidal oscillator with a specific frequency f_i and amplitude envelope $a_i(t)$. Using a time variant function $f_i(t)$ for the frequency allows continuous changes in pitch. The output of those oscillators is superimposed to produce a composite signal $s(t)$:

$$s(t) = \sum_i a_i(t) \sin(2\pi f_i t + \phi_i) \tag{3.1}$$

The sine function is used as a building block for the signal $s(t)$. Within practical implementations a continuous sine computation can be avoided and replaced by a table lookup and interpolation procedure. A table holds a period of a periodic signal, called **waveform**. Such a **table lookup oscillator** is used everywhere in computer music [318] and in many commercial synthesizers. As a natural generalization, Additive Synthesis can be performed with arbitrary waveforms using **table-lookup synthesis**. A great benefit offered through Additive Synthesis is that one gains an obvious and direct control over the resulting acoustical signal, due to the linear mapping from control parameters to sound.

Subtractive Synthesis

Subtractive Synthesis ([318]; [403]) is quite complementary to Additive Synthesis, as it shapes a spectral form of sound by filtering out undesired parts from a complex input signal, rather than building up a complex sound by adding spectrally simple parts. However, whereas additive models have difficulties in creating noise, it can be easily introduced in Subtractive Synthesis by using noisy excitation signals. A desired waveform is produced by applying time variant filter to an input signal. Hence, in Subtractive Synthesis, the two main parts are a complex source sound and a filter. Both components usually have time varying parameters. As such a distinction is found in a lot of physical systems, Subtractive Synthesis can be seen as a special case of Physical Modeling, which will be discussed below.

FM Synthesis

Frequency modulation (FM) for sound synthesis [403] was first introduced by Chowning [78]. The signal $s(t)$ is computed as:

$$s(t) = \sin(2\pi f_c t + I(t)\sin(2\pi f_m t)) \tag{3.2}$$

Thus, FM relies on modulating the frequency of a simple periodic waveform with another simple periodic waveform, as illustrated in figure 3.1(left). The frequency of a **modulator** f_m is in the same order of magnitude as a **carrier** frequency f_c and usually takes a fixed multiple $f_m = \lambda * f_c$. For a constant timbre, both the ratio $\lambda = \frac{f_m}{f_c}$ and the modulation index $I(t)$ are kept constant. A frequency analysis (see figure 3.1 (right)) of the signal $s(t)$ shows that frequency modulation basically takes energy from the carrier frequency, spreading it to the side band components at frequencies $f_{1,2}^k = f_c \pm k f_m$, for each k^{th} side band. The amplitudes of each k^{th} side-band are determined by “Bessel functions of the first kind and k^{th} order” $J_k(I)$ [78].The larger the k, the higher I has to be for that side band to have significant amplitude. “Negative side band frequencies”, as illustrated in figure 3.1(right) occur for higher orders of k. These frequency components’ phase has been shifted by π, which leads to a sign change and a “reflection at the origin”. Often λ is chosen to be either an integer or a small integer fraction. $\lambda = 1$ results in sounds that contain frequencies which are integer multiples of f_c. Thus, FM synthesis delivers a very simple model for creating a complex timbre using a single scalar parameter I. However, unlike additive analysis, a small change in the input parameter, i.e. I, does not necessarily result in an equivalent small change in the created sound structure. Specific modulation functions $I(t)$ allow to mimic several instruments, e.g. brass instruments with $I(t)$ starting at 0 and rising to some final value I_{max} within a constant attack time. Several classes of timbre can be implemented using other modulation ratios λ. $\lambda = 1.414$, $\lambda = 2$ or $\lambda = 3$ produces bell-like sounds. $\lambda = 2$ results in a more organ alike sound [389].

Non-Linear Synthesis

The previous synthesis techniques contained the possibility of using a separate control of amplitude envelope and timbre. However, in most real instruments these two attributes

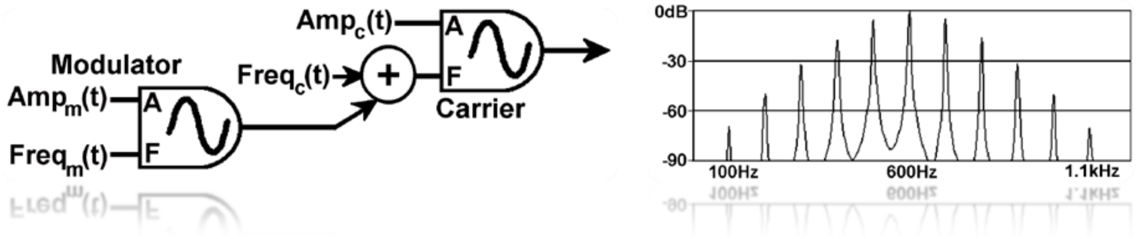


Figure 3.1: Left: A simple FM synthesis circuit with one carrier and one modulator sine wave. Right: Frequency spectrum of a simple FM synthesis with $f_c = 600Hz$, $f_m = 100Hz$ and $I = 3$. Pictures modified from [81]

are closely interrelated. As an example, the intensity when blowing a flute will determine both changes in timbre and the amplitude of the sound at the same time. Non-linear Synthesis [388] can be employed to simulate such a behavior using a nonlinear transfer function $g(.)$ to manipulate the instantaneous amplitude of a source signal $f(t)$:

$$s(t) = g(f(t)) \tag{3.3}$$

Granular Synthesis

Granular Synthesis [388] composes a more complex waveform from the superposition of thousands of very short so called acoustic “grains”. According to Gabor’s theory of “Acoustical Quanta and the Theory of Hearing” theory [97], such a granular representation is meant to describe larger complex sounds. Each grain’s spectral property determines a specific set of control parameters, whereas a temporal organization is controlled by the composition of grains:

$$s(t) = \sum_i a_i g(t - t_i \theta_i) \tag{3.4}$$

where g is the time domain representation of a single grain, whose shape would be formed by a function of further parameters θ. Grains are usually of durations of about 20 to 50 milliseconds.

Physical Modeling

Physical Modeling [81] is about simulating the essential parts of real physical instruments in order to create a similar sound and sound control in the model. As an example in an acoustic guitar the plucking position determines the resulting string sound. Implementing such "high-level" controls in other synthesis models would demand extensive parameter modifications. In contrast, in a physical model of the string, the plucking position can be controlled directly. There have been several approaches to Physical Modeling such as **Spring Mesh Models** [437]. Though quite intuitive, the more complex the system becomes, those models become computationally infeasible. Another approach is **Digital Waveguides (DWG)** [436], a technique suitable to model traveling waves as sound waves in the air that can be applied to model many physical system, e.g., the human vocal tract, wind instruments or string instruments. Generally Digital Waveguides are derived from the **wave equation** [319] that completely describes the motions of an ideal (without any damping or stiffness) string under tension:

$$\frac{\partial^2 y}{\partial t} = c^2 \frac{\partial^2 y}{\partial x} \tag{3.5}$$

Thus, the upward and downward acceleration at any position on the string is equal to a constant multiplied by the curvature of the string at that position. The constant c is the speed of wave motion on the string. It is proportional to the square root of the string tension and inversely proportional to the square root of the mass per unit length. The same equation can be applied to describe the flow of air within a cylindrical acoustic tube (such as a trombone, clarinet bore, or human vocal tract), except for the displacement y to be replaced by the air pressure P. This equation can be re-formulated as:

$$y(x,t) = y_l(t + \frac{x}{c}) + y_r(t - \frac{x}{c}) \tag{3.6}$$

It basically states that vibrations of the string can be represented as the combination of two separate traveling waves, one to the left y_l and another to the right y_r. Both move at rate c which refers to the velocity of the sound propagation on the string. For an ideal string as well as ideally rigid boundaries at both ends the wave reflects with an inversion at each end and travels back and forth indefinitely. Hence, such a simple Waveguide filter can be implemented using two **delay lines** ([436]; [243]) which model the propagation of left and right going traveling waves. Additionally, simple physical models can be extended to consider stiffness and non-linear interaction elements [438], such as bowing friction, can be utilized to model system components such as the mouthpiece of a clarinet or the bow of a violin. This non-linearity turns the steady linear motion of a bow into an oscillation of the string , and thus makes the sound to become more realistic.

3.2 Noise

Noise of whatever kind would rather be the product of, e.g., Subtractive Synthesis than a sound synthesis method itself. It is, however, briefly mentioned within this chapter as specifically pre-recorded Brown Noise will be used later within the work as a means to sonify certain information about the image.

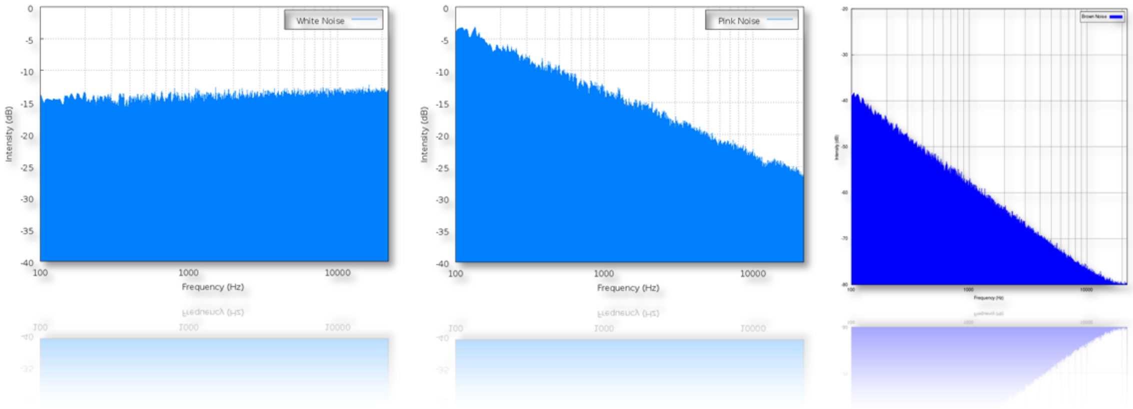


Figure 3.2: The frequency spectra of (left) White Noise, (middle) Pink Noise and (right) Brown Noise. Intensities (in dB) on the y-axis and frequencies on the x-axis

White Noise

The term "White Noise" arises from the analogy to white light, where the power spectral density $S(f)$ is distributed over the visible band so that all three color receptors of the human eye are approximately equally stimulated [107]. It is defined in [58] as: *a stationary random process having a constant spectral density function.* In other words, White Noise contains equal power within a fixed bandwidth at any given center frequency f_c. Therefore, $S(f) = S_0 = const.$

Pink Noise

Pink Noise, also known as "flicker" or $1/f$ Noise, describes a signal with a frequency spectrum whose power spectral density $S(f)$ is inversely proportional to the frequencies. Mathematically speaking, $S(f) \propto 1/f$. Each octave carries an equal amount of power. $1/f$ noise falls off at 3 decibel (dB) per octave. Its name originates from the pink appearance of visible light with such a distributed power spectrum [118]. Pink Noise can be found quite frequently in nature. It is present in the electromagnetic radiation output of some astronomical bodies. In biological systems, it was found in neural activity, heart beat rhythms as well as the statistics of DNA sequences ([15]; [234]).

Brownian Noise

In science, “Brownian Noise” or “Brown Noise”, is a signal produced by Brownian motion [57], hence it is often referred to as random walk noise and, therefore, the name Brown Noise. As the spectral density of Brownian Noise is inversely proportional to f^2 it has more energy at lower frequencies, decreasing in power by 6 dB per octave. Its spectrum is given by $S(f) = S_0^2/f^2$. Brownian Noise has specific “damped” or “soft” sounding properties compared to White and Pink Noise, such as a low roar resembling a waterfall or heavy rainfall.

3.3 Representation of Analog Sound Signals in Digital Computers

As stated in section 1.2, sound is essentially pressure variations within a specific medium. Therefore, it suffices to measure its time variant sound pressure signal function $s(t)$ to store a full representation of any given sound. To record the analogue signal $s(t)$ with any digital recording device, it is sampled at some specific sampling rate f_s, at equidistant positions in time t_s, so that $f_s = \frac{1}{t_s}$. The sampling rate or sampling frequency f_s determines the upper limit for recordable frequencies to be reconstructed properly, according to **Nyquist-Shannon theorem** ([428]; [226]; [298]), which states:

> *If a function $s(t)$ contains no frequencies higher than B hertz, it is completely determined by giving its ordinates at a series of points spaced $\frac{1}{2B}$ seconds apart*

Due to the bit resolution of employed data format, values at equidistant time positions t_s are quantized. The distortion of the signal, as a result of such a quantization, is audible as quantization noise and the Signal-to-Noise ratio (SNR) of this quantization noise, measured in decibel dB, is proportional to the number of bits per value. For instance, in recording audio for an audio CD a sampling rate of 44100 Hz and a sample data format of 16 bit integers are used, yielding a SNR of about 96 dB. The common representation of discrete sound signals in the context of digital output devices is that of a series of sample frames $\{s_1, s_1, ...\}$. A single frame s_i contains all quantized sample values for all available audio channels. Such a multi-channel audio signal can then be stored frame by frame within a computer's memory in standard floating point number format.

Part II

A Modular Computer Vision Sonification Model

Chapter 4

Introduction

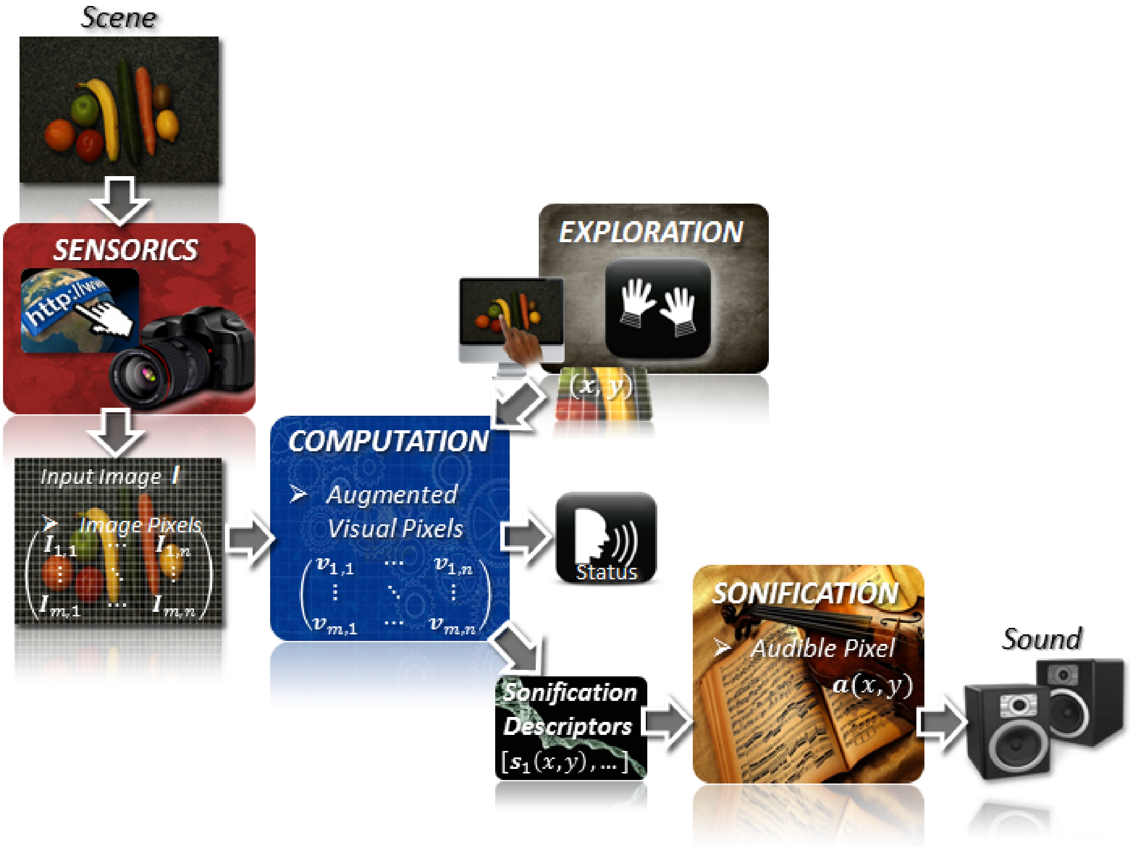


Figure 4.1: The Modular Computer Vision Sonification Model

Figure 4.1 gives an overview of the general concept of the **Modular Computer Vision Sonification Model**, with the components **sensorics - exploration - computation - sonification**, and a specific setup that implements this concept and that we present and evaluate below. As the notation "computer vision" implies, we focus upon working with visual data. More abstract models for the process of data sonification in general, as illustrated in figure 1.1, have already been formulized by others, including ([203]; [205]). There has also been some emphasize on the importance and possibilities of interaction in sonification ([217]; [386]; [204]; [218]). Designing such a modular computer vision sonification model we face the following challenges:

- Our sonification has to be intuitive enough to allow a blind and especially congenital blind person to quickly learn to understand the concept of colors and textures as well as their audification to be able to recognize objects and interpret images. F
- Our system shall be able to analyze images that users may find in a photo collection, on the internet, or captured with a still camera.
- We need to find appropriate visual descriptors that represent particular image values. They should be informative, general and stable under transformations such as illumination and pose.
- We have to define a way to sonify these image descriptors. The goal would be to convey as much information as possible without distracting interference of the acoustical signal as well as to give an auditory perception that enables users to develop an “intuition” about the visual data.
- We want to develop an “exploration paradigm”. Human vision has many aspects of parallel processing. Since much of the visual pathway transmits information from different parts of the visual field in parallel, and pre-attentive vision (pop-out effects) indicates parallel processing on higher levels and in contrast, an auditory signal is mostly a sequential data stream. This implies that it is hard to map an entire image to a single and constant auditory signal. Therefore, we decided that users should explore the image locally, using a direct finger based touch device.

4.1 The Visual to Acoustical Processing Pipeline

Figure 4.1 gives a brief overview over the processing pipeline of the modular computer vision sonification model. Stage one is the acquisition of a rasterized input image $\boldsymbol{I}$. For each color pixel $I(x, y)$ of the input image $\boldsymbol{I}$, every information that could be sonified is pre-computed by a set of computer vision algorithms and stored in a data-structure called **augmented visual pixel** $\boldsymbol{v}(x, y)$. Meanwhile, the user is kept up to date about the program status using speech output.

During interactive exploration, based on the user's current position (x, y) and his selection which of the features he wants to be sonified, the appropriate elements from $\boldsymbol{v}(x, y)$ are copied into an individual **sonification descriptor** $\boldsymbol{s}(x, y)$. A sequence of those sonification descriptors for all pixels on the exploration trajectory are added to a **queue structure** ([249]; [85]). Such a queue structure makes sure that no pixels are skipped even for fast motions. Furthermore, in part IV it ensures a smooth amplitude transition during color-sound synthesis, as will be explained in section 12.1. In real-time or buffered with a slight delay, the sonification module finally processes all sonification descriptors from the queue, turning all elements of each $\boldsymbol{s}(x, y)$ into a complex and internally synthesized sound, that we call **audible pixel** $\boldsymbol{a}(x, y)$.

$$\boldsymbol{I}(x, y) \rightarrow \begin{pmatrix} \boldsymbol{v}_h(x, y) \\ \boldsymbol{v}_s(x, y) \\ \vdots \end{pmatrix} \rightarrow \begin{pmatrix} \boldsymbol{s}_h(x, y) \\ \boldsymbol{s}_s(x, y) \\ \vdots \end{pmatrix} \rightarrow \boldsymbol{Queue} \rightarrow \boldsymbol{a}(x, y) \tag{4.1}$$

Chapter 5

Modules

5.1 Sensorics

The sensory module acquires the data to be sonified. The actual implementations of the system presented in this thesis, in parts III and IV, rely on still images that are available as files taken directly from a camera or the internet. The reason why we did not work with video material is due to the fact that our system does not strive for applications in real-time navigation. Nevertheless, we were able to implement our algorithms, including computer vision and machine learning techniques, to nearly comply with real-time expectations.

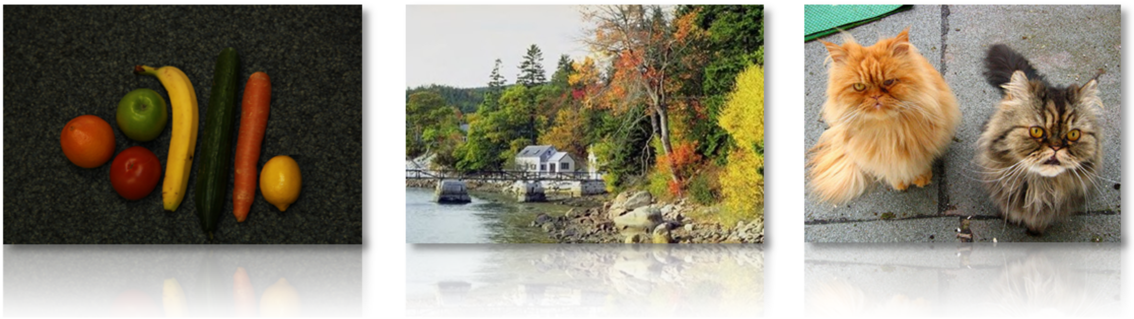


Figure 5.1: The different input image data sets we have been working with in various parts of the work. Left: Pictures we work with in part III were taken by ourselves. Middle: Pictures from the COREL Database for Content based Image Retrieval [325]. Right: Images taken from the PASCAL Visual Object Classes Challenge [135]. Both data sets (middle and right) are employed in part IV

The image data sets we work with in part III have been created by us (see figure 5.1 (left)). The image sizes are about 1920 × 1080 pixels. The data sets employed in part IV were taken from the **COREL Database for Content Based Image Retrieval** [325] (see figure 5.1 (middle)) with image sizes of 384×256 pixels, and the **Visual Object Classes Challenge (VOC)** [135] (see figure 5.1 (right)), about 500 × 400 pixels each.

5.2 Computation

The computation module pre-computes the augmented visual pixels $v(x, y)$ based on the input image pixels $I(x, y)$. The particular implementation of this module depends on the specific requirements that an application has. In this thesis, we present two possible applications and implementations:

- **Low-Level Feature Image Analysis:** For the task of "auditory object recognition" in part III, we focus on fundamental characteristics such as colors, roughness and edges of various orientations.
- **Multi-Level Image Analysis:** To allow for more sophisticated usage, such as "auditory image understanding" in part IV, computer vision and machine learning algorithms are leveraged to derive and later sonify image information on many levels, ranging from low-level color information to high-level object recognition.

Within both applications, the computation module is responsible to pre-process acquired images in terms of filtering. Although sophisticated methods, such as **edge preserving filtering** ([464]; [353]; [21]) are already applied, the additional employment of algorithms for specific tasks, such as **shadow removal** ([406]; [146]; [145]; [518]) or **specular highlight removal** ([457]; [253]; [431]) is relinquished. We deliberately refrain from incorporating those approaches, as they, despite the remarkable results of some of the referenced approaches, tend to introduce artifacts or unwanted changes in color.

5.3 Exploration

The exploration module enables the user to navigate within an image, consecutively generating various finger positions (x, y) that trigger the computation module to copy the appropriate elements of augmented visual pixels $\boldsymbol{v}(x, y)$ into an individual sonification descriptor $\boldsymbol{s}(x, y)$ each, and append it to the queue of sonification descriptors.

A Bas-Relief Inspired Exploration Paradigm

Human vision has many aspects of parallel processing: Much of the visual pathway in transmits information from different parts of the visual field in parallel, and pre-attentive vision (pop-out effects) indicates parallel processing on higher levels. In contrast, an auditory signal mostly is a sequential data stream. This implies that it is hard to map an entire image to a single and constant auditory signal. Therefore, we decided that users should explore the image locally. Very much like a blind person who explores a Braille text or a bas-relief image haptically with the tip of her finger, our users touch the image (via a touch pad or touch screen) and experience the local properties of the image as auditory feedback. In the analogy of a blind user exploring a bas-relief at a specific position, he would gain additional information about the direct environment of that position due to the thickness of his finger tip. Accordingly, we simulate this characteristic by taking a specific area around each pixel position (x, y) into account during the computation of augmented visual pixels $\boldsymbol{v}(x, y)$.

An Evaluation of Interface Technologies

Navigating within an image requires an appropriate interface. The computer-mouse, which is popular among users with normal vision, drops out as it does not deliver any absolute coordinates, which are necessary for a blind user to know the position in the image. Therefore, we worked with several interfaces to find out what suits best to a blind person.

- **Pen-Tablet:** The pen tablet interaction method (see figure 5.2) functioned far better than the mouse, as it can be set to absolute coordinates. However, it turned out, that a direct touch helps to orient within the flat image, as analogous to moving the tip of the finger along a relief.

- **Touch Screen:** For training and several user studies in chapters 9 and 13 we utilized a touch screen device (see figure 5.2 (right)) that allows the user to interact directly with the image plane without seeing the image. The image is rendered within the midst of the screen. The given absolute positioning and direct touch experience proved very successful.

- **Finger Tracking:** We experimented with cheaper interaction method opposed to the touch screen, as it might not make sense for a blind person to buy a touchable screen only to serve as an interface and not have any further use of it. The experimental system is based on a camera-based finger position tracking using on the ARToolKitPlus pose tracking system [496]. First, a camera is to acquire the visual data to be explored. After pre-computation the aqcuired image, the same camera is utilized to detect a marker (see figure 5.3 (left)), attached to the user's fingernail (see figure 5.3 (middle)), which is thereafter calculated back to estimate the fingers position within the image (see figure 5.3 (right)).

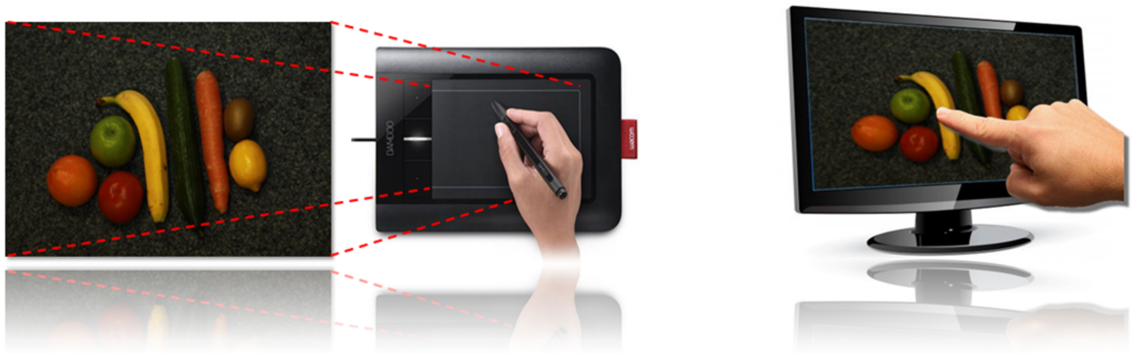


Figure 5.2: Left: Pen to tablet interface with absolute coordinates, Right: Touch screen working directly on the acquired image

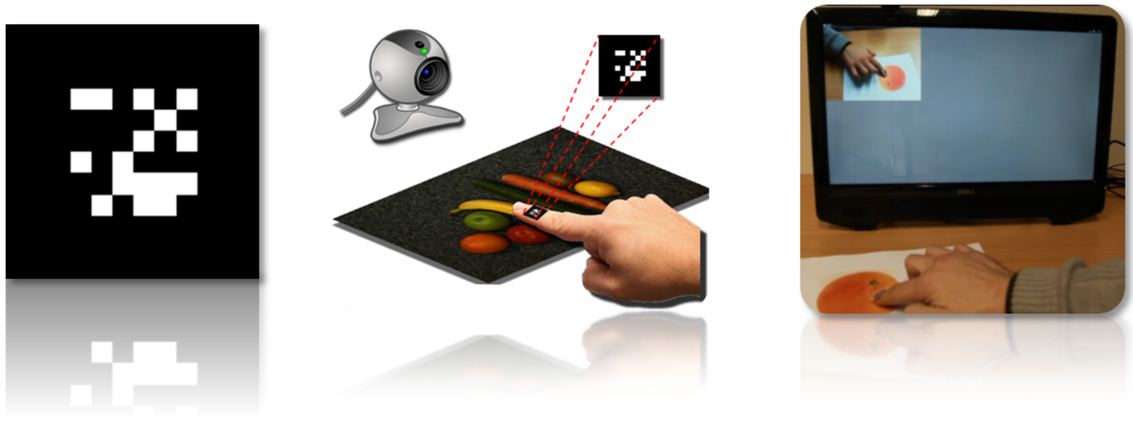


Figure 5.3: Left: A typical "marker" to be tracked. Middle: Finger position (x, y) is tracked using a marker system and estimated within the image. Right: The system in process

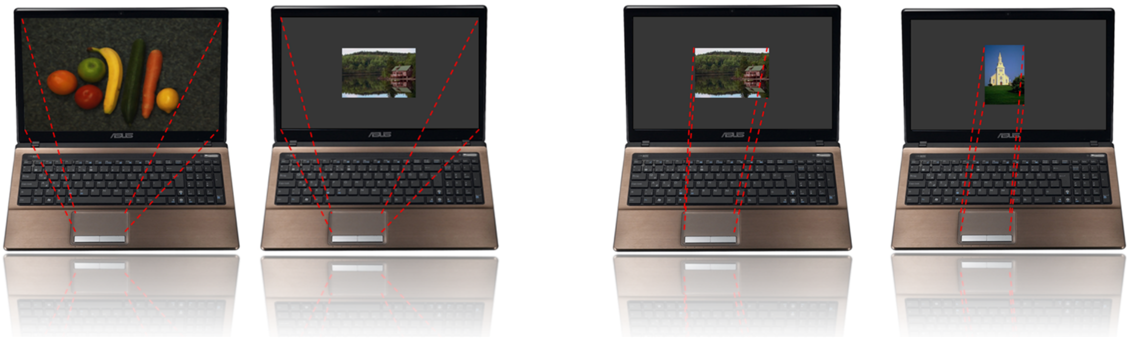


Figure 5.4: A common notebook's touch pad switched to absolute coordinates. Left: Image rendered in the center of the touch pad. Right: touch pad mapped to the region of the image

The Touch pad

Common notebooks are equipped with regular touch pads. Those usually provide only relative positioning, similar to a computer-mouse, which is why they are not used by visually impaired persons and why they are generally inappropriate to be used for image exploration. If, however, absolute coordinate positioning could be enabled, a touch pad might be an appropriate interface after all.
Accessing absolute finger positions from a touch pad directly within an application is not possible in general. Fortunately, those machines, equipped with a Synaptics touch pad, allow to access these absolute coordinates, based on their System Development Kit (SDK) [453]. As a lot of modern notebook computers are equipped with touch pads of this manufacturer, this interface might be an appropriate alternative to the touch screen.
We propose to render the image within the center of the touch pad, as visualized in figure 5.4 (left). An alternative would be to map the touch pad to the region of the image and is illustrated in figure 5.4 (right). Although this method often provides a higher spatial (explorative) resolution of the actual image, it comes with the significant drawback of possible distortions of the original image's geometry. For instance, the rightmost picture in figure 5.4 shows the "distortion" of the upright image. So, upright structures are compressed.

Note that our considerations concerning the usage of a regular Notebook's touchpad are only theoretical, as our available test machine, unfortunately, was not equipped with a Synaptics touch pad, and thus, we were not able to perform experimental tests.
However, regular touchpads have a significant drawback, which is, that their overall area, which can be used to represent an image, is extremely small.

Tablet PCs

Due to our experiments, modern tablet PCs seem to be the most suitable device for using our application concerning usability, mobility and performance, as they combine immense computational power with an appropriate screen size and high class touchscreen technology and can be taken along almost like any mobile phone.
Thus, as discussed in chapter 14, our system is used on a *Microsoft* 7 based Samsung Slate 7 tablet PC (figure 5.5) which is sold for about 500 Euro and allows for traditional third-party applications to run, even sophisticated 3D engines, such as the Unreal engine 3 [133]. Its integrated front-camera and touch screen make it ideally suited to be taken along by visually impaired people on, e.g., hiking tours, to grasp, especially, scenes that are "further away".

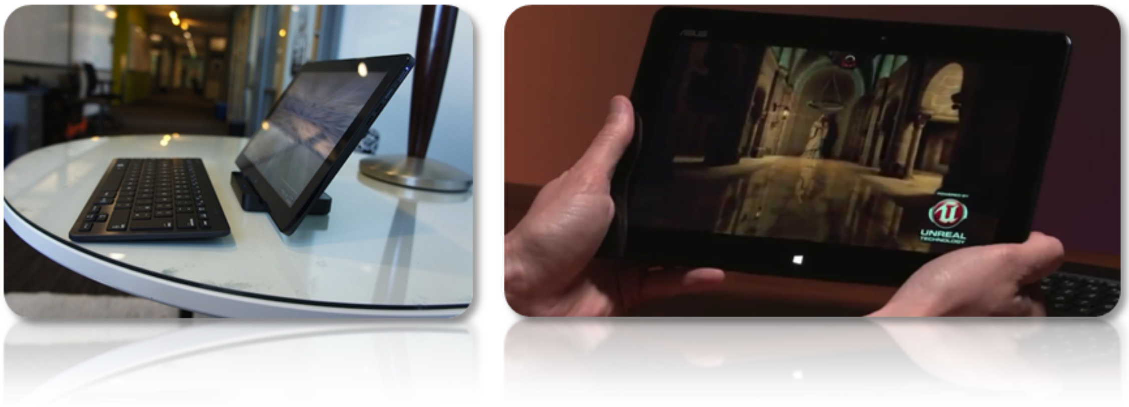


Figure 5.5: Left: The Samsung Slate 7 tablet PC. The Samsung Slate 7 running the Unreal 3D graphics engine [133]

5.4 Sonification

The sonification module sequentially processes all sonification descriptors $\boldsymbol{s}(x, y)$ from the queue, turning all its elements into a complex sound, the audible pixel $\boldsymbol{a}(x, y)$, as opposed to the augmented visual pixel $\boldsymbol{v}(x, y)$. Our sonification techniques in both part III and IV, are, therefore, a realization of Parameter Mapping Sonification, as presented in section 1.3.

A great challenge with sonification is to avoid conflicting signals and information overload, as well as the transformation of quasi-static two dimensional image data into a dynamic audio stream. Though humans can distinguish many attributes such as pitch, volume, amplitude envelope, timbre, roughness, vibrato or tremolo as discussed in section 2.1, it is still impossible to transport all potential descriptors of visual information simultaneously. Thus, our actual implementation of this module is driven by 3 demands:

- **Real-Time Exploration Driven Sound Synthesis:** We want the user, as previously described, to fully interact with the visual data in real-time and to be able to hear what is currently under his finger, unlike approaches that sonify a whole image sequentially, e.g. by scanning its pixels row by row as in [524].

- **Non-Conflicting Simultaneous Signals:** We want a method to simultaneously sonify features such as color and multi-level image features, instead of focusing all acoustical attributes on a single feature such as the progression of edges.

- **Aesthetics:** The sonification model should meet aesthetic demands that are important for comfortable and extensive usage (as discussed in section 1.1).

In this thesis we present two slightly varying approaches to meet those demands:

- **Color & Low-Level Feature Sonification:** For the task of auditory object recognition in part III, the focus is on the sonification of fundamental characteristics, such as colors, roughness and edges of various orientations. MIDI instruments are applied and combined to express those features.

- **Multi-Level Sonification:** For the task of auditory image understanding in part IV, image information on many levels, ranging from low-level color information to high-level object recognition is sonified. To accomplish, a fundamental sound synthesis combination of elementary sound characteristics is performed.

A Queue of Sonification Descriptors

To build up a queue of sonification descriptors $\boldsymbol{s}(x, y)$ that consecutively adds up all pixels on the exploration trajectory, it is necessary to ensure no pixel to be skipped even for fast motions. As in depth discussed in section 14.4, it will be further crucial in part IV for parallel real-time exploration and sonification in general to allow the system to continuously write and process variables at the same time. Other than in part III, where such real-time parallelism is handled by the external MIDI synthesizer, it has to be implemented from scratch in part IV.

Part III

Auditory Object Recognition

Chapter 6

Background & Objectives

The system presented in this part is mostly dedicated to scrutinize the possibilities and limitations of recognizing objects through sound. To provide a fundamental basis for such an "auditory object recognition", in this part, we refrain from employing complicated object recognition algorithms. We rather extract very fundamental image characteristics, such as edges of various orientations, called **orientation maps**, as well as a measure of roughness which are then sonified simultaneously. Figure 6.1 gives an overview of the specific implementations of the computation (see chapter 7) and sonification module (see chapter 8) of our modular computer vision sonification model, proposed in part II, that need to be performed. Beside low-level image characteristics such as orientation maps or roughness, in chapter 8, a color sonification concept is proposed, which will be refined in part IV of the thesis. The sonification of colors is important to us for 3 reasons:

- **Understanding Colors:** Color sonification can be used to offer a way to congenital blind people, who have never had any encounter with colors, to understand colors and to be able to communicate with normal sighted people about such fundamental quality of human vision.

- **Object Recognition:** In illuminated environments, colors are crucial in the process of detecting objects.

- **The Art of Color**: Color itself, without any useful purposes such as enhancing object recognition capabilities, can be seen as a fascinating entity of life, which is worth to be conveyed to the blind, using alternative ways. In the words of psychologist Ulrich Beer ([26]; [300]):

  > *Seldom, surely, is the psychological part of an appearance in nature so great as it is in the case of color. No one can encounter it and stay neutral. We are immediately, instinctively, and emotionally moved. We have sympathy or antipathy, pleasure or disapproval within us as soon as we perceive colors.*

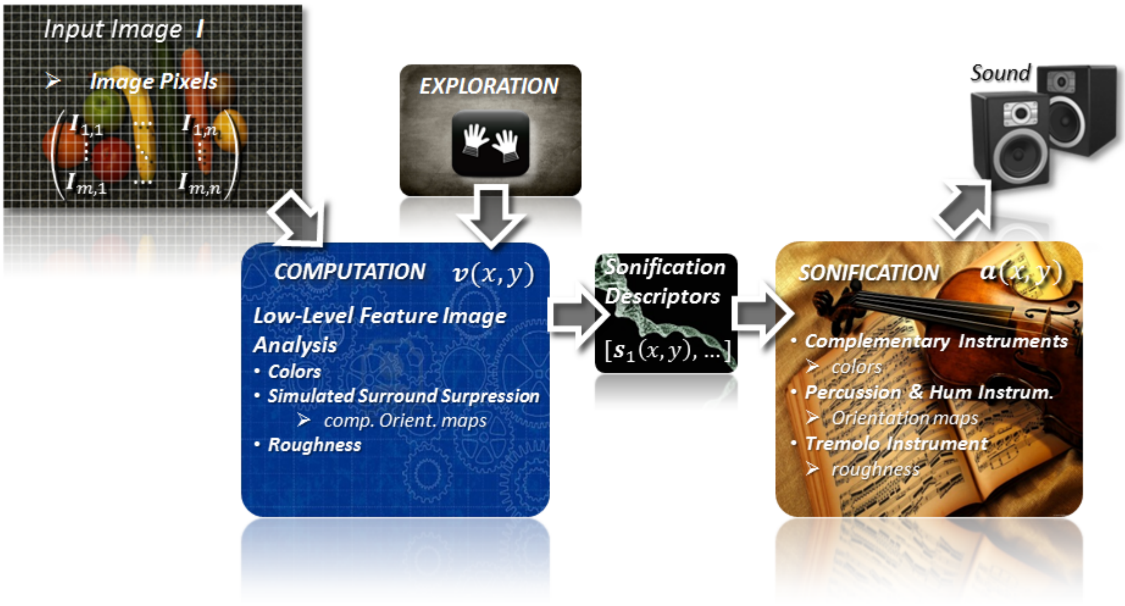


Figure 6.1: Implementations of the computation and sonification modules of the Modular Computer Vision Sonification Model for the task of auditory object recognition

The proposed approach proves to be intuitive enough not only to be understood and applied by congenital blind, but also to help convey the concept of colors and color mixing itself (see chapter 9). Furthermore, apart from object recognition, the sonification of the orientation of edges gave hope to be of great benefit to develop the spacial understanding, as discussed in section 1.5, of congenital blind persons.

Training and testing with the proposed system were performed with two groups of participants, as described in detail in chapter 9. First, a congenitally blind, 54 year old academic who had acquired a geometric understanding and spatial sense throughout his life due to special training. Then, we had a group of congenitally blind, 14-year-old teenagers, living at a local residential school for visually impaired[4], work with our system. Unlike the adult participant, they had little geometric understanding and sense of space. We performed an experimental evaluation of our system to measure their progress and compare it with the results of the adult participant.

The hope is that our system can not only support them in everyday life, but also help them to develop cognitive abilities in geometry and spatial orientation[5].

[4] Internat des Rheinischen Blindenfürsorgeverein 1886 Düren

[5] The work on “auditory object recognition” has been published in [18]

Chapter 7

Low-Level Feature Image Analysis

7.1 Color Information & Image Pre-Processing

Different color systems with several motivational backgrounds are discussed in appendix B.2. The **RGB** model [286] uses additive mixtures of red, green and blue. It is motivated by the human eye receptors [174] (see appendix B.1) and applied, e.g. in many display devices. However, providing a non-visual access to colors, which is the case in this thesis, requires a more intuitive system, especially for congenital blind people. This is why we prefer the **HSL** model ([286]; [441]), where each color value is described by hue h, saturation s and lightness l.

Edge Preserving Filtering

What makes color sonification difficult is the fact that color values often change rapidly from pixel to pixel even if there are only minute variations in textures and materials. Often, the reason is image noise produced by the camera. It is obvious that such changes clearly overburden a blind user. Therefore we smooth the image patch around the pixel position (x, y) based on **bilateral filtering** ([464]; [353]; [125]; [21]). Other than **Gaussian filtering** [222], bilateral filtering not only smoothes images but also preserves dominant edges, by means of non-linear combinations of nearby image values. Figure 7.1 shows an example of the data set we will be working in this part. To attain such a high noise reduction (figure 7.1 (right)), we perform bilateral filtering based on [464], provided by the OpenCV library ([49]; [273]), on the input image (figure 7.1 (left)) in several iterations.
Note that we transform the colors of the RGB input image $\boldsymbol{I}$ into **CIELab** space [286] for bilateral filtering to avoid specific color distortions, usually caused by gaussian filtering techniques [464]. See appendix A.1 for a more in-depth explanation of bilateral filtering and the mentioned risk of color distortions. Subsequently, the bilateral filtered image $\boldsymbol{I}_{bf}$ is converted to HSL color space, yielding $\boldsymbol{I}_{bf/HSL}$.

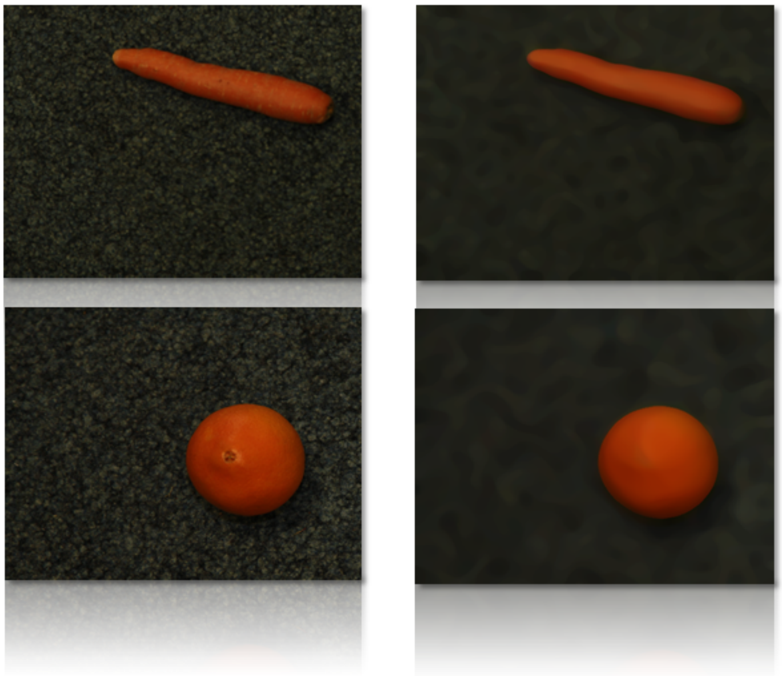


Figure 7.1: Left: Input image $\boldsymbol{I}$ (carrot, orange), Right: Bilateral filtered image $\boldsymbol{I}_{bf}$

Color attributes of each pixel $I_{bf/HSL}(x, y)$ are then stored within an augmented visual pixel $\boldsymbol{v}(x, y)$:

$$\begin{aligned} v_h(x, y) &= h_{bf}(x, y) \\ v_s(x, y) &= s_{bf}(x, y) \\ v_l(x, y) &= l_{bf}(x, y) \end{aligned}$$

Edge Detection

Edge detection is performed on the lightness channel of a spatial-resolution reduced version $\boldsymbol{I}_{\downarrow 2/bf/HSL}$ of $\boldsymbol{I}_{bf/HSL}$, as the alternative of converting the image to gray-scale might involve the risk of loosing certain potential edge information. Spatial reduction is based on **Gaussian image pyramids** ([64]; [63]; [12]; [222]) (see appendix A.4). **Gabor wavelet transform** ([156]; [100]; [99]), described in appendix A.3, relies on **Gabor wavelets** $\psi_{\varphi,\nu}(z)$ ([535]; [534]) of the form:

$$\psi_{\varphi,\nu}(z) = g_{\varphi,\nu,\sigma}(z)\left[e^{i\,k_{\varphi,\nu}\,(z)} - e^{-\frac{\sigma^2}{2}}\right] \tag{7.1}$$

with the **Gaussian envelope**:

$$g_{\varphi,\nu,\sigma}(z) = \frac{||k_{\varphi,\nu}||^2}{\sigma^2}\, e^{-\frac{||k_{\varphi,\nu}||^2\,||(z)||^2}{2\,\sigma^2}} \tag{7.2}$$

where the parameters φ and ν define the orientation and scale of the Gabor kernel and σ is the standard deviation of the Gaussian window in the kernel, i.e. the size of the window. $z = (x, y)$ indicates a point with x, the horizontal coordinate and y, the vertical coordinate. $k_{\varphi,\nu}$ is the wave vector, combining orientations and the spatial frequency in the frequency domain. Gabor wavelets are widely used in computer vision ([535]; [534]; [22]; [282]; [123]; [124]) because they provide an analysis of spatial frequency that is local, in contrast to the global analysis in a **Fourier transform** ([48]; [222]). The applied version of Gabor wavelet transform is provided by [534]. As we deal with rather coarse and big object shapes within this part, for edge extraction, we use of Gabor wavelets with a medium sized kernel ($\nu = 1$ and $\sigma = 2\pi$). Thus, Gabor wavelet transform is applied, in 32 orientations φ from $-90°$ to $90°$ with an angular difference of $5.625°$. An alternative would be to use a large kernel on the original image. However, advantages of the pyramid approach are that small scale variations in the image are filtered as well as computational complexity for filtering and further processing is reduced by a factor of 4.

Subsequently, at each image value (x, y), the 32 responses are evaluated creating a final gradient-orientation image $\boldsymbol{I}_{\downarrow 2/GO}$, see figure 7.2, that stores the orientation of the filter with the highest filter response at each pixel position (x, y).

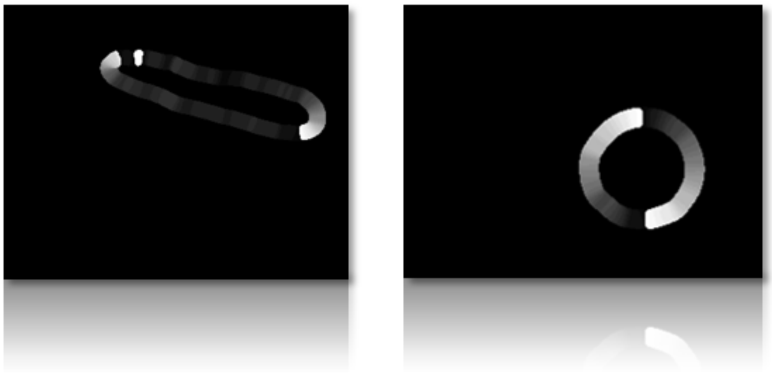


Figure 7.2: Edge orientations coded in gray-scale from white ($-90°$) to black ($90°$)

The evaluation procedure is inspired by the cascading of several **Simple cells** to form **Complex cells** ([174]; [68]; [215]) in the human visual cortex, as discussed in appendix B.1.

7.2 Simulated Surround Suppression Based on Orientation Maps

The rationale behind what we call **orientation maps** is to create something like an acoustical bas-relief that allows the user to hear what is under his fingers, instead of feeling it, along with the proposed bas-relief inspired exploration paradigm, discussed in section 5.3. Orientation maps represent dominant structures within the image, other than roughness. We consider dominant structures single or repetitive sets of significant edges of the same orientation and a particular direction of propagation. Our method is based on the observation that standard edge detectors such as the *Canny* algorithm [67] produce multiple edges and spurious, misleading signals that confuse the user. Therefore, the calculation of orientation maps involves filtering important from distracting structures, which may be motivated biologically from the **surround suppression** in the human visual system that improves contour detection ([229]; [439]; [339]) as described in appendix B.1. Moreover, a fundamental idea of orientation maps is that the user should not have to follow contours of objects or structures to estimate their silhouette, which would be tedious and slow. To make it easier for users to find contours, we distribute them around the edge by a kind of "diffusion" approach. Such an approach also connects nearby repetitive line patterns, again making it easier to recognize an object's shape even in between the edges.

Computation of Orientation Maps

Computing orientation maps $\boldsymbol{O}_\theta$ starts with quantification of the gradient-orientation image $\boldsymbol{I}_{\downarrow 2/GO}$ so that each of the 32 orientations φ is mapped to the next of 8 orientations:

$$\theta = \{0^\circ, 22.5^\circ, 45^\circ, 67.5^\circ, 90^\circ, 112.5^\circ, 135^\circ, 157.5^\circ\} \tag{7.3}$$

Each θ is represented in an individual gray scale image $\boldsymbol{I}_\theta$ where

$$\boldsymbol{I}_\theta(x,y) = \begin{cases} 255, & \text{if } (x,y) \in \text{edge of orientation } \theta \\ 0, & \text{otherwise} \end{cases}$$

The next step is the computation of a "diffusion" of edges. Therefore, we calculate for each image position (x,y) the **variance** $\sigma^2(x,y)$ ([458], [222]) of values $\boldsymbol{I}_\theta$ on its local neighborhood on each of the 8 images $\boldsymbol{I}_\theta$, to obtain $\boldsymbol{O}_\theta$:

$$\boldsymbol{O}_\theta(x,y) = \sigma^2(x,y) = \sum_{i=-\frac{w}{2}}^{\frac{w}{2}} \sum_{j=-\frac{w}{2}}^{\frac{w}{2}} (\boldsymbol{I}_\theta(x+i, y+j) - \mu(x,y))^2 \tag{7.4}$$

with

$$\mu(x,y) = \frac{1}{w^2} \sum_{i=-\frac{w}{2}}^{\frac{w}{2}} \sum_{j=-\frac{w}{2}}^{\frac{w}{2}} \boldsymbol{I}_\theta(x+i, y+j) \tag{7.5}$$

where $\mu(x,y)$ is the mean and w the size of the local neighborhood. Orientation maps $\boldsymbol{O}_\theta$ for the carrot and orange are shown in figure 7.3. Each coherent patch of gray scale pixels, on each orientation map $\boldsymbol{O}_\theta$ is referred to as **orientation patch** $\boldsymbol{V}_{\theta_i,i}$. As this diffusion approach might cause overlap in image positions (x,y) of different oriented orientation patches, we now compare orientation patches of different orientations θ and emphasize dominating ones while suppressing insignificant and therefore distracting others.

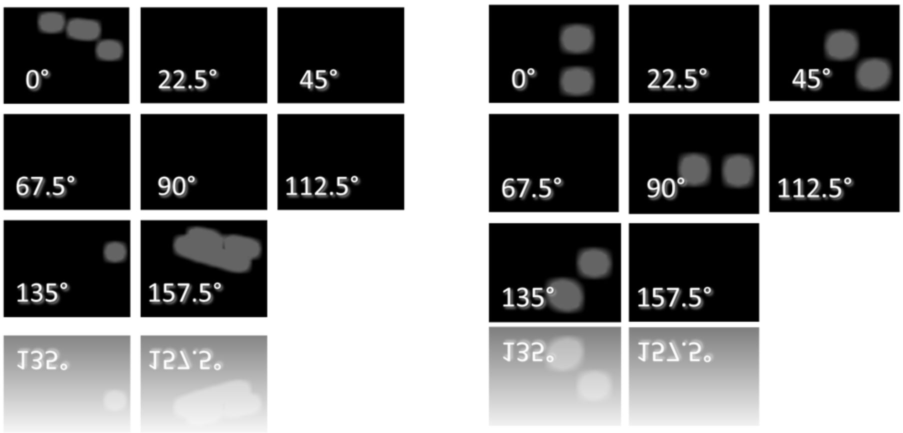


Figure 7.3: Orientation maps for the carrot image (left) and the orange image (right)

A Topological Representation of Orientation Patches

Orientation patches $\boldsymbol{V}_{\theta_i,i}$ are considered as dominant if they have a certain size, i.e. the number of their pixels positions $N_{\boldsymbol{V}_{\theta_i,i}}$. To formulate an iterative algorithm that is capable to keep dominant orientation patches while suppressing distracting ones, we define 4 cases, the algorithm will use for an iterative case-by-case analysis:

- **Case 1**: If an image area is dominated by 2 very big orientation patches $\boldsymbol{V}_{\theta_i,i}$ and $\boldsymbol{V}_{\theta_j,j}$, with $\theta_i \neq \theta_j$, of almost equal sizes, both above a certain threshold ($N_{\boldsymbol{V}_{\theta_i,i}}, N_{\boldsymbol{V}_{\theta_j,j}} > N_{thres}$), and such patches differ in orientation by more than 22.5°, they are retained as "coexisting". (This might be the case for, e.g., a rectangular grid or a wall of bricks where two orthogonal orientations are permanently present and form the particular textures of the image region.)

- **Case 2**: If the image area is dominated by 2 orientation patches $\boldsymbol{V}_{\theta_i,i}$ and $\boldsymbol{V}_{\theta_j,j}$, with $N_{\boldsymbol{V}_{\theta_i,i}}, N_{\boldsymbol{V}_{\theta_j,j}} > N_{thres}$, having an orientation difference of only $\|\theta_i - \theta_j\| = 22.5°$, which is the smallest possible difference, these patches are combined into a single orientation patch by merging the smaller one into the bigger one. Each pixel of the smaller orientation patch is assigned to the orientation map of the bigger one. Thereafter, the smaller orientation patch is erased from its orientation map.

- **Case 3**: If the image area contains a large orientation patch $\boldsymbol{V}_{\theta_i,i}$, with $N_{\boldsymbol{V}_{\theta_i,i}} > N_{thres}$, and further patches, e. g., ($\boldsymbol{V}_{\theta_j,j}$,$\boldsymbol{V}_{\theta_k,k}$,..), with $N_{\boldsymbol{V}_{\theta_j,j}}, N_{\boldsymbol{V}_{\theta_k,k}}, ... < N_{thres}$, such smaller patches are either (a) merged into the large patch $\boldsymbol{V}_{\theta_i,i}$, as described in Case 2, or (b) deleted from their corresponding orientation map (e.g. $\boldsymbol{O}_{\theta_j}, \boldsymbol{O}_{\theta_k}$), depending on whether their particular centers lie within $\boldsymbol{V}_{\theta_i,i}$.

- **Case 4**: If the image area contains several orientation patches, with $N_{\boldsymbol{V}_{\theta_i,i}}, N_{\boldsymbol{V}_{\theta_j,j}}, ... < N_{thres}$, they are merged as in Case 2, if $\|\theta_i - \theta_j\| = 22.5°$ and both their center positions overlap. Otherwise, they coexist.

Each iteration of the algorithm starts with **Step 1**, i.e. building a topological structure of all overlapping orientation patches by the following procedure:

- First, we apply a **connected-component analysis** ([451]; [109]) on each orientation map $\boldsymbol{O}_\theta$ that retrieves a sequence of all contour pixels as well as all enclosed image positions found within the map. Each contour, as well as the enclosed pixel positions, belongs to an orientation patch $\boldsymbol{V}_{\theta,i}$. Each $\boldsymbol{V}_{\theta,i}$ is represented as a single image, as illustrated in figure 7.5 (middle).

- So far, for each $\boldsymbol{V}_{\theta,i}$ we have its size, i.e. the number of pixel positions $N_{\boldsymbol{V}_{\theta,i}}$, and can now calculate its center position $C_{\boldsymbol{V}_{\theta,i}}$, known as **center of gravity** ([222], [458]).

- Starting with a particular $\boldsymbol{V}_{\theta,i}$, we now check pixel by pixel for overlaps with each orientation patch having a different orientation. In case of overlapping, we compute the number of mutual pixel positions $M_{\boldsymbol{V}_{\theta,i},\boldsymbol{V}_{\theta,j}}$, with $\theta_i \neq \theta_j$, and whether $C_{\boldsymbol{V}_{\theta,i}}$ of $\boldsymbol{V}_{\theta,i}$ lies within $\boldsymbol{V}_{\theta,j}$.

The results can be modeled using **graph theory** ([73]; [473]; [85]):

- Overlapping orientation patches can be modeled as an **undirected graph** $G_o = \{V, E\}$, where each knot V represents an orientation patch $\boldsymbol{V}_{\theta_i,i}$ and edges E represent the existence of an overlap between pairs of orientation patches $\boldsymbol{V}_{\theta,i}, \boldsymbol{V}_{\theta,j}$. $M_{\boldsymbol{V}_{\theta,i},\boldsymbol{V}_{\theta,j}}$ denotes the weight w of the edge.

- A second graph G_c is set up where the edges represent the existence of an overlap $M_{\boldsymbol{V}_{\theta,i},\boldsymbol{V}_{\theta,j}}$ between pairs of orientation patches where the center $C_{\boldsymbol{V}_{\theta_i,i}}$ and / or $C_{\boldsymbol{V}_{\theta_j,j}}$ lies inside the related orientation patch, as visualized in figure 7.5 (middle). In this case, connections may be only in one direction, which is why G_c would be defined as a **directed graph**. Both graphs, G_o and G_c, for all orientation patches of the carrot input image, are illustrated in figure 7.4.

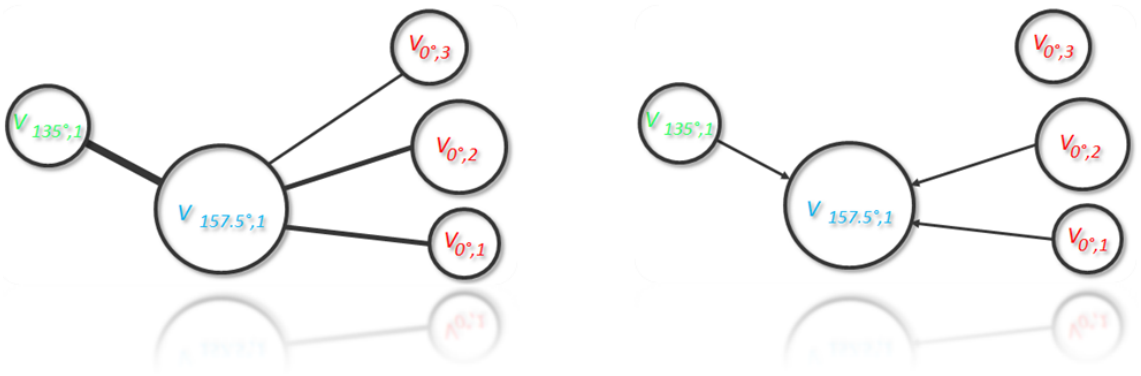


Figure 7.4: Left: Undirected Graph G_o, where knots V represent all orientation patches and edges E represent the existence of an overlap between a pair of different oriented orientation patches $\boldsymbol{V}_{\theta_i,i}$. Right: Directed graph G_c, set up to represent only overlaps, where at least the center c of one of the two orientation patches involved is found inside the related orientation patch. Sizes of V represent sizes of orientation patches. Thickness of E denote the number of overlapping pixel positions

Based on the topological structure represented by G_o and G_c, the algorithm, in **Step 2**, starts at an arbitrary knot of G_o checking with directly connected knots (i.e. orientation patches $\boldsymbol{V}_{\theta,i}$) which of the 4 previously mentioned cases fit.
Note that the algorithm automatically aborts execution of Step 2 and switches back to Step 1 as soon as any of the 4 cases could be applied because it implies a possible change in the topological structure of the graphs, representing the orientation maps. The algorithm finally stops when it ran through all knots of G_o without any of the cases applied. The general work-flow of the whole procedure is summarized in algorithm 1.

Algorithm 1 Compute final orientation patches $\boldsymbol{V}_{\theta_i,i}$ & orientation maps $\boldsymbol{O}_\theta$

```
repeat
  STEP 1 {Update Topological Structure}
  for all $\boldsymbol{O}_\theta$ do
    compute $\boldsymbol{V}_{\theta_i,i}$, $N_{\boldsymbol{V}_{\theta_i,i}}$, $C_{\boldsymbol{V}_{\theta_i,i}}$
  end for
  for all $\boldsymbol{V}_{\theta_i,i}$ do
    $V(G_o) \leftarrow \boldsymbol{V}_{\theta_i,i}$;  $V(G_c) \leftarrow \boldsymbol{V}_{\theta_i,i}$
    for all $\boldsymbol{V}_{\theta_j,j}$ do
      if $(\theta_i \neq \theta_j) \wedge (M_{\boldsymbol{V}_{\theta_i,i},\boldsymbol{V}_{\theta_j,j}} > 0)$ then
        $E(G_o) \leftarrow (\boldsymbol{V}_{\theta_i,i}, \boldsymbol{V}_{\theta_j,j})$;   $w = M_{\boldsymbol{V}_{\theta_i,i},\boldsymbol{V}_{\theta_j,j}}$
        if $C_{\boldsymbol{V}_{\theta_i,i}} \in \boldsymbol{V}_{\theta,j}$ then
          $E(G_c) \leftarrow (\boldsymbol{V}_{\theta_i,i}, \boldsymbol{V}_{\theta_j,j})$
        end if
      else
        proceed with next $\boldsymbol{V}_{\theta_j,j}$
      end if
    end for
  end for

  STEP 2 {Case-by-Case Analysis}
  for all $V(G_o)$ do
    for all $V \in$ neighbors of $V(G_o)$ do
      for all  $Case_i \in$ {1,..,4} do
        check criteria $Case_i$
        if if criteria fulfilled  then
          apply $Case_i$;   jump to STEP 1
        end if
      end for
    end for
  end for
until no case applied during STEP 2
```

To give an illustrative example, for the carrot input image (see figure 7.1 (left)), there would be 5 initial orientation patches on the 3 orientation maps (see figure 7.3(left)) whose interrelations can be described in Step 1 based on G_o and G_c (see figure 7.4). Step 2 is visualized in figure 7.5. $\boldsymbol{V}_{0^\circ,1}$ and,$\boldsymbol{V}_{0^\circ,2}$ would merge into $\boldsymbol{V}_{157.5^\circ,1}$ applying Case 3 (a) and $\boldsymbol{V}_{0^\circ,3}$ would be deleted, applying Case 3 (b) (see figure 7.5 (middle)). Hence, the final orientation maps $\boldsymbol{O}_{0^\circ}$, $\boldsymbol{O}_{135^\circ}$ and $\boldsymbol{O}_{157.5^\circ}$ are shown (see figure 7.5 (right)).

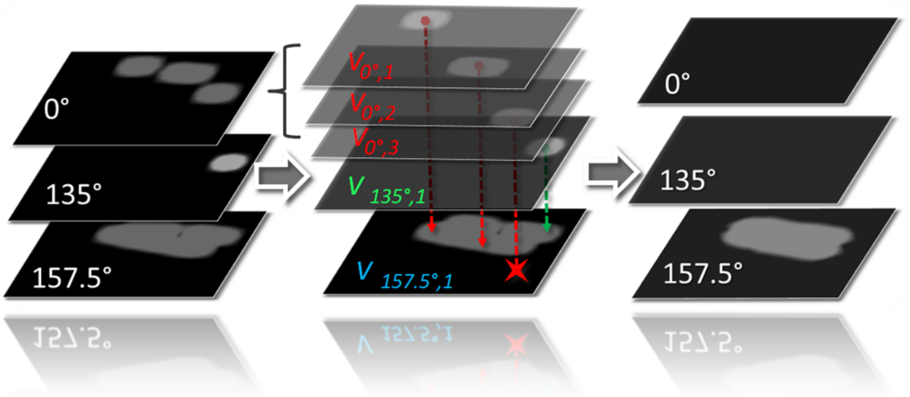


Figure 7.5: Left: Split initial orientation maps $\boldsymbol{O}_\theta$. Middle: Arrows denote the center positions of $\boldsymbol{V}_{0^\circ,1}$, $\boldsymbol{V}_{0^\circ,2}$ and $\boldsymbol{V}_{0^\circ,3}$ to lie within $\boldsymbol{V}_{157.5^\circ,1}$. The cross denotes the center of $\boldsymbol{V}_{135^\circ,1}$ to nots lie within $\boldsymbol{V}_{157.5^\circ,1}$. (Note that for illustrative reasons, only a selection of arrows has been visualized.) Right: Final orientation maps $\boldsymbol{O}_\theta$

In contrast, the Graph G_o based on the orientation maps of the orange image (see figure 7.3 (right)) result in a ring shaped connected structure where every knot is about the same size. Thus, such orientation patches would be left as they are, according to Case 4 which intuitively makes sense, as a spherical object like an orange does not have any significant propagated orientation.

Eventually, we can store the shares of all eight orientation maps $\boldsymbol{O}_\theta$ at a particular pixel position (x, y) in its corresponding augmented visual pixel $\boldsymbol{v}(x, y)$:

$$v_{\boldsymbol{O}_{\theta\in\{0^\circ,22.5^\circ,45^\circ,67.5^\circ,90^\circ,112.5^\circ,135^\circ,157.5^\circ\}}}(x, y) = \boldsymbol{O}_{\theta\in\{0^\circ,22.5^\circ,45^\circ,67.5^\circ,90^\circ,112.5^\circ,135^\circ,157.5^\circ\}}(x, y)$$

7.3 Textural Roughness

Beside the rather large salient structures such as orientation maps, we want to capture further information that describes small intensity variations within the image. Such smaller variations are generally described as **image texture**. Although it has been proven difficult to formulate a precise definition, Lew [286] summarizes two general properties:

- *Within a texture there is significant variation in intensity levels between nearby pixels; that is, at the limit of resolution, there is non-homogeneity.*
- *Texture is a homogeneous property at some spatial scale larger than the resolution of the image.*

Due to such a definition, on a larger scale, even patterns of orientation maps might form some sort of texture, which is why Lew [286] defines those larger patterns as **macro-textures**. However, we now want to extract information about rather small intensity variations and patterns, denoted in [286] as **micro-textures**. Tamura et al. [456] and Laws [277] identify various perceived qualities, texture, in general can be described by, such as "uniformity", "density", "coarseness", "roughness", "regularity", "linearity", "directionality", "direction", "frequency" and "phase", some of which dependent on one another. We decide to extract and sonify roughness of image regions, as a general and characteristic property of potential objects within an image. Because of these different perceptions inherent in texture there is a variety of approaches to analyze various aspects of texture ([101]; [357]; [502]; [474]; [190]; [120]; [445]; [286] [458]; [222]; [358]; [484]). Lew [286] describes three different groups of texture measures:

- **Statistical Texture Measures**: Statistical methods analyze the spatial distribution of intensity levels within an image by computing local features at each pixel position and deriving a set of statistics from the distributions of such local features ([222]; [458]; [445]). The rationale is that the spatial distribution of intensity values is one of the most distinct qualities of texture. Depending on the number of pixels defining each local feature, statistical methods can be further classified into **first order statistics** based approaches ([222]; [458]; [445]) as well as **second order statistics** or **higher order statistics** based approaches ([222]; [191]; [190]; [167]; [445]; [484]). First-order statistic based approaches only capture properties, such as mean or variance, of individual pixels, ignoring any potential spatial interactions between image pixels. In contrast, Second- and Higher-order statistics based approaches estimate characteristics of two or more pixel values occurring at specific locations relative to each other.

- **Stochastic Texture Modeling**: Due to this approach, texture is considered the realization of a stochastic process. Thus, a texture analysis is performed by defining a "model" and reproducing the stochastic process ([187]; [120]; [158]). The parameters affiliated with the model can then serve as features for ,e.g., texture classification. A major drawback of stochastic texture modeling is that many natural textures do not conform the restrictions of a specific model defined.
- **Structural Texture Measures**: This approach considers textures to be seen as two dimensional patterns that are formed by sets of sub-patterns that have been arranged due to specific rules. These rules allow for both, varying or deterministic shapes ([190]; [187]).

Gradient Based Entropy as a Local Roughness Measure

To capture the textural roughness of regions in the image, we define a measure based on the concept of **Entropy** ([334]; [54]; [87]), first developed in Classical Thermodynamics [79]. In image texture analysis, it is defined by **Haralick's texture measures** in ([191]; [190]). The general approach would ,therefore, belong to the group of Statistical Texture Measures. Our approach, however, slightly alters the original implementation of entropy for image analysis, as our method does not work with intensity levels but image gradients. This is due to a more robust differentiation of rough from smooth regions. Thus, Entropy is computed directly from a gradient-orientation image $\boldsymbol{I}_{GO,texture}$, that stores the orientation of the filter with the highest filter response at each pixel position (x, y) as a result of the Gabor Wavelet Transform applied to the lightness channel of $\boldsymbol{I}_{HSL}$. $\boldsymbol{I}_{GO,texture}$ is received exactly as discussed in section 7.1, except the Gabor Wavelet Transform is applied to the original full-sized and non-smoothed input image to preserve roughness information.

Entropy, generally measures the disorder within a physical system, and is used in various scientific fields. Having zero entropy means to have maximum information about the state of a system [334]. In information theory, it is formulated as ([427]; [429]):

$$H = -\sum_{i=1}^{N} p_i \, log \, p_i \quad \text{with} \quad p_i = \frac{N_i}{N} \tag{7.6}$$

We calculate the Entropy $H(x, y)$ for each pixel position (x, y) based on the Gabor transform response within a local quadratic neighborhood of 11×11 pixels. The variable p_i is the probability for a certain orientation φ_i estimated from its occurrence N_i divided by the total number N of all orientations φ that occur within the window, $N = 32$ in our case. $H(x, y)$ is further normalized to $0 - 1$:

$$H'(x, y) = \frac{H(x, y)}{H_{max}} \quad \text{with} \quad H_{max} = -log(\frac{1}{N}) \tag{7.7}$$

, where H_{max} would be the maximum entropy. The rationale behind is, an increase in roughness (from isotropic to an-isotropic) is represented in an increase in entropy. A smooth area is expressed as zero entropy.

Finally, for all pixel positions (x, y) detected to lie on any found edge in the up-scaled (to full-size version of the gradient-orientation image $\boldsymbol{I}_{GO}$, as computed in section 7.1), entropy $H'(x, y)$ is set to zero, so that

$$H''(x, y) = \begin{cases} 0, & \text{if (x,y)} \in \text{edge} \\ H'(x, y), & \text{otherwise} \end{cases}$$

This is done to not confuse salient edge regions with isotropic rough regions. Finally, $H''(x, y)$ is stored within the augmented visual pixel $\boldsymbol{v}(x, y)$:

$$v_H(x, y) \quad = \quad H''(x, y)$$

Note that this texture measure does not discriminate any specific orientation. A major drawback of this approach is the computation redundancy, as for neighboring pixels almost identical computations are performed. More sophisticated approaches to roughness computation will, therefore, be discussed in part IV of the thesis in section 11.3.

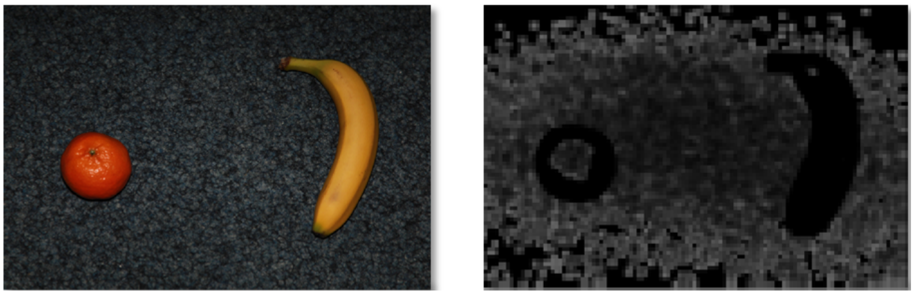


Figure 7.6: Left: Original Image. Right: The entropy calculated per pixel $H(x, y)$. Intensity coded from $0 - 1$ in grayscale

7.4 Shape Extraction of Basic Objects using Graph Cuts

The computation of especially orientation maps is rather complex depending on the number of found orientation patches, and makes foremost sense on a salient coherent object, rather than a complete image. Hence, we incorporate state of the art segmentation algorithms into our system, which are able to separate the image into regions that are likely to show different real life objects.

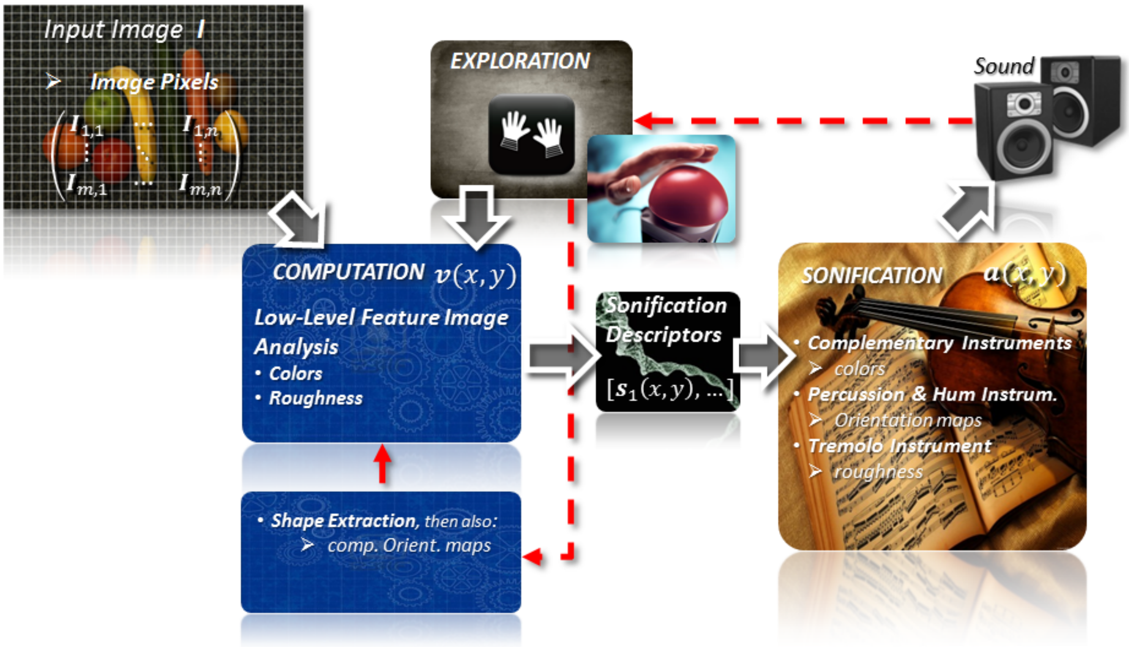


Figure 7.7: A slightly modified version of the realization of the Modular Computer Vision Sonification Model given in chapter 6. If the user hits the red “buzzer”, the segmentation process is initiated starting from his current (x, y) position

Figure 7.7 shows the slightly modified version of Modular Computer Vision Sonification Model implementation, given in chapter 6. Thus, the user starts exploring the image using pre-computed colors and roughness values only. As the user moves over a, e.g. color coherent area which he considers an object, he initiates the segmentation procedure hitting an external additional “buzzer” button. First, based on the users current position (x, y), we apply a **seeded region growing** algorithm ([397]; [222]) that iteratively adds pixels to an area around (x, y), if their color distance to the average color of the region is below a threshold, using (x, y) as an initial seed point. The color distances are calculated as the Euclidean distance $\|.\|$ in CIELab [286] color space. This region growing is due to gather enough pixel data which can then be marked as “definite foreground”, all others as “probably background”, and both groups serve as first segmentation estimation and as input to an foreground extraction algorithm based on **Graph Cuts** ([401]; [47]; [254]; [45])

which then calculates the final segmentation. For out application we use an implementation based on [401] provided by the OpenCV library ([49]; [273]). Finally, the rest of the image is set to "black" and orientation maps are computed on the segmented area only. Figure 7.8 shows some exemplary results of the segmentation, as well as the responses of the Gabor wavelet transform applied to such segmentations. The whole segmentation process takes approximately 4.5 seconds and can, therefore, be considered for interactive usage. Note that the same external "buzzer" can be used to switch back to exploring the original image, again based on colors and roughness.

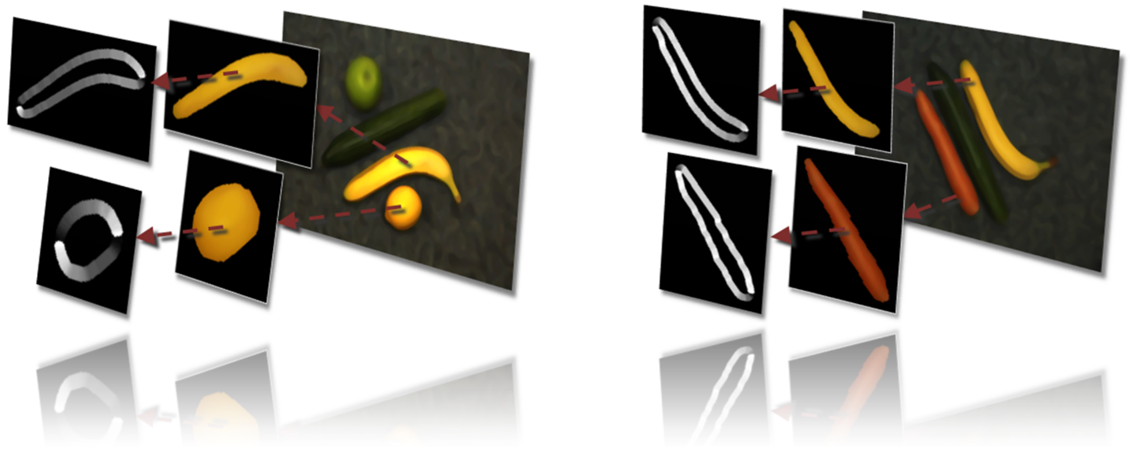


Figure 7.8: Examplary results of the image segmentation and subsequent responses of the Gabor wavelet transform applied

Chapter 8

Color and Low-Level Feature Sonification

8.1 Audible Color Space

As mentioned in chapter 4, color sonification is meant to be intuitive enough not only to be understood and applied by congenital blind, but also to convey, e.g., the concept of colors and color mixing itself. Thus, the approach proposed in this section maps each attribute of a particular color within the HSL color space, i.e., hue h, saturation s and lightness l to an intuitive counterpart within the sound space. Hence, we form some sort of "audible color space".

Throughout time, scientists and artists have been concerned with the correspondences between color and sound and numerous studies on the subject have been written ([66]; [342]; [23]; [350]; [163]). Already, in his book "De Sensu et Sensibili" [485], Aristotle assumes that the aesthetics of color groupings is governed by the same rules that govern musical consonances. Newton, in his famous work "Opticks" [336], compares light, which according to its wavelength excite the different sensations of color, with the air vibrations, which according to their length also excite the sensations of the different sounds. On this basis, a comparison is usually drawn between the chromatic scale of sounds and the hue circle. In this sense, different proposals have been developed by Newton [336], Munsell [329] and others ([221]; [366]). Goethe [494], however, denies any direct comparison between sound and color, still maintaining that both phenomena can be referred to some "superior formula". Some researches have expanded and refined the comparison, introduced by Newton, considering other variables of sound and color, too. Giannakis [163] maps color luminosity levels to octaves. Caivano [66] relates luminosity of color with loudness of sound as well as saturation of color with timbre of sound. Barrass [23] cites Padgham [342] and Caivano [66] to be considered the first to model a full acoustical representation of the entire color space. The auditory-visual associations they proposed are summarized in table 8.1.

	Hue	**Saturation**	**lightness**
Padgham [342]	Formants	Timbre	Loudness
Caivano [66]	Pitch	Timbre	Loudness
Barass [23]	Timbre	Brightness	Pitch

Table 8.1: Attempts to model sound using colour dimensions, based on the HSL color space representation

However, there are some noticeable differences between hue and pitch. Perhaps the most-striking is the one pointed out by Helmholtz [495], that the auditory range comprises around ten octaves while the visible range hardly covers a single "octave". Helmholtz, furthermore, notes other differences between audition and vision in [495]. These observation lead him to the conclusion to abandon color hue - pitch analogies in general.

Complementary Instruments Inspired by Hering's Theory of Opponent Colors

The concept we propose represents each color value in the HSL model as a mixture of instruments, based on General MIDI (GM), inspired by Herings theory of **Opponent colors** ([202]; [174]; [179]). This theory could be affirmed to play a significant role in the processing of incoming color stimuli from the retina [237], as discussed in appendix B.1. The rationale behind using GM instruments is that visual impaired people may find this a comfortable way to get perceptual access to colors and textures.

In principle, we use what we call **complementary instruments** to represent the opponent color pairs red-green and blue-yellow, and later combine adjacent instruments to represent color mixtures. As no mixture of a pair of opponent colors exists [174], there will be no mixture of a pair of complementary instruments in our sonification model either. Furthermore, we apply a musical scale to represent the luminance scale from black to white. Complementary instruments, therefore, must guarantee certain characteristics:

- **Stability**: Complementary instruments must possess a relatively stable frequency spectrum over time. That means that in terms of **Attack - Decay -Sustain - Release (ADSR) amplitude envelope** [114], as illustrated in figure 8.1 (left), they should have a short Attack- and Decay-, an infinite Sustain- and a short Release-phase.

- **Separability**: It ensures that instruments, assigned to adjacent colors can be clearly distinguished even when they are played as mixtures. This criterion does not need to be met by complementary instruments.

- **Uniqueness**: Even complementary instruments need to be unique enough to be associated with its particular color.

- **Non-Interference**: Finally, we want to make sure that mixtures of instruments do not sound like other, new instruments. To avoid mutual masking of instruments, their frequency spectra should have narrow bandwidths (i.e. little noise components).

Figure 8.1 (right) shows our final selection of instruments: **choir** (red), **bagpipe** (yellow), **organ** (green), **violins** (blue) and **flute** (white, black, gray-scale). However, working with an external MIDI synthesizer, as presented in section 8.4, the software allows the user to assign own selection of preferred instruments. The specific role of gray-scale, black and white with only one instrument will be explained below.

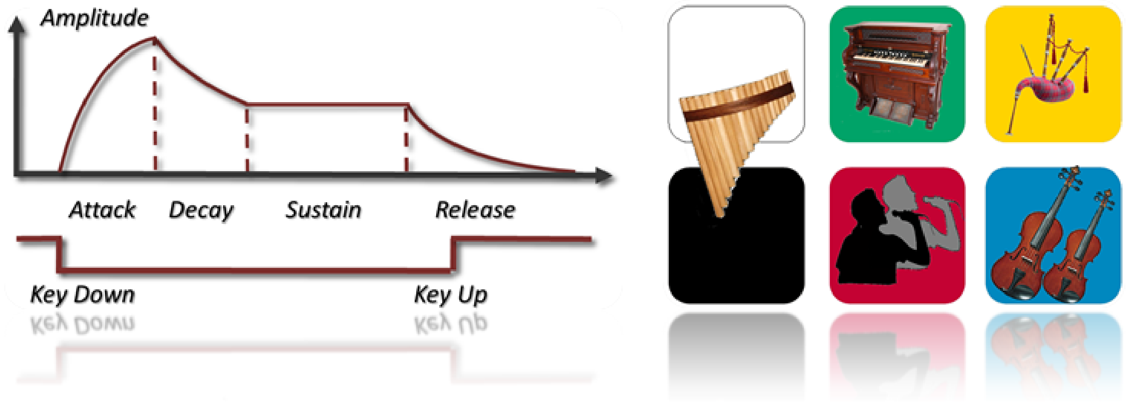


Figure 8.1: Left: An illustration of an ADSR amplitude envelope. Right: Our selection of complementary instruments to represent pairs of opponent colors. **choir** (red), **bagpipe** (yellow), **organ** (green), **violins** (blue) and **flute** (white, black, gray-scale)

Sonification of the HSL Color Space

Based on our idea to assign complementary instruments to certain hues, we sonify intermediate color tones as mixtures of two adjacent instruments, and represent the color mixture ratio by their partial volume. Hence, the share of each instrument is controlled by individually computed parameters, which we call **volume shapes** $\vartheta \in 0, .., 1$. The fade of saturation s, moving inward to the center of figure 8.2 (left), is considered as a general absolute decrease in volumes of any two color instruments playing simultaneously, while their relative volume ratio is maintained.

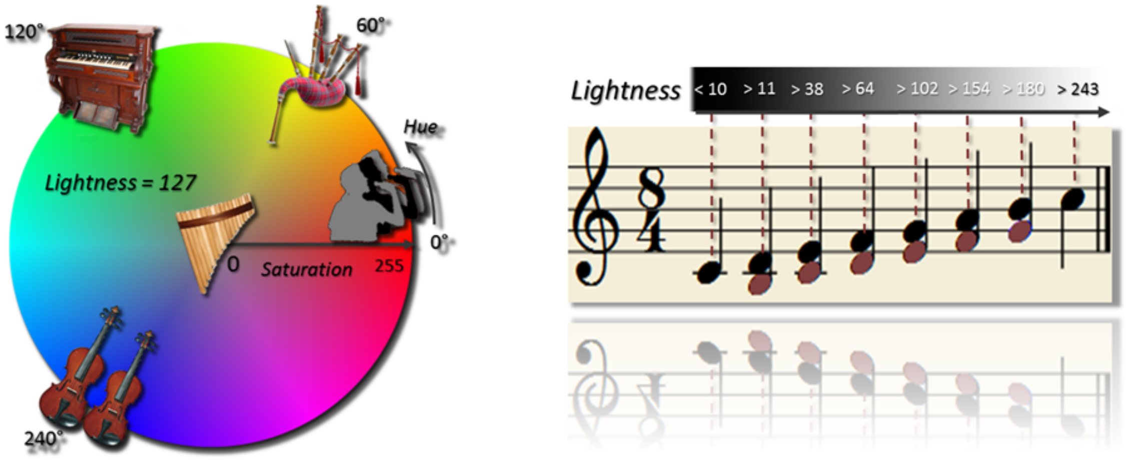


Figure 8.2: Left: Complementary instruments within the HSL model at $l = 127$ (50%). Right: Lightness $l \in 0, 1, , 255$ and musical scale each l is assigned to. Brown notes are the added thirds

Gray-scale and Lightness

Below a certain threshold $s_{min}(l)$, we regard the color as gray and sonify it using a single instrument, the flute at a constant volume. Furthermore, a color is also considered to be black or white, if its lightness l is below or above a specific threshold, $l < l_{min}$ and $l > l_{max}$, respectively. In general, gray is not considered a color, and the HSL model assigns it an arbitrary hue $h = -1$ and a saturation $s = 0$. Still, we found it helpful to use a separate instrument for gray, which partly reflects the fact that many languages have a separate name for it.

Note that the minimum saturation $s_{min}(l)$ differs, depending on the specific lightness value l, due to the properties of the HSL model, which is often deformed to be illustrated in a cylindrical instead of a double cone representation, as illustrated and discussed in appendix B.2. Thus, as $s_{min}(l)$ is approximately independent of h we can first select a few representative lightness positions l as well as their corresponding $s_{min}(l)$ values and use those pairs as control points to fit a **Catmull-Rom** spline ([139]; [138]) through them.

This approach, as described in ([16]; [17]), yields a value s_{min} for every l that can be used as the minimum threshold for that l. Second, we can build up a **solid of revolution** ([193]; [247]) structure (see appendix A.8), rotating the $s_{min}(l)$-curve around the l-axis. In contrast to $s_{min}(l)$, within this $3D$ structure $\vartheta_{gray}(h, s, l)$ every entry would be set to either 1 or 0, depending whether a specific triplet (h, s, l) would lie within $s_{min}(l)$ or beyond and further denoting whether the flute, representing gray, is to play or be muted. $\vartheta_{gray}(h, s, l)$ would have an additional entry referring to $h = -1$ being $\vartheta_{gray}(-1, s, l) = 1$.

The lightness l of gray or any other (combination of) colors is sonified as the pitch of the tone, which might be an intuitive acoustical representation to effectively recognize significant changes in lightness. Pitch further allows to define some intuitive "borders" in the lightness representation. Thus, "black" as the darkest color could be assigned to a certain comfortable frequency whereas an intuitive upper border, representing "white" would then probably be e.g. the first or second harmonics over such a frequency.

Based on a musical scale, as shown in figure 8.2 (right), black, as the lowest lightness value, is assigned to the tonic keynote, whereas white to its octave. In between there are six whole tones and 12 semitones. For harmonic reasons we only utilize the whole tones of a single octave and map each lightness value l between 0 and 255 to one of the eight tones, forming a 1D lookup structure $note(l)$:

key $note(l)$	**lightness range**
C	$0 - 10$
D	$11 - 37$
E	$38 - 63$
F	$64 - 101$
G	$102 - 153$
A	$154 - 179$
H	$180 - 242$
C	$242 - 255$

The mapping of l values to specific tones is determined judging by our own visual perception of approximately equal lightness intervals, although there has been some research on "human perception based color segmentation" that might corroborate the concept of quantizing colors ([479]; [426];[370]; [241]; [6]; [80]).

Further, we add thirds to all six intermediate tones (see figure 8.2 (right)). This creates a more comforting and aesthetic resonance and offers an elegant way to recognize whether one has reached the top or bottom of the scale, as they are played without thirds. Otherwise, users would need perfect pitch to recognize black and white. The motivation to use distinct notes instead of a continuous pitch would be, first, that the use of MIDI instruments allows only for pitches in the range of a single whole tone. Second, in part IV of the work, we extend the idea of using musical scales to emphasize the sad monotony of gray-scale values in contrast to the rather "joyful" colors switching between a rather sad "natural minor scale" to a more "joyful" "major scale" depending on each current color value sonified.

Note that in general, the perception of the lightness at a specific position within an image depends on the position's surroundings, as discussed in appendix B.1. This characteristic is on a very small scale reflected through the initial image filtering. However, it might be an interesting issue for further research to represent such specific characteristics of the visual system in an acoustical way.

A Color Sound Synthesis Equation

Finally, we formulate a **color sound synthesis equation** that describes the mapping from color values (h, s, l) stored within each augmented visual pixel $\boldsymbol{v}(x, y)$, via sonification descriptor $\boldsymbol{s}(x, y)$, into a sound, referred to as the color sound attribute $a_{color}(x, y)$ of the audible pixel $\boldsymbol{a}(x, y)$:

$$\begin{aligned} a_{color}(x,y) &= \vartheta_{gray}(h,s,l) * \boldsymbol{flute}(\eta) \\ &+ (1-\vartheta_{gray}(h,s,l)) * \Big[\vartheta_{red}(h,s) * \boldsymbol{choir}(\eta) + \vartheta_{green}(h,s) * \boldsymbol{organ}(\eta) \\ &+ \vartheta_{yellow}(h,s) * \boldsymbol{bagpipe}(\eta) + \vartheta_{blue}(h,s) * \boldsymbol{violins}(\eta)\Big] \end{aligned} \tag{8.1}$$

with:

$$h = s_h(x,y) = v_h(x,y), \quad s = s_s(x,y) = v_s(x,y), \quad l = s_l(x,y) = v_l(x,y)$$

and:

$$\eta = note(l)$$

Note that this equation and its mappings are relatively easy reversible, which is crucial for a congenital blind to learn and understand the concepts of colors. Generally, volume shapes $\vartheta(h, s)$ of colors are independent of l.

Interpolation of Audible Colors Based on Thin Plate Splines

Calculating the volumes of instruments, i.e., their volume shapes ϑ, in a mixture of sounds for all intermediate colors can be formulated as an interpolation problem. The basic idea would be to define a mapping ϑ on a set of control points manually to achieve the desired volumes for specific color values and mixtures and interpolate all values in between. Simple linear (barycentric) interpolation is be too restricted because once the overall volume of each instrument is set, there is no way to counteract the dominance of some instruments in some specific mixtures. Therefore, we employ non-linear interpolation using on **Thin Plate Splines (TPS)** [117], [41] based on a set of control points, as discussed in appendix A.10. The implementation of Thin Plate Splines used in our framework is provided by [131], which is mostly based on [117], which we modify to work for our specialized purposes. The fundamental idea behind Thin Plate Splines is the physical model of a flat thin medal plate that is deformed by a few punctual strains, the control values c. The plate is than forced into a new form that minimizes the deformation energy.

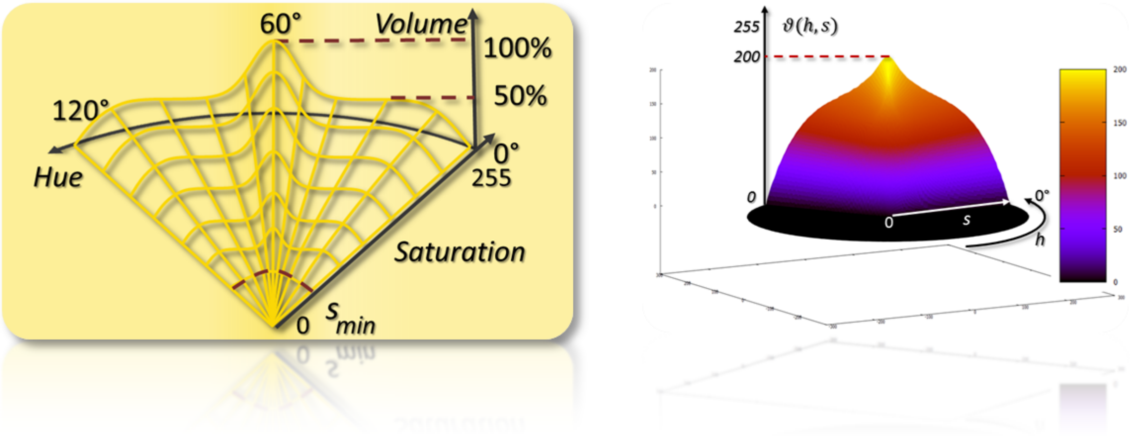


Figure 8.3: Left: Illustration of a volume Shape $\vartheta(h, s)$. Right: $3D$ representation of $\vartheta_{yellow}(h, s)$ (figure 8.4 (bottom-left)). Note that the highest possible volume level is set to 0.8 (200) instead of 1.0 (255). For computational complexity reasons, values were stored as less memory requiring **character**, instead of **floating point** values.

Based on this method, we can now calculate volume shapes $\vartheta(h, s)$ for each complementary instrument to each color (h, s) quite elegantly. As an example, the computational considerations for $\vartheta_{yellow}(h, s)$ are visualized in figure 8.3 (left). Generally, the volume should be $100\% = 1$ at the exact position of the corresponding opponent color, such as yellow in the example at hue $h = 60°$ and full saturation $s = 255$, and 0 at hues h equal to 0° and 120° or greater, disregarding any saturation s.

To control the volumes in mixed sounds, we add control values c at various positions (h_c, s_c). This would be, e.g., $(h_c, s_c) = (60°, 255)$ with $c = 0.8$. The use of control point based interpolation allows us to compensate for dominant instruments within mixtures, as with the realization of yellow through a significant but also rather dominant bagpipe instrument (see 8.3 (right)). Regular MIDI synthesizers offer the possibility to set the volume of a particular instrument within a given interval (see section 8.4). However, as discussed in section 2.1 this does not necessarily mean that the perceived loudness of instruments would be identical. Hence, an individual computation of volume shapes via control point based interpolation is an elegant way to compensate for that.

The results of computing these volume shapes $\vartheta(h, s)$, responsible for instrument interpolations, can be seen in figure 8.4. Note that the range of values for each $\vartheta(h, s)$ is from 0 to max. 255 instead of 1. This is because each $\vartheta(h, s)$ is stored and loaded as a lookup table using less memory requiring “character”, instead of “floating point” values.

As mentioned, volume shapes $\vartheta(h, s)$ are so far independent of l and hence apply at every position (h, s, l) if $\vartheta_{gray}(h, s, l) = 1$. However, in part IV we refine the idea of volume shapes $\vartheta(h, s)$ to incorporate a change in color intensity along l.

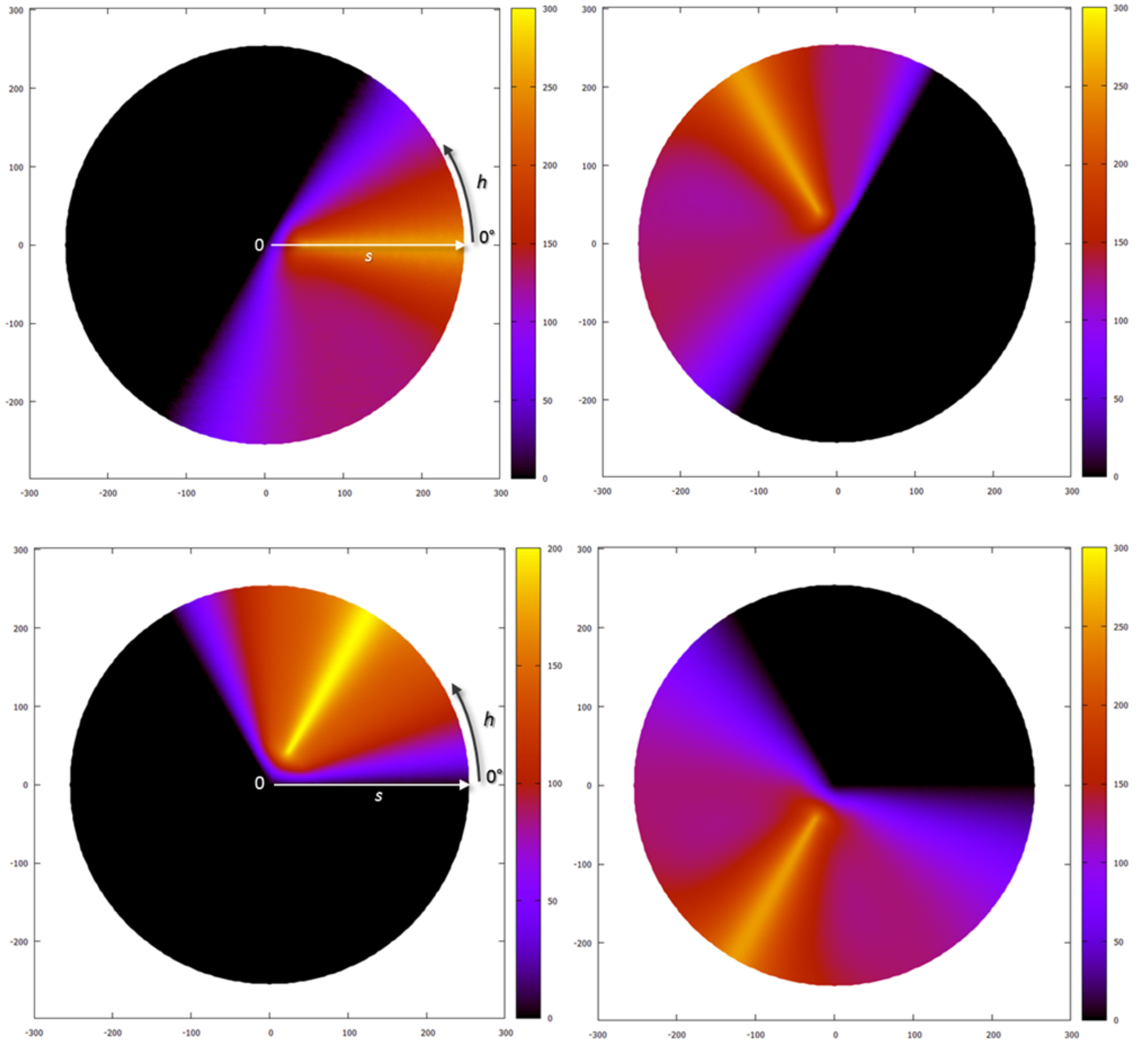


Figure 8.4: Volume shapes $\vartheta_{red}(h,s)$ (top-left) and $\vartheta_{green}(h,s)$ (top-right) and $\vartheta_{yellow}(h,s)$ (bottom-left) and $\vartheta_{blue}(h,s)$ (bottom-right). Note that the range of values for ϑ_{red}, ϑ_{green} and ϑ_{blue} is from 0 (black) to max. 255 (orange-yellow). The range of ϑ_{yellow}, representing the volume of a rather dominant instrument, has been set from 0 (black) to max. 200 (yellow)

An Audible HSL-Opponent Color Space

Note that though working with the HSL color space, in which the distances between red, yellow, green and blue are not equal, due to volume shapes $\vartheta(h, s)$ computation based color interpolation and sonification, we are able to "stretch" and "compress" the color space acoustically, illustrated in figure 8.5, creating an "audible color space" that is as easy to understand as the HSL color space, while maintaining equidistant distances between opponent colors as in the CIELab space. The reason why we do not use CIELab space in the first place is its characteristic to represent only "equally bright perceived" colors and, therefore, it is not consistent, meaning, there is no equivalent opponent color for every specific bright color. Thus, it cannot be represented as filled solid body, such as a cylinder, as described in appendix B.2.

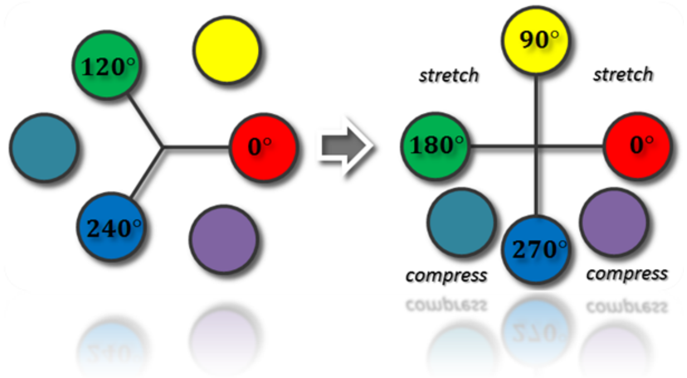


Figure 8.5: An "audible color space" creation that is as easy to understand as the HSL color space (left), while maintaining equidistant spacings between opponent colors as in the CIELAB space (right)

8.2 Auditory Edge Detection - the Sonification of Orientation Maps

Orientation maps are represented using additional instruments. However, as too many additional instruments interfere and make either recognition of colors or anything else impossible, we utilize only four more instruments, playing an octave below our keynote, to represent the 4 orientation maps $\boldsymbol{O}_{\theta=0^\circ}$, $\boldsymbol{O}_{\theta=45^\circ}$, $\boldsymbol{O}_{\theta=90^\circ}$ and $\boldsymbol{O}_{\theta=135^\circ}$.
Figure 8.6 shows our selection of instruments: **Digeridoo** ($\boldsymbol{O}_{\theta=0^\circ}$), **wooden percussion** ($\boldsymbol{O}_{\theta=45^\circ}$), **Uilleann pipes** ($\boldsymbol{O}_{\theta=90^\circ}$) and **metal percussion** ($\boldsymbol{O}_{\theta=135^\circ}$). This way, we create a hum alike sound at 0° and 90° and a percussion sound at 45° and 135°, which might be appropriate to quickly distinguish horizontal or vertical from diagonal structures.

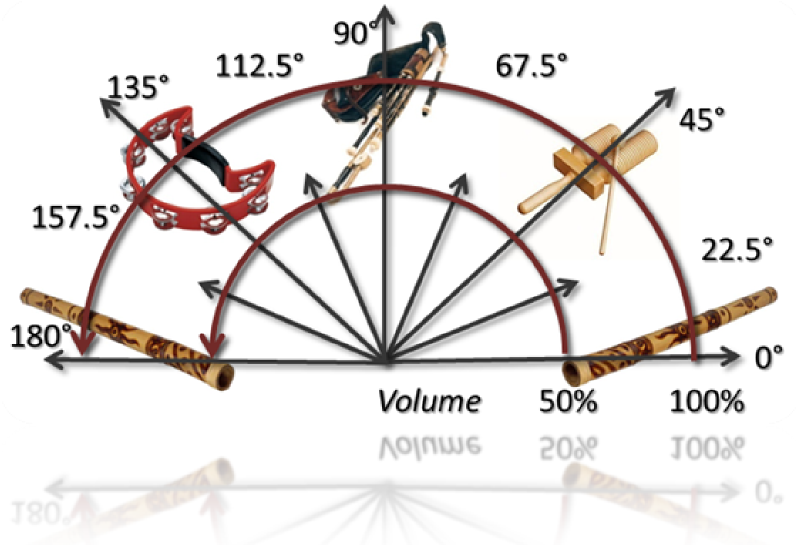


Figure 8.6: Our selection of instruments to sonify orientation maps. **Digeridoo** ($\boldsymbol{O}_{\theta=0^\circ}$), **wooden percussion** ($\boldsymbol{O}_{\theta=45^\circ}$), **Uilleann pipes** ($\boldsymbol{O}_{\theta=90^\circ}$) and **metal percussion** ($\boldsymbol{O}_{\theta=135^\circ}$). The arrows of different length denote either $vol_{\boldsymbol{O}_{\theta}(x,y)} = 100$ % of instruments at $\theta \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$ or $vol_{\boldsymbol{O}_{\theta}(x,y)} = 50$ % of the two neighbored instruments to $\theta \in \{22.5^\circ, 67.5^\circ, 112.5^\circ, 157.5^\circ\}$

The four orientation maps in between of the previous discussed, $\boldsymbol{O}_{\theta=22.5^\circ}$, $\boldsymbol{O}_{\theta=67.5^\circ}$, $\boldsymbol{O}_{\theta=112.5^\circ}$ and $\boldsymbol{O}_{\theta=157.5^\circ}$, are expressed based on a combinational approach. Thus, to sonify one of the “in between” orientation maps just mentioned, 2 of the directly neighbored orientation map’s instruments are utilized. To distinguish “in between” from the previously mentioned ($\boldsymbol{O}_{\theta=0^\circ}$, $\boldsymbol{O}_{\theta=45^\circ}$, $\boldsymbol{O}_{\theta=90^\circ}$ and $\boldsymbol{O}_{\theta=135^\circ}$), those are played at $0.5 * vol_{\theta}$ each (illustrated in figure 8.6).

Finally, each orientation maps' sound attribute $a_{\boldsymbol{O}_\theta}(x, y)$ of the audible pixel $\boldsymbol{a}(x, y)$ would be:

$$a_{\boldsymbol{O}_\theta}(x, y) = vol_{\boldsymbol{O}_\theta(x,y)} = \begin{cases} 1, & \text{if } (x, y) \in \boldsymbol{V}_{\theta,i} \text{ with } \theta \in \{0°, 45°, 90°, 135°\} \\ 0.5, & \text{if } (x, y) \in \boldsymbol{V}_{\theta,i} \text{ with } \theta \in \{22.5°, 67.5°, 112.5°, 157.5°\} \\ 0, & \text{otherwise} \end{cases}$$

Again, the framework allows exchanging instruments according to personal taste. To avoid auditory masking, described in section 2.1, we set all $vol_{\boldsymbol{O}_\theta(x,y)}$ to guarantee for the perceived loudness of all orientation map instruments to be always lower than that controlled by $\varphi(h, s)$ for the color instruments for all h and s.

8.3 Audible Roughness - the Sonification of Local Entropy

The most appropriate acoustical representation of textural noise might be audible noise, presented in section 3.2. However, working with MIDI instruments do not directly provide an appropriate instrument. Hence, we chose an instrument that has a quite vibrant temper. As Seashore [421] pointed out:

> *a good vibrato is a pulsation of pitch, usually accompanied with synchronous pulsations of loudness and timbre, of such extent and rate as to give a pleasing flexibility, tenderness, and richness to the tone.*

The instrument chosen were a special compilation of vibrating strings, and, as we experienced, it is a pretty intuitive way to represent roughness acoustically. The higher the computed roughness value $v_H(x, y)$, the louder the vibrant roughness instrument is:

$$a_{roughness}(x, y) = vol_{roughness} = \begin{cases} v_H(x, y), & \text{if } v_H(x, y) > 0 \\ 0, & v_H(x, y) = 0 \end{cases}$$

8.4 MIDI Based Sonification

As the sonification in this part relies on MIDI instruments to sonify color and features, a considerable amount of preparation has to be done. First, an external MIDI synthesizer has to be chosen that will accept common MIDI messages, as illustrated below, sent by our application to play notes and control instruments volumes, a.s.o. Additionally to the synthesizer, data sets of instruments have to be loaded and specific instruments have to be assigned to specific midi channels. In our case this is one for each of the 5 instruments representing color (see figure 8.7), as well as 4 further channels for the 4 instruments assigned to orientation maps and another single channel for the roughness instrument. To proof the ideas of instruments based sonification we choose the high quality MIDI synthesizer **Native Instruments Kontakt 5** [332] along with high end sets of instruments.

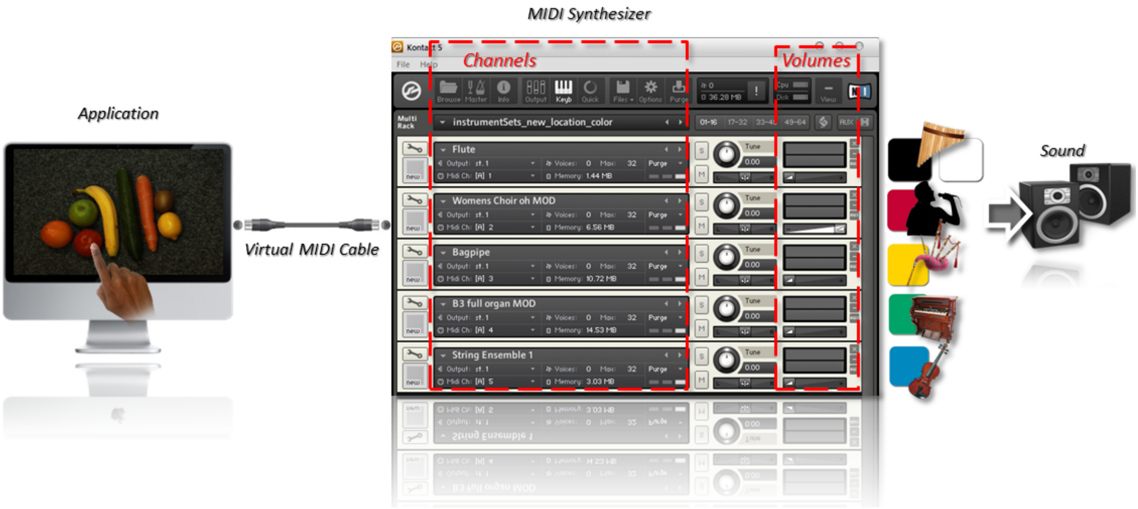


Figure 8.7: Left: A "virtual midi cable" software connects our framework with an external synthesizer. Right: 5 of the channels assigned to the 5 complementary instruments

To send MIDI messages out of our system, we use **RtMidi**, a set of object oriented classes, provided with the **Synthesis Toolkit (STK)** ([82]; [83]) (see appendix A.9) as an application programming interface (API) for real-time MIDI input and output. Within the RtMidi framework, MIDI input and output functionalities are separated into the **RtMidiIn** and **RtMidiOut** sub-classes, of which we only employ the latter.

The RtMidiOut class provides simple functionality to immediately send messages over a MIDI connection, similar to the following example:

```
RtMidiOut midiout;
std::vector<unsigned char> message(3);

// Open first available port.
midiout.openPort( 0 );

// Compose a Note On message.
message[0] = 144;
message[1] = 69;
message[2] = 90;

// Send the message immediately.
midiout.sendMessage( &message );
```

The syntax of “message” follows that of General MIDI. “message[0]” denotes the channel and command. In the example this would be to start playing a note on the first channel. “message[1]” stands for the key number on a standard piano keyboard, referring to an A at 440 Hz. “message[2]” finally denotes the velocity or “force” by which the note is played. Again, the concept comes from a piano keyboard. The faster that one strikes a piano keys, the louder the note will sound.

To deliver those messages from our framework to the midi synthesizer, we need a platform dependent in between software, known as **virtual midi cable** [305], connecting our framework and the synthesizer, as illustrated in figure 8.7.

When working with scales in MIDI, each note has to be triggered and released, which, again, is why a very short Attack- and Decay-and as well a short, as discussed in section 8.1, Release-Phase is essential to maintain a close-to-continuous signal. In contrast, mixing colors on a constant luminance takes place solely within the Sustain phase for arbitrary time - the note itself does not change.

A fundamental problem with using MIDI instruments is they inherit complex frequency spectra and, therefore, the risk to unwanted interferences when mixed. Furthermore, it is rather unlikely that any unique mapping could be found between a specific instrument and a particular color, as will be discussed in chapter 12.1. Finally, an external MIDI synthesizer is needed, what contradicts our vision to minimize the software, so that a blind user is able to process it on his notebook, tablet or mobile phone. Thus, in chapter 12.1 an intuitive color sonification concept, based on the one just presented, will be proposed which represents colors the way they are perceived visually by appropriate fundamental sound characteristics, instead of instruments, and comes without the need of an external MIDI synthesizer.

Chapter 9

User Studies

To scrutinize the possibilities of manual acoustical object recognition, we performed several experiments on two different groups of participants, following different motivations. Both groups, the congenital blind 54 year old academic as well as the 3 congenital blind teenagers had about 4 - 5 hours of introducing the system and training before tests. In contrast to the adult participant, the group of congenital blind 14 year old teenagers had little geometric understanding and sense of space. Thus, training also included fundamental lecturing about basic geometry, shadowing and perspective.
Generally, training was based on basic object shape recognition (as in figure 9.1 (left)), as well "Two-Alternative Forced Choice" (2AFC) training sets, (figure 9.1 (middle)). A specific object had to be recognized, in direct comparison to a similar colored object. Furthermore, the adult participant was asked to find a certain object within a more complex accumulation of different objects (figure 9.1 (right)).

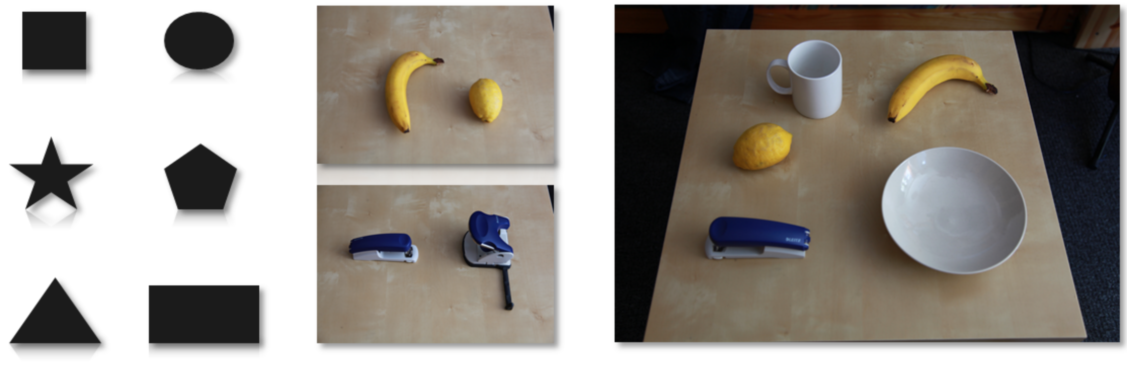


Figure 9.1: Left: Basic Shapes. Middle "Two-Alternative Forced Choice" training sets. Right: A "complex" scene

We then performed an experimental evaluation of our system to measure the teenagers' progress and to compare it with the results of the adult participant. Surprisingly, after training, the teenagers were able to perform the tests with approximately the same hit rates and times as our adult participant. We, therefore, hope that our system can not only support them in everyday life, but also help them to develop cognitive abilities in geometry and spatial orientation. As already mentioned, we intentionally work with congenital blind people, as we are specifically interested in evaluating our audible color space approaches on people who have never seen any colors at all. Unfortunately, the group of possible participants is thus significantly smaller as if we would work with early or late blind people.

9.1 Experiments

The “Two-Alternative Forced Choice” tasks are less applicable for a more qualitative evaluation of our system, as pure guessing works out in 50 percent of all cases. Hence in experiments I-IV, objects to recognize are displayed individually on the screen (as shown in 9.2). That reduces chance level (pure guessing) to 25 percent.

Experiment I - Object Recognition by Color only

The first experiment is about identifying one out of four elements (orange, tomato, apple and lemon as in 9.2) only by color while sonification of orientation maps and roughness is deactivated. Note that the target objects used for the task have the same spherical shape. In each of 60 trials, one of the 4 objects is selected at random and displayed at an arbitrary position on the touch screen. This is achieved by selecting one out of 40 images (10 per object, with the object in different positions) at random. The task of the participant is to find and name the object. In the evaluation (table 9.1 and figure 9.4), we focus on the time between the moment when the participant finds the object (which depends on where he starts and is, therefore, not very informative), and the moment when he names the object verbally to the experimenter. The average time to find an object is about 1.7 seconds. As mentioned, chance level (pure guessing) is 25 percent in this experiment.

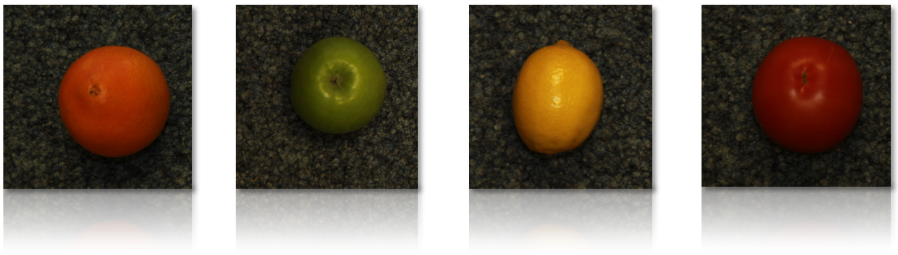


Figure 9.2: Setup for experiments I. Objects to recognize are displayed individually

Experiment II - Object Recognition by Color and Orientation Maps (Simplified)

The second experiment involves orientation maps and color. This time, the participant is to recognize one out of 7 objects (orange, tomato, apple, banana, cucumber, carrot, lemon), as shown in 9.3 (left), so both color and shape are important for correctly naming the object. Again, each element is presented individually (chance level: 14 percent) at arbitrary positions either vertical or horizontal. The database consists of 56 images (8 for each element, varying position and orientation). Again, times are measured between finding an naming the object verbally, as shown in table 9.1.

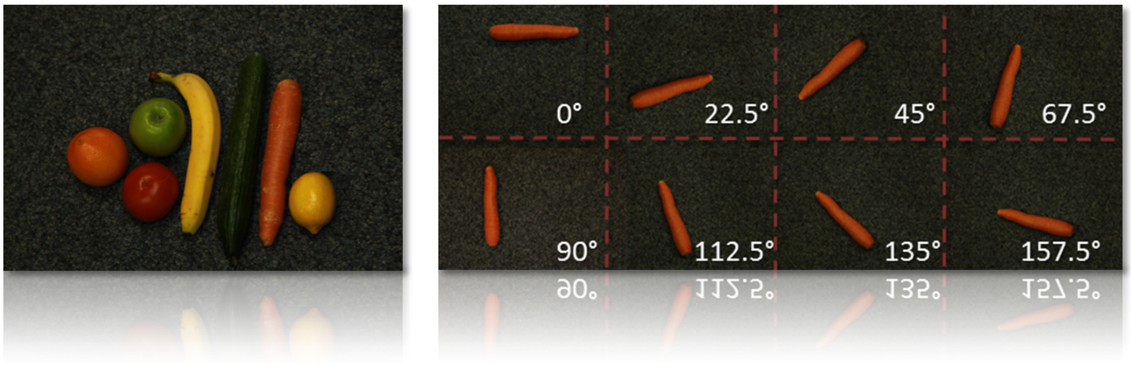


Figure 9.3: Left: An example of the training set for experiment IV. Right: An example of the possible orientations and locations of objects in experiment III

Experiment III - Object Recognition by Color and Orientation Maps (Complex)

The third experiment's setup is similar to that of experiment II, except that each element is presented, not only horizontal or vertical, but in one of eight orientations, as illustrated in figure 9.3 (right). The database again consists of 56 images (8 for each element, varying position and orientation). Again, times are measured between finding and naming the object verbally, as shown in table 9.1 and figure 9.4.

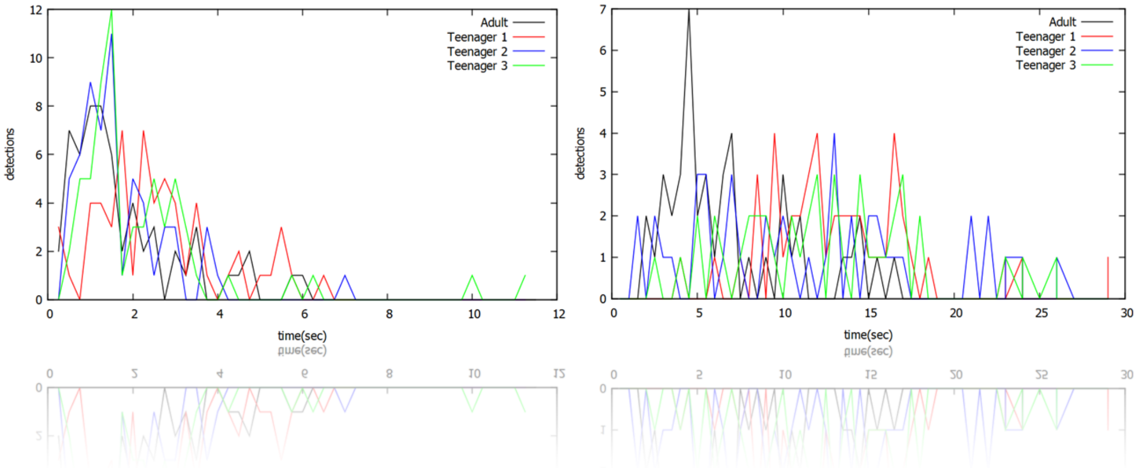


Figure 9.4: Left: Histogram of Experiments I. Right: Histogram of Experiment III. N elements (y axis) recognized in how many seconds (x axis) each

Experiment IV - Object Recognition within a Set of Objects

The fourth experiment is on recognizing an object within a set of other objects and is performed by the adult participant only. We present images such as the one shown in figure 9.3 (left) on the touch-screen. In our database of 7 images, we make sure that two objects of equal color (e.g. banana and lemon) are not be positioned next to each other. In each trial, an image is presented and, based on a random generator, the participant is told, which object he has to find. This time, we measure the overall time until the specific element is located an correctly identified.

Results

Experiment	Participant	Hitrate (% , N)	$\tilde{\text{X}}$ (sec.)	μ (sec.)	σ (sec.)
I	Adult	100 % (60/60)	1.3	1.8	1.4
	Teenager 1	91.6 % ,(55/60)	2.2	2.6	1.5
	Teenager 2	100 % ,(60/60)	1.3	1.7	1.1
	Teenager 3	93.3 % ,(56/60)	1.5	2.3	1.5
II	Teenager 1	88.8 % ,(40/45)	13.1	14.4	7.0
	Teenager 2	93.3 % ,(42/45)	9.2	9.5	5.4
	Teenager 3	88.8 % ,(40/45)	13.4	13.7	7.8
III	Adult	93.3 % ,(42/45)	5.6	7.0	3.9
	Teenager 1	88.8 % ,(40/45)	12.1	13.3	4.7
	Teenager 2	88.8 % ,(40/45)	10.1	11.4	6.6
	Teenager 3	93.3 % ,(42/45)	11.9	12.5	5.5
IV	Adult	100 % ,(45/45)	5.6	10.6	12.0

Table 9.1: Results of experiments I - IV. Hit rates and times (median $\tilde{\text{X}}$, mean μ, and standard deviation σ in seconds), for each trial and participant

Table 9.1 shows that all congenital blind participants, with very different educational backgrounds, were able to pass all the given tasks in experiments I - IV within reasonable time-spans. Further, the table proves that their results are clearly above chance level.

9.2 Discussion

The experiments raise the assumption that manual “auditory object recognition” can be done, although on a very limited complexity scale. Surprisingly, congenital blind teenagers, with no background in spatial geometry, were able to perform the tests with approximately the same hit rates and times as our adult participant. Hence, apart from object recognition the sonification of the orientation of edges might be of great benefit to develop the spacial understanding of congenital blind people.

Furthermore, the experiments give hope that the proposed color sonification approach that proves to be intuitive enough to be understood and applied by 4 congenital blind people of different backgrounds in very little time will prove also successful on a larger group of congenital blind people.

The extension of the system to extract specific entities within the image, as proposed in section 7.4, allows a more focused examination of specific parts in the image, similar to a person, e.g., grabs a single fruit out of a basket to have a closer look at it.

However, there is a number of interesting improvements, which led to a minimized (blind) user friendly stand-alone program, that will be described in detail in part IV:

- The sonification approach in this part relied on many MIDI instruments. Such instruments inherit complex frequency spectra and, therefore, the risk to unwanted interferences when mixed. Furthermore, an external MIDI synthesizer is needed, what contradicts our vision to minimize the software, so that a blind user is able to process it on his notebook, tablet or mobile phone.

- It would be interesting to develop an intuitive color sonification concept, based on the one presented in this part, which represents colors the way they are perceived visually by appropriate fundamental sound characteristics.

- The limits of manual acoustical object recognition could be handled, carefully employing sophisticated computer vision and machine learning techniques to classify certain regions within an image and individually selecting specific low-level features to compute on certain regions.

Part IV

Auditory Image Understanding

Chapter 10

Motivation

The implementation in part III is dedicated to scrutinize the possibilities and limitations of some sort of audible object recognition. Therefore, rather fundamental image characteristics are extracted, such as colors, edges of various orientations (i.e. orientation maps) as well as a measure of textural roughness. MIDI instruments are assigned to represent such features. Basic recognition tests, described in chapter 9 revealed that audible object recognition can be performed, although on a rather limited complexity scale.

Hence, we considered leveraging computer vision and machine learning algorithms to derive and sonify image information on many levels, ranging from low-level such as color information to high-level, as for example object recognition. Machine learning techniques could be successfully employed to even pre-select the extraction of specific low-level features in certain areas. Still, the results of these algorithms remain tied to the image pixel where the feature occurs, so the user always knows locations of objects and structures. Incorporating the imaginative capabilities of a blind person's brain as a fundamental element of the process proves to be a promising combination for more sophisticated tasks, such as scene understanding. A system that allows such an "auditory image understanding" could be used by the visually impaired to analyze images that they find on the internet, making it more accessible, and also for personal photos that their friends or loved ones want to share with them. [7]

As the system presented in part III, it could be also utilized to help congenital blind people to develop spacial understanding, although on a different level. As the user's imagination is trained in part III for the shape, orientation and perspective of single objects, now, whole scenes and spatial relations of objects within are to be visualized. In this context, the direct perceptual access becomes most valuable.

[7] The work on "auditory image understanding" has in part been published in [20]

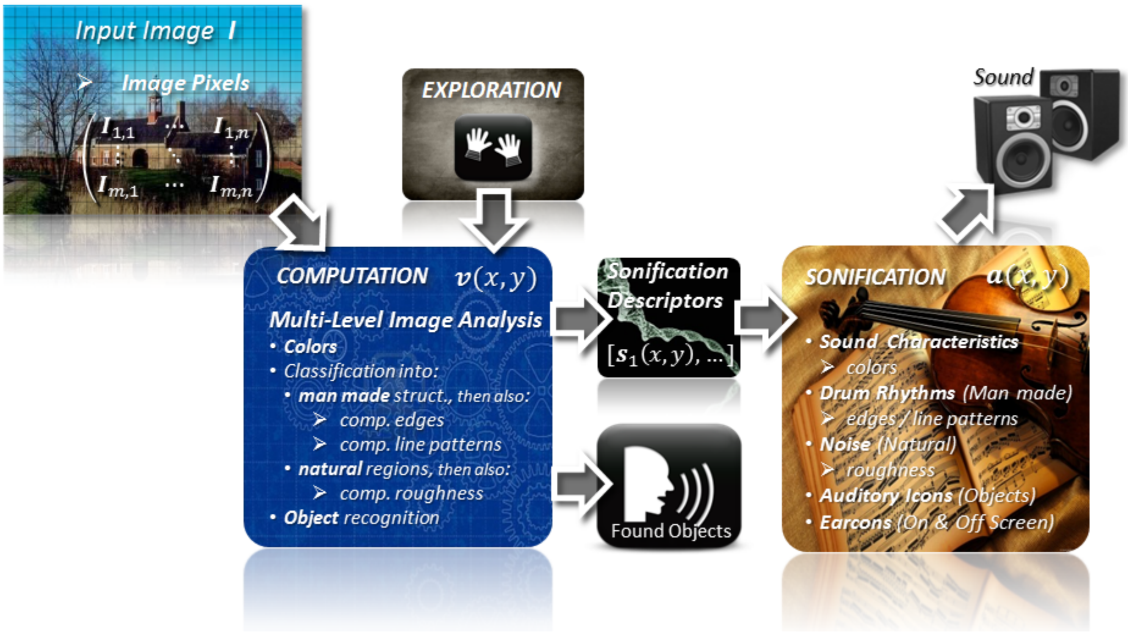


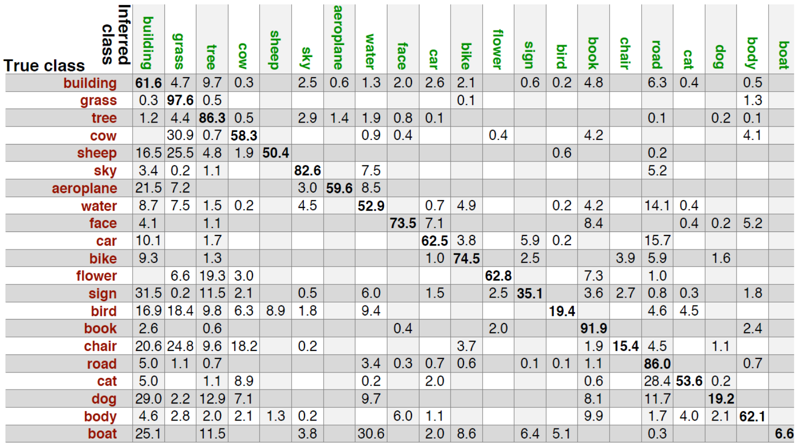
Figure 10.1: Implementations of the computation and sonification modules of the modular computer vision sonification model for the task of auditory image understanding

Additionally, within this part of the work, the color sonification will be refined, due to problems we observed with the one presented in part III, discussed in section 9.2. The user feedback that we receive for the system, presented in this part, indicates that visually impaired people appreciate the fact that they obtain more than an abstract verbal description and that images cease to be meaningless entities to them. Figure 10.1 gives an overview of the specific implementations of the computation and sonification module of the modular computer vision sonification model, presented in chapter 4, that need to be done for the task of "auditory image understanding".

Chapter 11

Multi-Level Image Analysis

As the system in this part is designed to allow visually impaired people to analyze images that they find on the internet or personal photos from their friends it is crucial to extract (and later sonify) specifically that sort of information, which is commonly present in these images. This information might include, e.g., landscapes, man made structures, animals, people, cars or every day objects. This kind of image understanding is a task of primary importance for a wide range of practical applications and has been topic of considerable research. One important step towards understanding an image could be to perform a “full-scene labeling” also known as a “scene parsing”, which consists in labeling every pixel in the image with the category of the object it belongs to. Scene parsing has been addressed with a variety of methods in recent years, most of which rely on the usage of Markov or Conditional Random Fields ([31]; [272]) or other types of graphical models to account for context and ensure the consistency of the labeling ([177]; [285]; [383]; [197]; [404]; [264]; [328]; [461]; [520], [433]; [137]). Figure 11.1 illustrates some accuracies of the scene parsing model on the MSRC 21-class database, proposed in [433].



True class / Inferred class	building	grass	tree	cow	sheep	sky	aeroplane	water	face	car	bike	flower	sign	bird	book	chair	road	cat	dog	body	boat
building	**61.6**	4.7	9.7	0.3		2.5	0.6	1.3	2.0	2.6	2.1		0.6	0.2	4.8		6.3	0.4		0.5	
grass	0.3	**97.6**	0.5								0.1									1.3	
tree	1.2	4.4	**86.3**	0.5		2.9	1.4	1.9	0.8	0.1							0.1		0.2	0.1	
cow		30.9	0.7	**58.3**				0.9	0.4			0.4			4.2					4.1	
sheep	16.5	25.5	4.8	1.9	**50.4**									0.6			0.2				
sky	3.4	0.2	1.1			**82.6**		7.5									5.2				
aeroplane	21.5	7.2				3.0	**59.6**	8.5													
water	8.7	7.5	1.5	0.2		4.5		**52.9**		0.7	4.9			0.2	4.2		14.1	0.4			
face	4.1		1.1						**73.5**	7.1					8.4			0.4	0.2	5.2	
car	10.1		1.7							**62.5**	3.8		5.9	0.2			15.7				
bike	9.3		1.3							1.0	**74.5**		2.5			3.9	5.9		1.6		
flower		6.6	19.3	3.0								**62.8**			7.3		1.0				
sign	31.5	0.2	11.5	2.1		0.5		6.0		1.5		2.5	**35.1**		3.6	2.7	0.8	0.3		1.8	
bird	16.9	18.4	9.8	6.3	8.9	1.8		9.4						**19.4**			4.6	4.5			
book	2.6		0.6						0.4			2.0			**91.9**					2.4	
chair	20.6	24.8	9.6	18.2		0.2					3.7				1.9	**15.4**	4.5		1.1		
road	5.0	1.1	0.7					3.4	0.3	0.7	0.6		0.1	0.1	1.1		**86.0**			0.7	
cat	5.0		1.1	8.9				0.2		2.0					0.6		28.4	**53.6**	0.2		
dog	29.0	2.2	12.9	7.1				9.7							8.1		11.7		**19.2**		
body	4.6	2.8	2.0	2.1	1.3	0.2			6.0	1.1					9.9		1.7	4.0	2.1	**62.1**	
boat	25.1		11.5			3.8		30.6		2.0	8.6		6.4	5.1			0.3				**6.6**

Figure 11.1: Accuracy of segmentation for the MSRC 21-class database in [433]. Confusion matrix with percentages row-normalized. Overall pixel-wise accuracy in [433] is 72.2 %

A challenging task of scene labeling is that it combines multi-label recognition and segmentation in a single process. As figure 11.1 indicates the incorporation of multiple classes to recognize favors the occurrence of miss-classifications. Interestingly, although object class recognition fails the segmentation might still be accurate, as illustrated in figure 11.2. However, in the context of full-scene labeling for the visually impaired, it is crucial that no not-present classes are introduced during recognition.

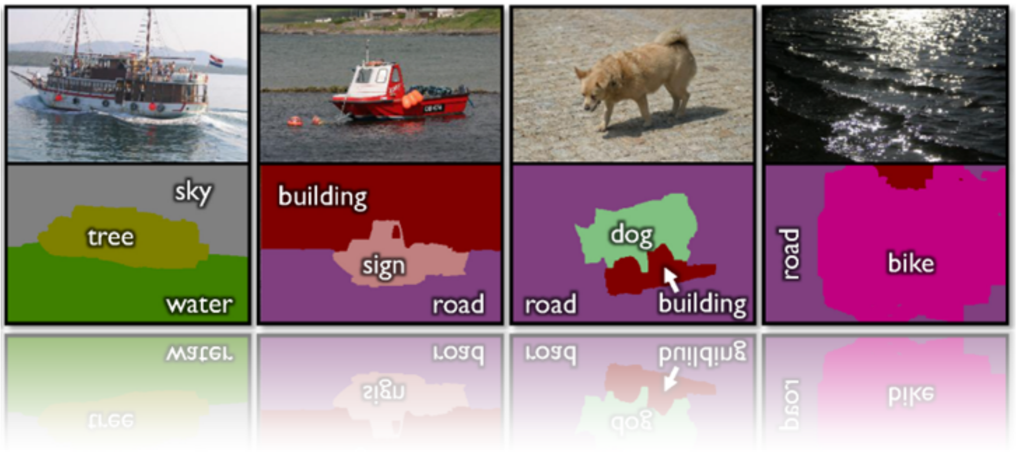


Figure 11.2: Examples where recognition works less well. Input test images with corresponding color-coded output object-class maps. Even when recognition fails, segmentation may still be quite accurate. Picture taken from [433]

Another great challenge with scene labeling, especially in our context, is to maintain near real-time performance while not sacrificing accuracy. Table 11.1 illustrates the dependency between recognition accuracy and computation times of 3 state of the art scene labeling approaches performed on the Stanford Background data set [177], which contains images of outdoor scenes composed of 8 classes.

	Pixel Accuracy (%)	**Class Accuracy (%)**	**Computation Time (sec.)**
Munoz et al. 2010 [328]	76.9	66.2	12
Lempitzky et al. 2011 [285]	81.9	72.4	> 60
Farabet et al. 2013 [137]	81.4	76.0	60.5

Table 11.1: Performance of 3 scene parsing systems on the Stanford Background dataset: per-pixel / average per-class accuracy. The third column reports compute times, as reported by the authors

Furthermore, the selection of classes to detect might be challenging, if no prior knowledge about the images to be explored, exist. To avoid the introduction of false object classes and support the modular design of our system, for our specific application, we separate the process of image labeling and object detection. Object recognition, described in section 11.4, is based on two state of the art approaches, followed by learning based approach that we propose to filter correct from incorrect detections. To make scene labeling applicable on a more general level, without the risk of introducing false object classes, we make use of an idea initially proposed by Kumar and Hebert [269] and further developed in ([269]; [268]; [265]; [345]; [343]; [533]; [266]; [463]) that uses binary classification for man made structure detection in natural scenes, as described in section 11.2. In this approach, images are subdivided into rectangular patches and the classification of an image consists of determining the correct labels of each patch in an image. This procedure, therefore, does not represent a pixel-exact labeling as it “quantizes” the image and its labeling, which in general might be undesirable. However, in the context of providing information to visually impaired, the continuous range visual data clearly demands to much of them and quantizations have to be applied within several steps of the process. Furthermore, the loss in continuity is compensated by an increase in robustness and generalizability. In the following sections we present a novel probabilistic graphical model, called **Dual Support Vector Fields** [8] and an advanced feature set as an alternative to the approach by Kumar [266]. Due to the modular design of our modular computer vision sonification model, the proposed approaches can of course be extended or even exchanged at any time. Our specific research contributions to the field of machine learning and computer vision in this part of the thesis can be summarized as:

- A novel type of probabilistic graphical model, called **Dual Support Vector Fields** for man made structure detection or other labeling problems that deal with spatial dependencies.

- A novel feature set for man made structure detection that goes beyond low level features.

- An algorithm (and feature set) to verify true or discard false object detections before sonification to avoid con- fusion on the side of the blind user, who can not check for a correct detection visually.

- Due to their design, both proposed algorithms can be also employed in other applications than “auditory image understanding”, e.g., for fully-automated computer vision systems.

[8] The proposed Dual Support Vector Fields have been published in [19]

Auditory Scene Labeling

Fascinatingly, the user is “incorporated ” in the image understanding process. Although we only classify natural and man made regions, during exploration, the user can utilize detected man made structures or specific natural regions as reference points to classify other natural regions by their individual location, color and texture. Figure 11.3 illustrates graphically, how congenital blind participants within the user studies in chapter 13 employ that strategy successfully to interpret and understand a scene. Regions have been labeled according to the verbal scene interpretation given below (see table 11).

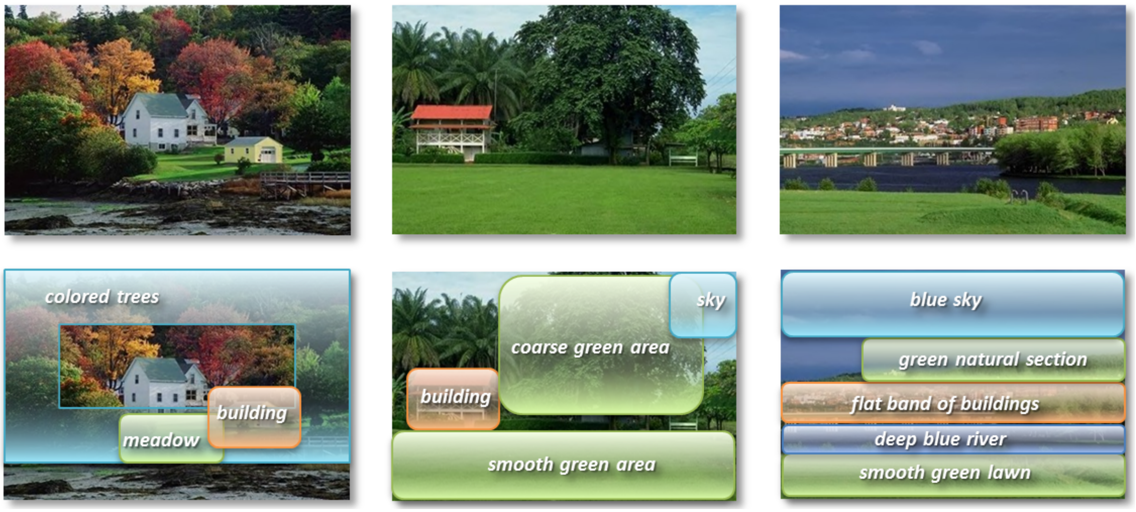


Figure 11.3: Some examples of user based scene understanding. Top row: Original images. Bottom row: Regions “labeled” due to human user classification

Verbally descripted Scene Interpretation
There is a yellow building. A green area beneath the building would presumably by some sort of meadow. The different colored spots surrounding the meadow and the building might be colored trees. [teenage participant on figure 11.3 (left)]
The lower part of the image from left to right is some intensive green area. There is a strong contrast in roughness on the right from the smooth green area to a coarser green area in the mid-section. There is some light blue spot, which will be sky, on the top right corner and salient red building on the left. [adult part. on fig. 11.3 (middle)]
The lower part of the image from left to right is smooth green, such as a lawn. Then there is a deep blue stripe which is supposedly some sort of water, such as a river. Above the river is a very flat band of buildings, followed by some green natural section. The top region is blue, presumably sky. [adult part. on fig. 11.3 (right)]

11.1 Image Pre-Processing

As in part III, section 7.1, Bilateral Filtering is applied to smooth pixel values $I(x, y)$ of an input image $\boldsymbol{I}$ before finally storing them within augmented visual pixels $\boldsymbol{v}(x, y)$. However, as we are working with much more delicate image data as in part III (see figure 11.4 (left)), we refrain from iterating and only apply bilateral filtering once (figure 11.4 (right)).

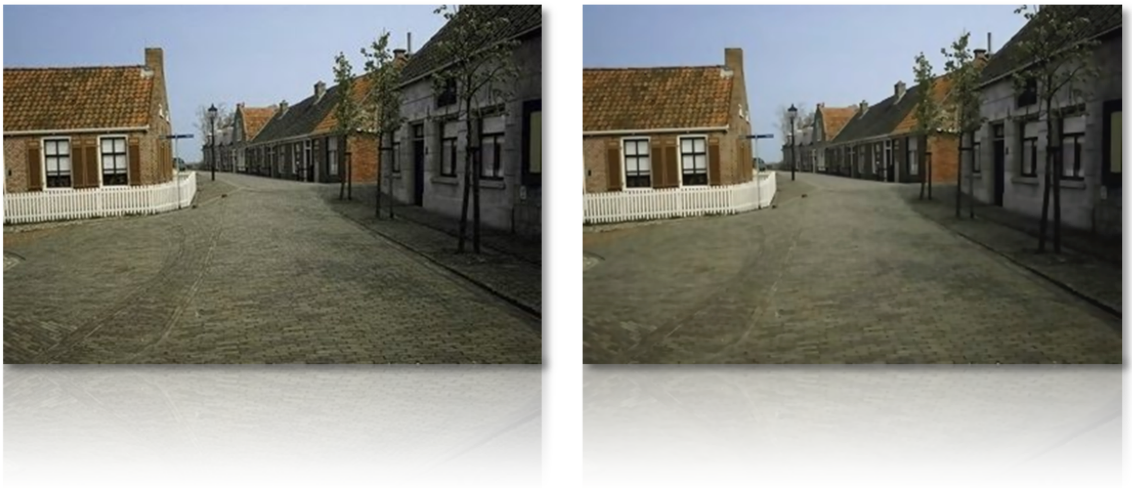


Figure 11.4: Left: Input image $\boldsymbol{I}$, Right: Bilateral filtered image $\boldsymbol{I}_{bf}$

11.2 Detection of Man Made Structures in Natural Scenes

Images that contain man-made structures exhibit strong contextual dependencies in the form of spatial interactions among image elements. Neighboring pixels are likely to have similar class labels, and different regions appear in restricted spatial configurations. Additionally, detection is challenging due to ambiguities in the appearance of the visual data. E.g., a patch corresponding to a tree might appear, on a local scale, very similar to that belonging to a building, as illustrated in figure 11.5. Thus, the use of context can help alleviate this problem significantly.

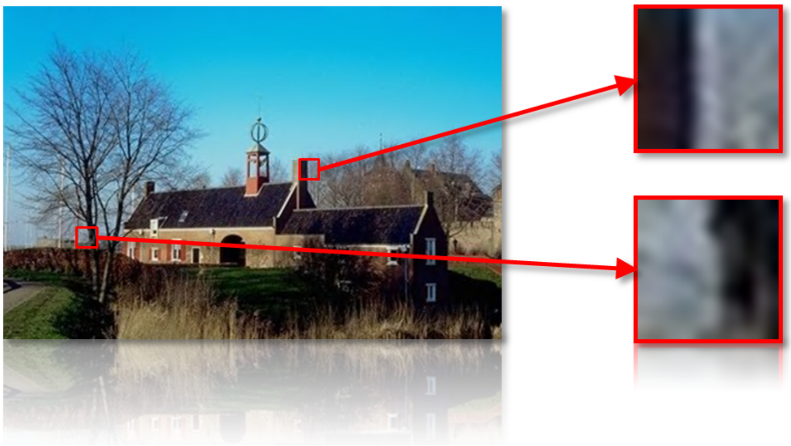


Figure 11.5: Tree and building regions look similar. Context can help resolve these ambiguities

Modeling these spatial structures is crucial to achieve good classification accuracy, and help alleviate ambiguities. As just mentioned, there has been done considerable research on contextual models that exploit spatial dependencies between objects, such as Markov random fields (MRFs) and conditional random fields (CRFs) for probabilistic modeling of local dependencies, e.g., in ([198]; [465]; [331]; [466]; [372]; [373]; [416]).
In [265], Kumar states that the detection of man-made structures from a single static ground-level image is still a non-trivial challenge because of three main reasons:

- *Realistic views of a structured object captured from a ground-level camera are unconstrained unlike the aerial views [259], which complicates the use of predefined models or model-specific properties in detection.*

- *No motion or depth information is available, precluding the use of geometrical information pertaining to the structure.*

- *Images of natural scenes contain large amount of clutter, and the edge extraction is very noisy. This makes the computation of the image primitives such as junctions, angles etc., which rely on explicit edge or line detection, prone to errors.*

The discriminative conditional random field framework proposed by ([268]; [265]; [266]) allows to relax the strong assumption of conditional independence of the observed data generally used in MRF frameworks. This is crucial, as the lines or edges at spatially adjoining regions in man-made structures follow some inherent rules of organization rather than being completely random. These models are by nature non-causal and are typically represented by undirected graphs.

Modeling Spatial Dependencies in Natural Images

The general representation of an image as a **Conditional Random Field (CRF)** [272] will follow the notation of Kumar and Hebert in ([268]; [265]; [266]). Thus, images are subdivided into rectangular patches, called "sites" of 16×16 pixels each, and the classification of an image consists of determining the correct labels of each site s_i. Prior to proceeding to the actual model of an image the following definitions and notations for a Conditional Random Field shall be restated for clarity, given by Lafferty et al. (Lafferty et al., 2001). Let the observed data from an input image be given by $\boldsymbol{y} = \{\boldsymbol{y}_i\}_{i \in S}$ where $\boldsymbol{y}_i$ is the data from i^{th} site and $\boldsymbol{y}_i \in \mathcal{R}^c$. The corresponding labels at the image sites are given by $\boldsymbol{x} = \{x_i\}_{i \in S}$.

Definition:

> *Let $G = (V, E)$ be a graph such that x is indexed by the vertices of G. Then (x, y) is said to be a conditional random field if, when conditioned on y, the random variables x_i obey the Markov property with respect to the graph: $p(x_i|y, x_{V\{i\}}) = p(x_i|y, x_{\mathcal{N}_i})$, where $V\{i\}$ is the set of all nodes in G except the node i, $\mathcal{N}_i$ is the set of neighbours of the node i in G, and x represents the set of labels of the nodes in set ω.*

When modeling an image using conditional random fields, the set of image sites corresponds to the set of vertices within the graphical model, illustrated in figure 11.6. Accordingly, edges correspond to the connections between neighboring sites.

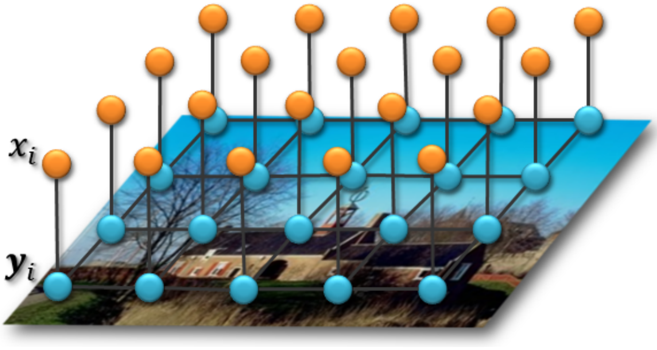


Figure 11.6: Modeling an image using CRF. $\boldsymbol{y}_i$ is the data from i^{th} site and x_i the corresponding label

In their CRF model for images, Kumar uses the Hammersley-Clifford theorem [272] and the assumption that only pairwise clique potentials are non-zero, i.e., only immediate neighbors interact [266]. From this they obtain a conditional distribution over the labels given observations $\boldsymbol{y}$ defined by:

$$p(x|y) = \frac{1}{Z} \exp \left(\sum_{i \in V} A(x_i, \boldsymbol{y}) + \sum_{i \in V} \sum_{j \in \mathcal{N}_i} I(x_i, x_j, \boldsymbol{y}) \right) \tag{11.1}$$

Z denotes a normalizing factor referred to as the partition function. For our application we are concerned with binary classification only, namely $x_i \in 1, 1$, indicating a site s_i is natural or man-made, respectively. Kumar and Hebert refer to the unary potential $A(x_i, \boldsymbol{y})$ and the pairwise potential $I(x_i, x_j, \boldsymbol{y})$ as the **Association** and **Interaction** potentials, respectively.

The association potential, $A(x_i, \boldsymbol{y})$, can be regarded as a measure of how likely some image site s_i will take label x_i given a series of features $\boldsymbol{y}$, computed at that particular site and leaving out any effects of other sites within the image. In contrast, the interaction potential can be interpreted as a measure of interactions of the labels at neighboring sites s_i and s_j given the image observations.

Leaving out the interaction term ($I(x_i, x_j, \boldsymbol{y}) = 0$) reduces the model to the **Logistic Classifier** of an image, which does not incorporate any interaction between neighboring image sites. Then, $A(x_i, \boldsymbol{y})$ is modeled using a local discriminative model that outputs the association of the site s_i with class x_i as:

$$A(x_i, \boldsymbol{y}) = \log p(x_i | \boldsymbol{y}_i) \tag{11.2}$$

$p(x_i | \boldsymbol{y}_i))$ is the local class conditional at site s_i. This form allows one to use an arbitrary domain-specific probabilistic discriminative classifier for a given task. This can be seen as a parallel to the traditional MRF models where one can use any local generative classifier to describe the unary potential. One potential selection of $p(x_i | \boldsymbol{y}_i)$ might be Generalized Linear Models (GLM) that have been used in statistics to model the class posteriors given the observations [312]. Kumar and Hebert propose the logistic function as a link in the GLM and define the local class conditional as:

$$p(x_i = 1 | \boldsymbol{y}_i) = \sigma(w_0 + \boldsymbol{w}^T \, \boldsymbol{y}_i) = \frac{1}{1 + e^{-(w_0 + \boldsymbol{w}^T \, \boldsymbol{y}_i)}} \tag{11.3}$$

w_0 and $\boldsymbol{w}$ are the parameters of such a reduced model, corresponding to the length of the observed feature data $\boldsymbol{y}$. The specific form of $p(x_i | \boldsymbol{y}_i)$ yields a linear decision boundary within the feature space spanned by vectors $\boldsymbol{y}_i$.

To extend the logistic model to induce a non-linear decision boundary, Kumar and Hebert introduce a transformed feature vector $\boldsymbol{f}(\boldsymbol{y}_i)$ at each site s_i, employing arbitrary non-linear functions, which might be seen as a sort of mapping of the original feature vector into a high dimensional space, yielding $p(x_i|\boldsymbol{f}(\boldsymbol{y}_i))$.

The idea of using Kernels to avoid such an explicit, less efficient, mapping in discrimative conditional random fields has been introduced and a employed for Protein Secondary Structure Prediction by Lafferty et al. [271]. The concept has also been implemented in combination with Import Vector Machines in [396].

The first element of the each feature vector $\boldsymbol{f}(\boldsymbol{y}_i)$ is set equal to 1 to accommodate the constant parameter w_0. Further, since $x_i \in \{-1, 1\}$, the probability in (11.3) can be compactly expressed as:

$$p(x_i|\boldsymbol{y}_i) = \sigma(x_i\, \boldsymbol{w}^T \boldsymbol{y}_i) \tag{11.4}$$

And therefore the association potential can be written as:

$$A(x_i|\boldsymbol{y}_i) = \log \sigma(x_i\, \boldsymbol{w}^T \boldsymbol{y}_i) \tag{11.5}$$

Such final transformation ensures that the *CRF* is equivalent to a Logistic Classifier if the interaction potential in (11.1) is set to zero.

Non-Linear Support Vector Machines

Instead of introducing a transformed feature vector $\boldsymbol{f}(\boldsymbol{y}_i)$ at each site s_i using non-linear functions, we propose to employ non-linear Support Vector Machines (SVMs) [417] as association potential (i.e., $A(x_i, \boldsymbol{y}) = \log p_{svm}(x_i|\boldsymbol{y}_i)$), as they inhere appealing theoretical properties, as discussed in appendix A.7, and tend to outperform GLMs, especially when the classes in the feature space overlap [430]. Beneficially, the CRF framework allows for a flexible selection of association potentials. However, the decision function computed by SVMs measures distances to the decision boundary, while the association potential requires a posterior probability function. Thus, we utilize the approach described in [512] and provided by [71] to convert the decision function to a posterior probability function. The idea to extend SVMs to consider spatial correlations, based on CRFs, has been initially proposed for linear SVMs by Lee et al. [280] and successfully applied,e.g., in medical image segmentation [279]. SVMs have also been harnessed in the context of Markov Random Fields [511].

Dual Support Vector Fields

The CRF models represents an extension of the Markov Random Field (MRF) ([290]; [322]; [248]), which itself is a simple extension of the Logistic Classifier. For the homogeneous MRF, the interaction potential is defined as $I(x_i, x_j, \boldsymbol{y}) = v\, x_i\, x_j$, for a scalar parameter v, which penalizes every dissimilar pair of labels. Such a form of interaction favors piece-wise constant smoothing of the labels without considering discontinuities in the observed data explicitly. In contrast, the CRF framework, proposed by Kumar and Hebert, computes the interaction potentials as a function of all observations $\boldsymbol{y}$. In addition to model pairwise relational information between sites, such a rather data-dependent smoothing can compensate for the errors in describing the association potential. To model these pairwise terms, the main idea is to have identical labeling at a pair of sites for which the observations support such a hypothesis. Kumar chooses the interaction potential to be $I(x_i, x_j, \boldsymbol{y}) = x_i\, x_j\, \boldsymbol{v}^T\, \boldsymbol{\mu}_{ij}(\boldsymbol{y})$, with $\boldsymbol{\mu}_{ij}(\boldsymbol{y})$ being the concatenated feature vectors $\boldsymbol{f}(\boldsymbol{y}_i)$ and $\boldsymbol{f}(\boldsymbol{y}_j)$.

In contrast, we introduce a novel type of interaction potential based on Support Vector Machines as well:

$$I(x_i, x_j, \boldsymbol{y}) = v\, x_i\, x_j\, (1 - \| p_{svm}(x_i{=}1|\boldsymbol{y}_i) - p_{svm}(x_j{=}1|\boldsymbol{y}_j) \|)$$

with a scalar parameter v. The proposed distance measure of the nonlinear SVM responses in the interaction potential encourages label continuity, while discouraging discontinuity. It further reduces learning of additional parameters to computing v only.

As our novel CRF model incorporates Support Vector Machines in both, $A(x_i, \boldsymbol{y})$ as well as $I(x_i, x_j, \boldsymbol{y})$, we name our approach **Dual Support Vector Fields (DSVF)**.

Feature Set

So far, all major approaches ([269]; [268]; [265]; [345]; [343]; [533]) to explicitly detect man made structure from ground-level natural images , refer to the feature set initially proposed by Kumar and Hebert in [269]. Although the design of our own feature set is in some ways inspired by their approach, we strive to engineer sophisticated features to further reduce the level of ambiguity. Thus, we now describe the details of our novel feature set.

Smoothed Histograms of Gradient Orientations

As image pre-processing, bilateral filtering is applied to an input image $\boldsymbol{I}$, as it smooths the image while preserving dominant edges. Subsequently, the bilateral filtered image $\boldsymbol{I}_{bf}$ is converted to HSL color space, yielding $\boldsymbol{I}_{bf/HSL}$. To extract edges, Gabor wavelet transform [535] is performed on the lightness channel of $\boldsymbol{I}_{bf/HSL}$. As in part III, Gabor wavelets of the form:

$$\psi_{\varphi,\nu}(z) = g_{\varphi,\nu,\sigma}(z)\left[e^{i\,k_{\varphi,\nu}\,(z)} - e^{-\frac{\sigma^2}{2}}\right]$$

with the Gaussian envelope:

$$g_{\varphi,\nu,\sigma}(z) = \frac{||k_{\varphi,\nu}||^2}{\sigma^2}\, e^{-\frac{||k_{\varphi,\nu}||^2\,||(z)||^2}{2\,\sigma^2}}$$

are applied in 32 orientations φ from $-90°$ to $90°$ with an angular difference of $5.625°$ and a rather small sized kernel ($\nu = 0$ and $\sigma = \frac{\pi}{2}$) to account for the delicate structures in the images. Again, $z = (x, y)$ indicates a point with x, the horizontal coordinate and y, the vertical coordinate. Subsequently, non-maximum suppression ([338]; [106]; [335]; [67]) is utilized to thin edges (figure 11.7).

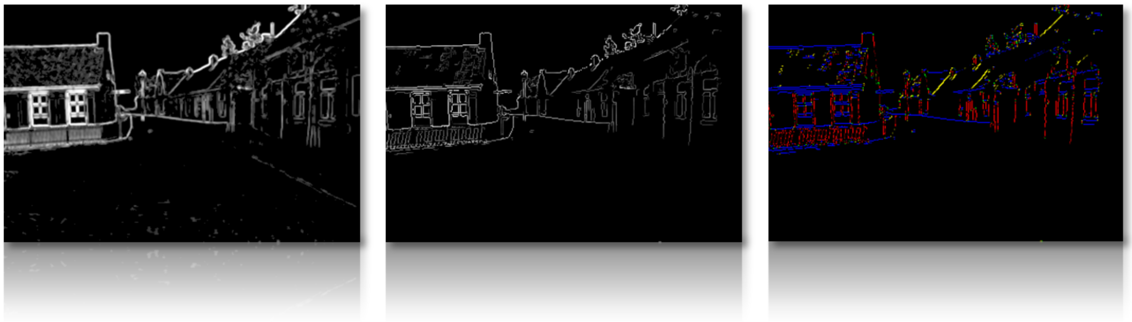


Figure 11.7: Gabor Voting. Left: Edge gradients intensity coded from 0 to 1. Middle: Gabor voted image after non-maximum supression. Right: Edge orientations color coded $\approx 0°$ (blue),$\approx 90° \vee -90°$ (red), $\approx 45°$ (yellow) and $\approx -45°$ (green)

Thereafter, as in [269], for each image site s_i, the gradients contained within a window w_c at different scales c around the center of s_i are combined to yield a histogram $\boldsymbol{H}_{s_i}(c)$ (per scale c) over gradient orientations.

We employ five scales, instead of three as in [269], $c \in \{16 \times 16, 32 \times 32, 48 \times 48, 56 \times 56, 64 \times 64\}$. Instead of weighting each count by the gradient magnitude at that pixel as in [269], we simply increment the counts in the histograms. This is due to the observation, that occurring high magnitude gradients, which are to be captured using "weighted histograms", might indicate a building, they may, however, also result from strong edges that occur in nature, e.g., around the trunk of a tree. Such an observation is confirmed by Rees [379] in his evaluations of the feature set defined in [269].

Once the histograms are computed, Kernel Smoothing [471] is employed to alleviate the problem of hard binning of the data. With $N = 32$ being the total number of bins in the histogram, h_i the count of the i^{th} bin of $\boldsymbol{H}_{s_i}(c)$, and a symmetric positive kernel smoothing function $K(x)$ with bandwidth b, the smoothed bin counts are given by:

$$h'_j = \frac{\sum_{i=1}^{N} K((h_j - i)/b)\, h_i}{\sum_{i=1}^{N} K((h_j - i)/b)} \quad \text{with} \quad K(x) = \frac{1}{e^{x^2}} \tag{11.6}$$

Kumar and Hebert [268] suggest $b = 0.7$ to restrict smoothing only to neighboring histogram bins, yielding smoothed histograms $\boldsymbol{H'}_{s_i}(c)$. We then employ a TABU search ([170]; [59]), a local search method, to find local maxima above mean and Insertion Sort ([249]; [85]) to find and sort orientations φ of found peaks in each $\boldsymbol{H'}_{s_i}(c)$ from highest to smallest. Thus, we can detect the orientation φ_{∇_1} of the highest bin h'_{∇_1}, i.e., the most dominant gradient within the image site. φ_i is then mapped from 0 to 1 using a sinusoidal function. The mapping slightly favors the occurrence of vertical edges of almost 90°, as those tend to often occur in man made structures. Such a mapping has been previously used in the context of perceptual grouping of pre-specified image primitives [259]. The feature is computed for all scales c. Note that this feature is the only one in common with [269]. Additionally, we use the raw value of h'_{∇_1} along with $\sin(\varphi_{\nabla_1})$ as feature.

Junctions & Line Patterns

Man made structures, in general, exhibit a great amount of parallel lines as well as near right angle junctions. We harness such properties as a measure of discriminancy, defining specialized features to capture them. Kumar and Hebert [269] suggest evaluations of the histograms $\boldsymbol{H'}_{s_i}(c)$ using heaved central-shifted moments of various orders to capture what they call the average "structuredness" in image sites. However, these moment based features might not necessarily be the obvious choice for the search for man-made structures, as the presence of high magnitude gradients within an image site alone, does not suffice to constitute a man-made structure, as edges exist in nature too. Additionally, such moment based features do not yield information about differences in orientations between the high magnitude gradients capture, which is why Kumar and Hebert suggest the use of angular differences between the first two highest local maxima in each $\boldsymbol{H'}_{s_i}(c)$. To get a more qualitative measure about the number of found gradients as well as orientational differences which incorporates all found peaks, we propose a different set of features. For scales $c \in \{2, 3, 4, 5\}$ we compute the number n_∇ of dominant gradients per each image in s for each $\boldsymbol{H'}_{s_i}(c)$. A found peak in $\boldsymbol{H'}_c$ is defined as a "dominant" gradient, if its value is at least 60 % of that of the highest gradient h'_{∇_1}. Additionally, we compute the average angle $\overline{\Delta\varphi_\nabla}$ between all found dominant gradients:

$$\overline{\Delta\varphi_\nabla} = \begin{cases} \|\sin(\frac{1}{n_\nabla \times (n_\nabla - 1)} \sum_{i,j}^{n_\nabla \times n_\nabla} \|\varphi_{\nabla_i} - \varphi_{\nabla_j}\|)\|, & \text{if } i \neq j \\ -1, & \text{otherwise} \end{cases}$$

Additionally, we perform an analysis on line junctions and repetitive line patterns indicating significant or repeating building elements such as doors or windows. This analysis is inspired by human grating cells, described appendix B.1, which were discovered in 1992 by Von der Heydt et al. [492]. Briefly, grating cells respond vigorously to gratings of bars of appropriate orientation, position and periodicity. In contrast, these cells respond weakly or not at all to single bars.

First, line segments are detected applying the Line Segment Detector (LSD) ([183]; [493]) to the l channel of the bilateral filtered image $\boldsymbol{I}_{bf/HSL}$, as visualized in figure 11.8 (left). LSD is a state of the art linear-time line segment detector giving sub-pixel accurate results. It is based on Burns, Hanson, and Riseman's method [62] and designed to work without any parameter tuning. LSD was chosen after comparing results against Progressive Probabilistic Hough Transform (PPHT) ([157]; [307]) and Line Segment Detection Using Weighted Mean Shift (LSWMS) [337]. The PPHT harnesses differences in the fraction of votes crucial to reliably detect lines with different numbers of supporting points, other than the Probabilistic Hough Transform, where the standard Hough Transform ([222]; [219]) is performed on a pre-selected fraction of input points. LSWMS on the other hand is designed to work unsupervised without tuning of input parameters as well. It uses sampling strategy that sequentially proposes points on the image that likely belong to line segments and a subsequent line growing algorithm based on the Bresenham algorithm that is combined with an altered version of the mean shift algorithm to give accurate line segments as well as increasing noise robustness. In our experiments LSD proved to be most reliable for detecting smaller line segments, which is crucial in the context of the delicate structures we are dealing with in this part.

Results of the line segment detection are quantized and grouped into 8 orientations of 22.5° angular difference between -90° and 90°. We then apply two filter approaches on the grouped lines. First, line segments that are in length below a specific threshold, are excluded. Thresholds for nearly horizontal and vertical lines are slightly smaller than that for orientations in between, as also small vertical and horizontal lines bear important information in the context of man made structure detection. Thus, this differentiation in the thresholding procedure works as a task specific noise filter as well.
All lines that do not lie on or near to a gradient of almost similar orientation (figure 11.8 (middle)), extracted by the Gabor wavelet transform, are discarded as well (figure 11.8 (right)). This is due to observations that even very small intensity variations that were not detected as a gradient in previous edge extraction, might invoke a line due to the LSD.

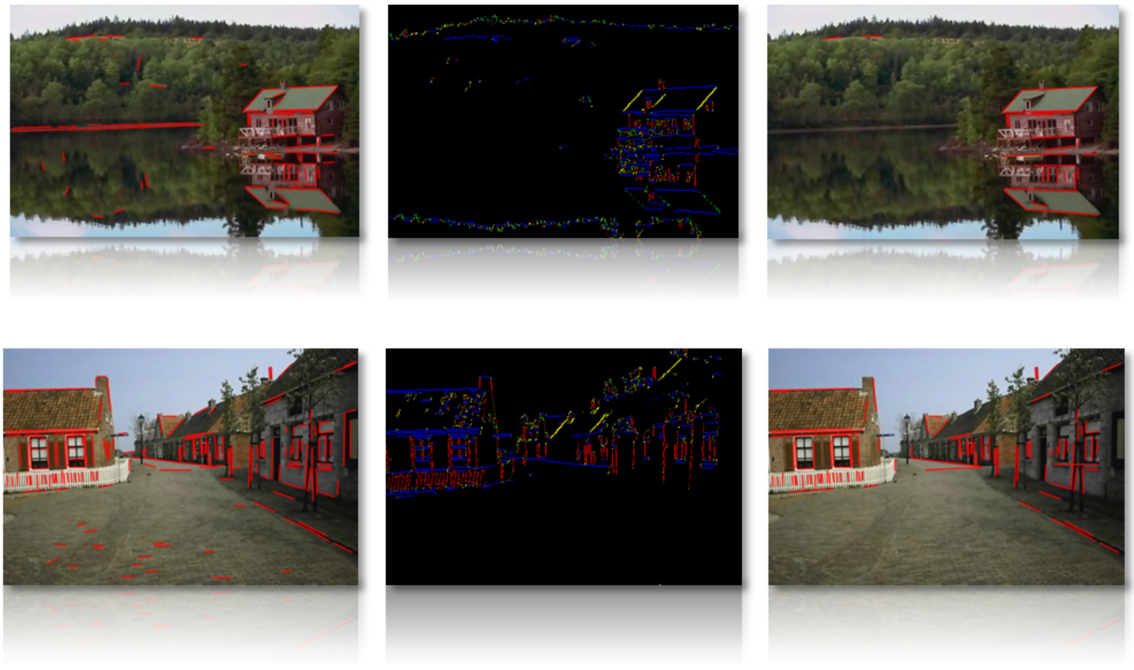


Figure 11.8: Left: Results of the line segment detection quantized and grouped into 8 orientations of 22.5° angular difference between -90° and 90°. Middle: Edge orientations color coded $\approx 0^\circ$ (blue),$\approx 90^\circ \vee -90^\circ$ (red), $\approx 45^\circ$ (yellow) and $\approx -45^\circ$ (green). Right: Lines that do not lie on or near to a gradient of almost similar orientation (middle) are discarded

For each image site s_i and scales $c = \{2, 3, 4, 5\}$, we then compute the number of parallel lines $n_{\|_{0^\circ}}$ and $n_{\|_{90^\circ}}$ for 0° and $-90^\circ/90^\circ$ in w_c. For scales $c \in \{2, 3, 4, 5\}$, we further compute the number $n_{\diagup}$ of orientations that contribute a minimum number of lines in w_c as well as the average angle $\overline{\Delta\varphi_{\diagup}}$ between all such found dominant line orientations:

$$\overline{\Delta\varphi_{\diagup}} = \begin{cases} \| \sin(\frac{1}{n_{\nabla} \times (n_{\nabla} - 1)} \sum_{i,j}^{n_{\nabla} \times n_{\nabla}} \|\varphi_{\diagup_i} - \varphi_{\diagup_j}\|)\|, & \text{if } i \neq j \\ -1, & \text{otherwise} \end{cases}$$

Note that scale $c \in \{1\}$ has been tested and deliberately neglected for these kind of features, as it is to small to provide non ambiguous information.

Corner Point Patterns

Using corner points as a feature is motivated by the observation that corners in and around man made structures often occur on near right angle corners and junctions. Thus, we assume, that a clustering of such corner points in a specific image region might indicate the occurrence of a man made structure in that region and we can simply use the number n_{cp} of such corner points within w_c as a measure for the region to be more likely man made than natural.

Generally, the study of spatial arrangements of points in n dimensional spaces and, specifically, whether they are clustered, randomly or regularly distributed is covered by Point Pattern Analysis (PPA) ([14];[108];[508]). The two major types of approaches in Point Pattern Analysis would be first- and second order analysis methods. Both approaches compare their evaluations against the model of Complete Spatial Randomness (CSR), which is defined as the number of points in the region under study following a Poisson distribution. This implies that the points are distributed uniformly, randomly and independently, i.e. a pattern typically generated by a Spatial Poisson Process. First order analysis compares the average nearest neighbor distance of points of the study region against that of a Poisson pattern, given the number of points and the size of the study region. Generally, a problem with nearest neighbor analysis is the lack of discrimination between scales. E.g., points might be clustered at small scales, whereas these clusters themselves might be dispersed. Second order methods, such as Ripley's K Function incorporate such a behaviour into the measurement. Therefore, the points within increasing distances from one central point are counted and such counts are averaged over all central points. Given a Poisson pattern one gets a function with the number of points increasing as the square of distance. Departures from such a Poisson pattern can be captured by Monte Carlo simulations [390], simulating a vast number of random point patterns to produce confidence envelopes.

We scrutinized the possibilities of both approaches for our specific task. To calculate the nearest neighbor distance of points in first order analysis we employed nearest neighbor search algorithms using the FLANN library [326]. For the computation of Ripley's K Function, we created a Monte Carlo procedure of simulated 2D Spatial Poisson Patterns that were generated based on a method described in [53], called Poisson Disk Sampling (PDS). However, results were not discriminative enough, which is why we propose a task specific corner point feature ourselves.

First, corner points are detected applying the Shi-Tomasi corner detector [273], a corner detection algorithm based on an improved Harris corner detector [192], to the l channel of $\boldsymbol{I}_{bf/HSL}$. Second, for each detected corner point p we select the image site s_i it occurs within to take its corresponding w_c as a reference region and check whether the average gradient orientation difference would be $\overline{\Delta\varphi_{\nabla}} > 0.95$ for at least one $c \in \{1, 2, 3, 4, 5\}$. If so, the corner point is marked as a “right angle corner point”. Finally, for each image site s_i, we compute the number of right angle corner points n_{cp} for scales $c \in \{1, 2, 3, 4, 5\}$. In an additional step, we also added corner points within a very close distance to the right angle corner points along with corner points in a slightly farther distance that lie on horizontal or vertical gradients to n_{cp}. This is due to the assumption, that these points belong to the man made structure as well. Figure 11.9 (right) illustrates some of the results of our approach.

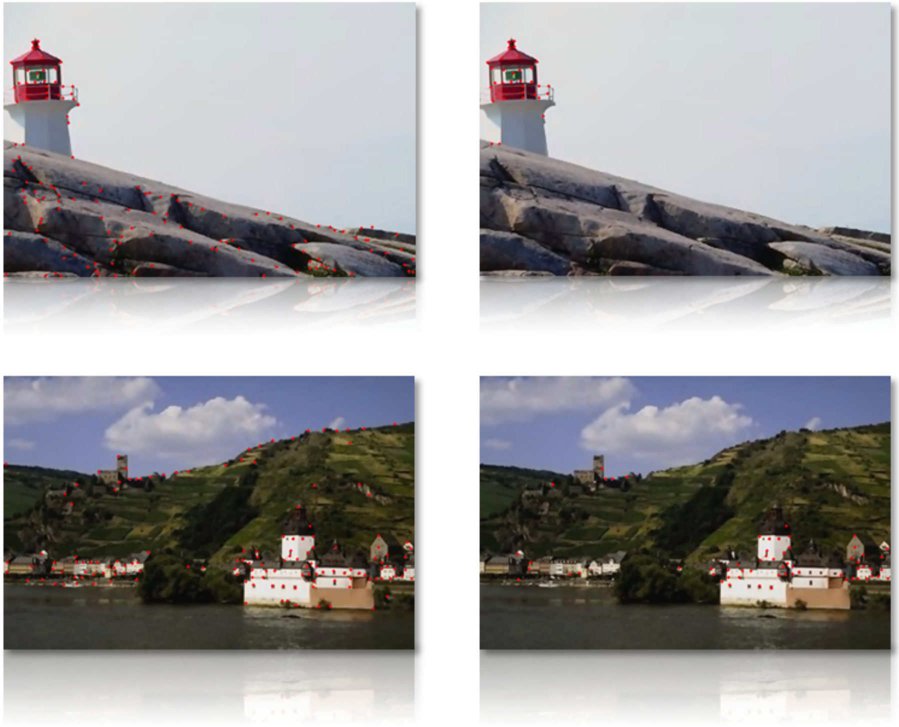


Figure 11.9: Some exemplary results of our algorithm to select only those corner points on near right angle junctions (right) from all corner points detected by the Shi Tomasi corner detector (left)

Additionally, in his technical report on [268], Rees [379] suggest the usage of color features in the feature set. We, however, exclude color as a feature for two reasons. First, the feature set as well as the detection algorithm shall be applicable to gray-scale scenes as well. Second, color itself is not a very discriminative feature within our context and variations in color are already captured in our approaches based on Gabor extraction.

Parameter Learning

In their framework ([265]; [266]), to determine parameters $\boldsymbol{w}^T$ and $\boldsymbol{v}^T$, Kumar and Hebert maximize the "penalized log pseudo-likelihood" ([255];[267]) of (11.1) on a training set of images and given ground truth labeling. They assume a Gaussian prior over the interaction parameters $\boldsymbol{v}$ such that $p(\boldsymbol{v}|\tau) = \mathcal{N}(\boldsymbol{v}, 0\tau^2\boldsymbol{I})$ where $\boldsymbol{I}$ is the identity matrix. Additionally, they assume the prior over parameters $\boldsymbol{w}^T$ to be uniform. Kumar and Hebert ([268]; [269]) suggest $\tau = 0.001$.

In our framework, given a set of M independent training images (and ground truth labels), first, $p_{svm}(x_i|\boldsymbol{y}_i)$ is trained. As mentioned, one great benefit of our proposed Dual Support Vector Field model is that it reduces additional parameter learning to learning only a single scalar parameter v for the interaction potential. Thus, our objective function $l(v)$ to be maximized is given by:

$$\begin{aligned}\hat{v} = \arg\max_{v} \sum_{m=1}^{M} \sum_{i \in V} \Big\{ & \log p_{svm}(x_i|\boldsymbol{y}_i) \\ & + \sum_{j \in \mathcal{N}_i} v\, x_i\, x_j\, (1 - \|p_{svm}(x_i{=}1|\boldsymbol{y}_i) - p_{svm}(x_j{=}1|\boldsymbol{y}_j)\|) - \log z_i \Big\} - \frac{1}{2\tau^2}\, v^2\end{aligned} \tag{11.7}$$

with:

$$z_i = \sum_{x_i \in \{-1,1\}} \exp\Big\{ \log p_{svm}(x_i|\boldsymbol{y}_i) + \sum_{j \in \mathcal{N}_i} v\, x_i\, x_j\, (1 - \|p_{svm}(x_i{=}1|\boldsymbol{y}_i) - p_{svm}(x_j{=}1|\boldsymbol{y}_j)\|) \Big\} \tag{11.8}$$

If τ is given, the penalized log pseudolikelihood in (11.7) is convex with respect to the model parameters and can be maximized using gradient ascent ([432]; [368]; [38]).

Inference

To find an "optimal" label configuration on a new test image, we use **max-flow/min-cut** algorithms, described in appendix A.5, as these can be utilized, for binary classifications and if the probability distribution meets certain conditions [181], to exactly compute the Maximum A Posteriori (MAP) estimate for an undirected graph ([254]; [252]; [251]; [46]). Our tests revealed best results for a specific higher order neighborhoods $\mathcal{N}_i$, i.e. (n = 2).

Results & Discussion

Our model was trained and tested on the image data and label sets provided by Kumar (108 train and 129 test images). The Logistic classifier approach (i.e., $I(x_i, x_j, \boldsymbol{y}) = 0$) of Kumar [266] and our own serves as profound references to evaluate the discriminative power of our proposed enhanced feature set in combination with non-linear SVMs. As illustrated in table 11.2, our enhanced feature set is able to detect up to 11 percent more man made structures while having an almost identical false positive rate than Kumar.
The application of Dual Support Vector Fields maintains the high discriminative power while reducing false detections.

	DR (in %)	**FP** (per img.)
Kumar [266]		
Logistic Classifier	61.79	2.28
Discrim. Random Field	72.54	1.76
our approach		
Logistic Classifier	72.58	2.53
Dual Support Vector Field	72.18	1.74

Table 11.2: Results of our algorithm compared to Kumar [268]. Detection Rates DR are given in percent % and False Positives FP in false detection per image

The computation of features on a new test image, as well as scene labeling takes 7 - 14 seconds on an *Intel i5* 2.53GHz machine, depending on $\mathcal{N}_i$, and is therefore suitable in our application. Figure 11.11 shows some exemplary results of our Dual Support Vector Field approach. Figure 11.10 additionally illustrates some cases where the DSVF outperforms the Logistic Classifier approach.

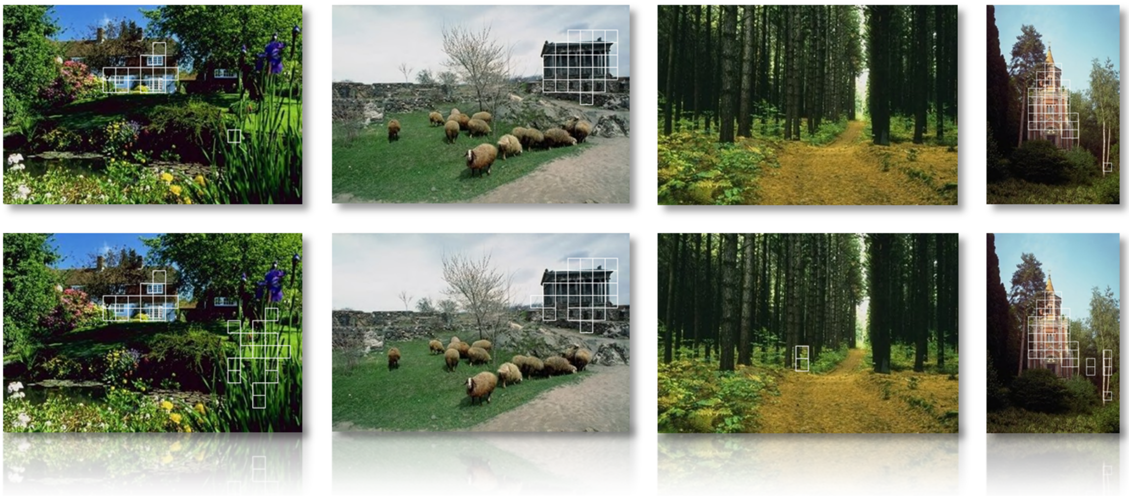


Figure 11.10: Dual Support Vector Fields (top row) outperforming the Logistic Classifier (bottom row). Man made structures highlighted via white squares

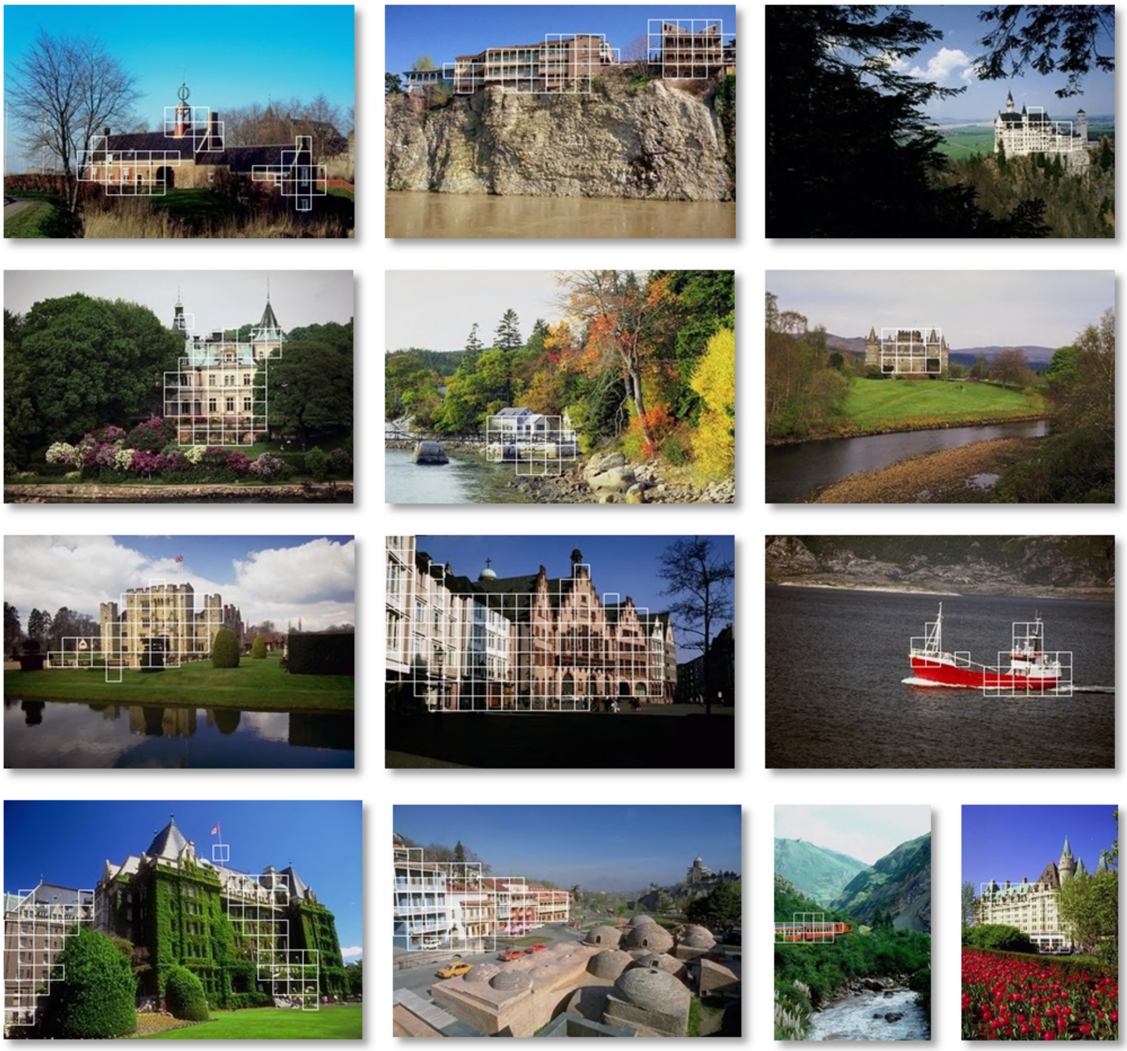


Figure 11.11: Man made structures (highlighted) detected by Dual Support Vector Fields

Visual Uncertainty

Note that there is a general ambiguity concerning smooth regions. Such regions might either belong to some natural plane region, like a clear blue sky or to a man-made structure, e.g. such as a plane wall. To keep classification a 2-class problem, we rank smooth regions among natural structures. We then post-process natural regions for sonification, as will be described in section 11.3. Briefly, we use the an specific measure to estimate the roughness grade of such patches. Within the sonification model, smooth regions will thus not be sonified using man-made or natural sonification method at all, which finally generates a "acoustically three class" sonification, using the user's acoustical discriminative capabilities.

Figure 11.12 illustrates further cases where a strict classification into "man made" or "natural" cannot be afforded distinctly, neither by a machine nor by a sighted human. Should a grass-covered wall be labeled as grass or man-made structure or something in between? However, in these cases, visual information alone seems to be insufficient for the classification task and, thus, blind people that are trained in "auditory scene understanding" might out-perform machine learning based scene labeling systems.

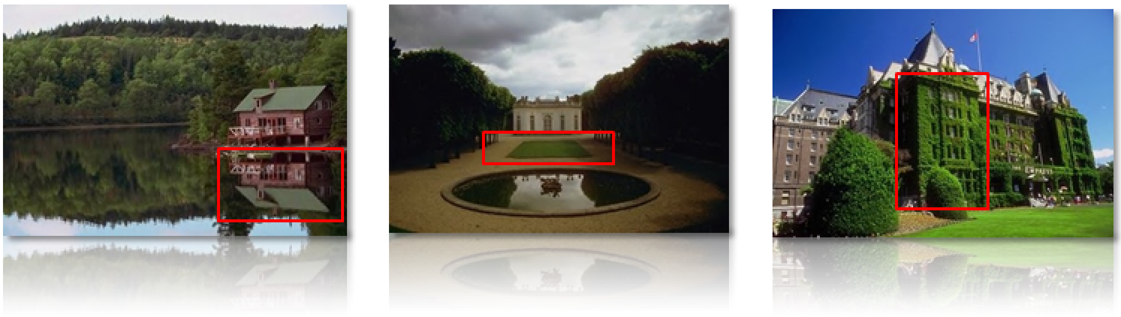


Figure 11.12: Visual Uncertainty: Should the framed areas be considered natural or man made?

11.3 Low-Level Feature Extraction Controlled by High-Level Classification

Natural Regions

Natural Regions (i.e. those that are not classified as man made) are evaluated applying a textural roughness measure called *fractal dimension (FD).* Pentland [351] showed, that the *Fractal Dimension (FD)* [303], [484], [242] of a surface corresponds quite closely to our intuitive notion or roughness, as discussed in appendix A.2, which is why it was preferred to gradient based entropy, presented in 7.3. Hence, on all natural classified regions we compute the fractal dimension using the bilateral filtered image. A well known estimate for the fractal dimension would be the *differential box-counting (DBC)* method [410].
The method represents a gray level $2D$ image $\boldsymbol{I}_{2D}$ as a $2D$ surface within a euclidean $3D$ space with the the gray level denoting a z along with each (x, y) pixel position within the image and estimates the fractal dimension by dividing the number of little boxes, needed to cover the overall $3D$ area, by their diameter. However, due to a series of problems, outlined in [289], the accuracy of the original DBC method is limited. [289] present three main modifications in their box-counting estimation method. Hence we implemented their method as described in appendix A.2.

We compute the FD for each image site i at scale $w_{c_2} = 32 \times 32$ pixels based on the original (not smoothed) input image. The fractal dimension of $2D$ regions in principle would be between 2.0 for a smooth $2D$ surface, and 3.0 for a perfect $3D$ cube. Thus, we map our results to 0 to 1. However, generally, results will rather lie between 2.0 and 2.5. As a result we obtain an estimate for the fractal dimension FD_i of each image site i, which we then assign to the augmented visual pixels $\boldsymbol{v}(x, y)$ of all pixel positions (x, y) of i:

$$v_{FD}(x, y) = FD_i$$

Man Made Structures

One benefit of our novel feature set is that we gain features that are useful for both, man made structure detection and sonification. To sonify important information about man made structures, we, therefore, do not have to compute additional features and can directly choose from the feature set already calculated.
Beside the sonification of man made structure image sites, we select two features to sonify within these found man made structures from the feature set:

- $v_{\varphi_{\nabla_1}}(x, y) = \sin(\varphi_{\nabla_1})$ - the mapped most dominant gradient orientation within each image site at scale $c = 2$.

- $v_{n_{\|}}(x, y) = n_{\|_{0^\circ}} \vee n_{\|_{90^\circ}}$ - the number of parallel lines $n_{\|_{0^\circ}}$ and $n_{\|_{90^\circ}}$ at scale $c = 2$.

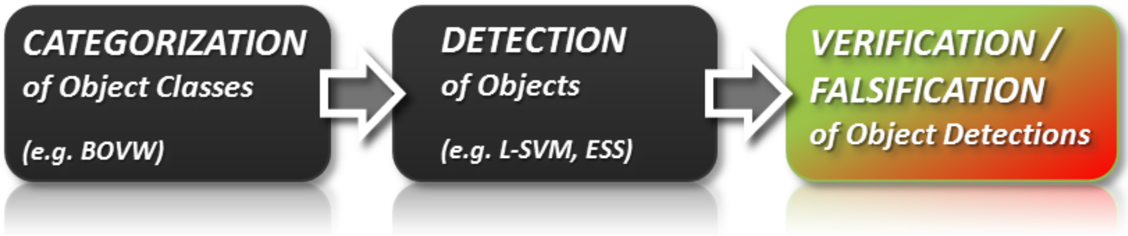


Figure 11.13: Overview over the Ideal Object Recognition & Verification Processing Pipeline

11.4 Object Recognition & Verification

Figure 11.13 gives an overview over the processing pipeline from object detection to classification. It consists of three main stages:

- **Stage I - Categorization**: simply checks whether elements of an object class do occur in the image. It, however, does not detect or localize any object of each class within the image. Hence, it operates considerably faster than the subsequent detection algorithm. Thus, detection is only performed for objects of found object classes.

- **Stage II - Detection**: searches for instances of detected object classes. Each detected instances is described as a surrounding rectangle and a detection confidence value.

- **Stage III - Verification / Falsification**: performs a subsequent classification based on an individually formed feature vector for each object detection, to decide whether such an object detection is a correct one rather than a false one.

Hence, the proposed approach to verify or falsify object detections is applied to the outcome of any object recognition algorithm, which can be treated similar to a "black box". To the best of our knowledge, it is the first of its kind. For object categorization we chose a **Bag of Visual Words** approach [93] and for detection an approach based on **Discriminatively Trained Part Based Models** [142], both trained on 5 classes of the "Visual Object Classes Challenge 2010 (VOC2010)" [135]. The classes, "car", "cat","airplane", "horse", "person" were chosen, as they are rather distinguishable for categorization / detection than, e.g., "cat" & "dog", which allows for a better evaluation of our proposed **Verification / Falsification** algorithm [9].

[9] The proposed Verification / Falsification algorithm has been published in [19]

The aforementioned algorithms for categorization and detection are chosen, as they are current state of the art and both are provided by the OpenCV library ([49]; [273]). In principle, our algorithm can be trained and appended to the output and characteristics of any object categorization algorithms as well as any detection approaches, such as Efficient Subwindow Search (ESS) ([275]; [276]). The application of a Verification / Falsification approach subsequently to previous object detection within the context of image pre-processing for the visually impaired has several crucial benefits:

- No matter how good an object detection algorithm will ever become, it will certainly never reach 100 percent correct detection rate. Table 11.4 shows the average precision scores of current state of the art detection algorithm [142] for each object category of the PASCAL Visual Object Classes Challenge 2010 [135].
- Especially within the context of pre-processing the image for the visually impaired it is rather important that most of the false positive detections will be removed before sonification to avoid confusion on the side of the blind user, who can not check for a correct detection via sight. Hence, our algorithm follows what could be called a "conservative" strategy, which means it should rather neglect a correct detection, than accept a false correction.

For computational complexity reduction we executed object categorization and detection as well as parts of our own algorithm in parallel, using **OpenMP** ([70]; [72]), an API for parallel computing in C/C++. Our testing machine provides 4 Cores, resulting in 4 parallel processes.

plane	**bicycle**	**bird**	**boat**	**bottle**	**bus**	**car**	**cat**	**chair**	**cow**
49.2	53.8	13.1	15.3	35.5	53.4	49.7	27.0	17.2	28.8
table	**dog**	**horse**	**mbike**	**person**	**plant**	**sheep**	**sofa**	**train**	**tv**
14.7	17.8	46.4	51.2	47.7	10.8	34.2	20.7	43.8	38.3

Table 11.3: Average precision scores obtained in [142] for each object category of the PASCAL Visual Object Challenge 2010 [135]

Object Categorization Using a Bag of Visual Words Model

One basic approach to object categorization in images is to treat those as a collection of regions, describing only appearance and the occurrence of salient features and ignoring any spatial structure. This approach, known as **Bag of Words (BoW)** model, originated in **Natural Language Processing (NLP)**. The simplifying assumption is made that the order of the words in a sentence or text document is of negligible importance for classifying the general category of the document.
To classify a document using a **BoW** model, as a first step, a dictionary containing a large number of words relevant to the application domain has to be created. Relevant words are then words which have a high probability of being contained in one class of texts and a low probability of being contained in others. Once the dictionary has been built it is possible to describe a document in terms of word counts. Thus, a vector describing the document has a length equal to the number of words in the dictionary. Each dimension represents the number of occurrences of a certain word within the document. Using such a representation, methods such as **probabalistic Latent Semantic Analysis (pLSA)** [210] and **Latent Dirichlet Allocation (LDA)** [36] can be harnessed to filter coherent topics in collections of documents in an unsupervised way.

The rationale behind this approach can be applied in Computer Vision ([141]; [435]; [93]). The assumption is that the spatial relationship of so called **visual words** in an image is of negligible importance. Those visual words can be numeric descriptions of certain salient areas of the image. These areas might be corners or junctions, or others which are most likely invariant of scale, rotation or illumination. These highly structured regions are often called “key points” and their numerical descriptions “features” or, as already mentioned, visual words. There is a variety of image feature detectors, such as **Scale-Invariant Feature Transform (SIFT)** [296] or **Speeded Up Robust Features (SURF)** ([478]; [477];[25]). In our implementation we employ SURF, as it could be proven to show significant performance over all other mentioned detectors in accuracy and speed [25].

However, in Computer Vision it would not make sense to construct a dictionary from all the features obtained from a training set directly (as done in NLP), as it would be of overwhelming size. An intermediary step has to be to find a limited number of feature vectors which represent the feature space well for constructing a dictionary. This is often done by **k-means** clustering [326], an iterative algorithm for finding clusters in data. After the dictionary has been constructed new images can be described by extracting features from them and matching them with the features in the dictionary which are closest, based on a trained classifier, such as Support Vector Machines ([86]; [61]), discussed in appendix A.7. The implementation provided by OpenCV library is based on [93].

Object Detection with Discriminatively Trained Part Based Models

The approach employed for object detection / localization and provided by the OpenCV library has been initially proposed by P.F. Felzenszwalb in [142]. It is based on a **histogram of oriented gradients (HOG)** detector by Dalal-Triggs [98] that uses filters consisting of HOG features to represent object categories. A **sliding window** approach is employed where filters are applied at all positions and different scales of an image. The Dalal-Triggs model is enhanced, using a **star-structured part-based model** defined by, what Felzenszwalb [142] calls, a “root” filter and an additional set of parts filters and associated deformation models.

The model is represented as a part based deformable model, known as star model, and the score of one of such models at a given position and scale would be the score of the root filter and the sum over parts of the maximum, over placements of that part, of the part filter score on its location subtracting a deformation cost which evaluates the deviation of the part from its ideal position relative to the root filter. The dot product between a filter, interpreted as a set of weights, and a sub-window of a feature pyramid is employed to score both, root and part filters. Further, Felzenszwalb [142] propose a representation of the class of models by a mixture of star models, where a score of a mixture model at a particular position and scale is would be the maximum over components, of the score of that component model at a given location.

We employ “Release Version 3” [169] of Support Vector Machine models, provided by Felzenszwalb [142]. These models are already trained on the 20 classes of Visual Object Classes Challenge [135], including the 5 specific classes we used for testing our algorithm.

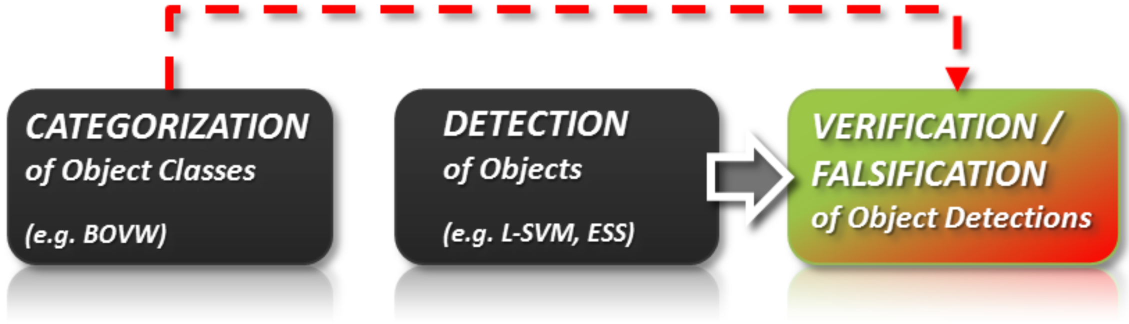


Figure 11.14: Overview over the modified "object recognition & verification processing pipeline". Results from the categorization algorithm are additionally fed into the Verification / Falsification module

A Learning-based Approach to Verification / Falsification of Object Recognitions

As mentioned, a Verification / Falsification stage subsequently to previous object detection, in the context of image evaluation for the visually impaired, is important to avoid confusing a blind user with incorrect object detections. Our approach, therefore, is not to improve recognition levels of the employed categorization or detection algorithms but rather separate correct from incorrect object localizations.

Furthermore, the "ideal" cascade of the categorization to detection pipeline, as shown in figure 11.13, has to be slightly dissolved. According to this ideal cascade, only those object entities would be searched for by an object localization algorithm, whose existence within the scene has been confirmed by the categorization algorithm. However, our experiments based on ground truth data revealed that the categorization algorithm sometimes could not find a specific object class which, however, was represented within the image and whose instances could be localized by the detection algorithm. Thus, we performed both algorithms individually on the all of the 5 classes and processed their outcomes within a verification / falsification stage (as illustrated in 11.14). To reduce computation time, it is, however, generally recommended to follow the strategy of an "ideal" processing pipeline (as in figure 11.13) and only have the detection algorithm localize objects which have been recognized by the categorization algorithm, accepting the miss of some object instances.
Finally, we present a learning-based approach to object detection verification / falsification, which includes:

- A "conservative" strategy of rather neglecting a true detection than accepting a false one, which would create confusion.
- Building an additional feature set that uses relative information between all found objects within an image besides categorization and detection probabilities to corroborate this "conservative" strategy, i.e., correct falsification.

- Allowing uncertainty. Some objects tend to be strongly classified in several categories. For instance, an upright sitting cat is eventually classified as “cat” as well as “person”. Our Verification / Falsification algorithm allows uncertainty as several similarities exist indeed. The task will then be up to the user to explore and categorize the object with additional acoustical low-level features.

Feature Set

The only absolute information we get for each object detection o_i are its categorization and detection probabilities, which are not sufficient to base a Verification / Falsification classification upon. Although the detection probability for a specific object detection is a strong indicator for its validity, we can gather more information and build additional features based on observed characteristic correlations in the results of the categorization and object detection algorithm we employ.

Figure 11.16 represents all object detections that have been localized by the detection algorithm. As one can observe, multiple entities of four object classes have been detected and only two of them would be considered a correct detection. We now list the observations $\boldsymbol{O}$ that we made when using the detection algorithm:

- $\boldsymbol{O_I}$: A rather large object detection can be an indication for a correct detection.
- $\boldsymbol{O_{II}}$: The higher the number of object detections of a specific object class and the smaller their size within the image, the more unlikely their validity.
- $\boldsymbol{O_{III}}$: Multiple small and clustered object detections often tend to be incorrect.
- $\boldsymbol{O_{IV}}$: Specific object detections that have been detected multiple times and significantly overlap tend to be correct detections.
- $\boldsymbol{O_V}$: Even if the probability of a specific object detection o_i is rather small, if it is big in comparison to other object detections of the same object class, o_i is more likely to be a true detection. The same observation is made for object detections compared across object classes.
- $\boldsymbol{O_{VI}}$: Even if the probability of a specific object class categorization c_{o_i} is rather small, if it is big in comparison to other object class categorizations, objects of c_{o_i} are more likely to be present in the image.
- $\boldsymbol{O_{VII}}$: While detecting a correct object, the detection algorithm tends to find multiple smaller incorrect object detections of other classes.
- $\boldsymbol{O_{VIII}}$: A correct object detections tends to be rather big compared to the multiple smaller incorrect object detections of other classes. These often group within the region of the correctly localized object.

We now harness these observations to build additional "relative" features and create a 16 dimensional feature vector $\boldsymbol{fv}_{o_i}$ for each object detection. Note that, when using a different detection algorithm, one has to check whether this observations still apply.

Further note that, before extracting features for each object detection o_i, we perform a prior algorithm that checks for major overlaps of detections within each object class. Major overlaps are, for instance, two detections that overlap by at least 70 percent. Those detections will then be rather assumed to be a single detection. Thus, a single detection is built or "fused" from the former two, forming a single great bounding box out of the smaller ones. The detection probability value of the new single detection is the greater one of the former detections.

Generally, all information within $\boldsymbol{fv}_{o_i}$ can be divided in two major groups. First, all elements computed based on the information of all detected objects within the object class c_{o_i} of o_i, called (**Intra Object Class Information**). Second, all features computed based on the information of all detected objects across all classes called (**Inter Object Class Information**):

Intra Object Class Features

- $v_{categ.}(c_{o_i})$ - categorization probability value for class c_{o_i} (mapped to $\{-1, 1\}$). The higher $v_{categ.}(c_{o_i})$, the more likely objects of c_{o_i} to occur in the image.
- $r(i, c_{o_i})$ - ratio of the area of o_i (of object class c_i) divided by the area of the image. If $r(o_i, c_{o_i}) \approx 0$, o_i covers almost no part of the image. If $r(o_i, c_{o_i}) \approx 1$, o_i covers almost the whole image. (Based on $\boldsymbol{O_I}$).
- $\sum(r(o_i, c_{o_i}))$ - sum of all $r(o_i, c_{o_i})$ of all o_i of c_{o_i}. (Based on $\boldsymbol{O_I}$ and $\boldsymbol{O_{II}}$).
- $n_{nb}(o_i)$ - number of all neighbored object detections of o_i of c_{o_i}. If high, all detections of c_{o_i} become unlikely. (Based on $\boldsymbol{O_{II}}$).
- $\mu_d(c_{o_i})$ - mean distance and "cluster index" of all o_i of c_{o_i}. Multiple small and clustered o_i often tend to be incorrect each. (Based on $\boldsymbol{O_{III}}$).
- $n_{fusions}(o_i)$ - number of object detections that overlapped by more than 70 % in c_{o_i} and have been "fused" to create o_i. If greater zero, o_i is often a true detection. (Based on $\boldsymbol{O_{IV}}$).
- $\mu_d(o_i, c_{o_i})$ - mean distance from o_i to all neighbored object detections within c_{o_i}. The greater $\mu_d(o_i, c_{o_i})$, the more likely o_i not to belong to a certain cluster. (Based on $\boldsymbol{O_{III}}$).
- $v_{det.}(o_i, c_{o_i})$ - probability estimate of the detection algorithm for o_i of class c_{o_i} (mapped to $\{-1, 1\}$). The higher $v_{det.}(o_i, c_{o_i})$, the more likely o_i.

- $\mu_{\uparrow}(v_{det.}(o_i, c_{o_i})) = \frac{1}{n}\sum_j d_{\uparrow}(v_{det.}(o_i, c_{o_i}), v_{det.}(o_j, c_{o_i}))$,
  with $d_{\uparrow}(v_{det.}(o_i, c_{o_i}), v_{det.}(o_j, c_{o_i}))$
  $$= \begin{cases} d = \|v_{det.}(o_i, c_{o_i}) - v_{det.}(o_j, c_{o_i})\|, & \text{if } d > 0 \\ 0, & \text{otherwise} \end{cases}$$
  with n denoting number of objects of class c_{o_i}. The higher $\mu_{\uparrow}(v_{det.}(o_i, c_{o_i}))$, the more likely o_i, although $v_{det.}(o_i, c_{o_i})$ might be small. (Based on $\boldsymbol{O_V}$).

- $\mu_{\downarrow}(v_{det.}(o_i, c_{o_i})) = \frac{1}{n}\sum_j d_{\downarrow}(v_{det.}(o_i, c_{o_i}), v_{det.}(o_j, c_{o_i}))$,
  with $d_{\downarrow}(v_{det.}(o_i, c_{o_i}), v_{det.}(o_j, c_{o_i}))$
  $$= \begin{cases} d = \|v_{det.}(o_i, c_{o_i}) - v_{det.}(o_j, c_{o_i})\|, & \text{if } d < 0 \\ 0, & \text{otherwise} \end{cases}$$
  with n denoting number of objects of class c_{o_i}. The higher $\mu_{\downarrow}(v_{det.}(o_i, c_{o_i}))$, the more unlikely o_i, especially if $v_{det.}(o_i, c_{o_i})$ is already small. (Based on $\boldsymbol{O_V}$).

Inter Object Class Features

- $\mu_{\uparrow}(v_{categ.}(c_{o_i})) = \frac{1}{n}\sum_j d_{\uparrow}(v_{categ.}(c_{o_{o_i}}), v_{categ.}(c_j))$,
  with $d_{\uparrow}(v_{categ.}(c_{o_i}), v_{categ.}(c_j))$
  $$= \begin{cases} d = \|v_{categ.}(c_{o_i}) - v_{categ.}(c_j)\|, & \text{if } d > 0 \\ 0, & \text{otherwise} \end{cases}$$
  n denotes number of all found object classes c_j and $d_{\uparrow}(v_{categ.}(c_{o_{o_i}}), v_{categ.}(c_j))$ is computed $\forall c_j \neq c_{o_i}$. The higher $\mu_{\uparrow}(v_{categ.,c_{o_i}})$, the more likely c_j, although $v_{categ.,c_{o_i}}$ might be small. (Based on $\boldsymbol{O_V}$).

- $\mu_{\downarrow}(v_{categ.}(c_{o_i})) = \frac{1}{n}\sum_j d_{\downarrow}(v_{categ.}(c_{o_i}), v_{categ.}(c_j))$,
  with $d_{\downarrow}(v_{categ.}(c_{o_i}), v_{categ.}(c_j))$
  $$= \begin{cases} d = \|v_{categ.}(c_{o_i}) - v_{categ.}(c_j)\|, & \text{if } d < 0 \\ 0, & \text{otherwise} \end{cases}$$
  n denotes the number of all found object classes c_j and $d_{\uparrow}(v_{categ.}(c_{o_{o_i}}), v_{categ.}(c_j))$ is computed $\forall c_j \neq c_{o_i}$. The higher $\mu_{\downarrow}(v_{categ.}(c_{o_i}))$, the more unlikely c_j, especially if $v_{categ.}(c_{o_i})$ is already small. (Based on $\boldsymbol{O_V}$).

- $\sum_{\uparrow}(o_i, c_{o_j})$ - measure for the number of o_j of different classes ($c_{o_i} \neq c_{o_j}$) that do overlap with o_i of c_{o_i} by more than 70 % of their sizes. The higher, the more likely object o_i to contain smaller objects o_j. (Indication for o_i being a correct detection, as the detection algorithm, while detecting a correct object o_i of class c_{o_i}, tends to find multiple smaller incorrect object detections of other classes c_{o_j} within the region of o_i. (Based on $\boldsymbol{O_{VII}}$ and $\boldsymbol{O_{VIII}}$).

- $\sum_{\downarrow}(o_i, c_{o_j})$ - measure for the number of object detections o_j of different classes ($c_{o_i} \neq c_{o_j}$) that do overlap with o_i of c_{o_j} by more than 70 % the size of o_i. The higher, the more likely that o_i lying in another bigger object o_j. (Indication for o_i being incorrect, as the detection algorithm, while detecting a correct object o_j of class c_{o_j}, tends to find multiple smaller incorrect object detections within the region of o_j). (Based on $\boldsymbol{O_{VII}}$ and $\boldsymbol{O_{VIII}}$).

- $\mu_{\uparrow}(v_{det.}(o_i, c_{o_j})) = \frac{1}{n}\sum_j d_{\uparrow}(v_{det.}(o_i, c_{o_i}), v_{det.}(o_j, c_{o_j}))$,
  with $d_{\uparrow}(v_{det.}(o_i, c_{o_i}), v_{det.}(o_j, c_{o_j}))$

  $$= \begin{cases} d = \|v_{det.}(o_i, c_{o_i}) - v_{det.}(o_j, c_{o_j})\|, & \text{if } d > 0 \\ 0, & \text{otherwise} \end{cases}$$

  n denotes number of all object detections o_j in all classes c_{o_j} and $d_{\uparrow}(v_{det.}(o_i, c_{o_i}), v_{det.}(o_j, c_{o_j}))$ is computed $\forall c_{o_j} \neq c_{o_i}$. The higher $\mu_{\uparrow}(v_{det.}(o_i, c_{o_j}))$, the more likely o_i, although $v_{det.}(o_i, c_{o_i})$ might be small.(Based on $\boldsymbol{O_V}$).

- $\mu_{\downarrow}(v_{det.}(o_i, c_{o_j})) = \frac{1}{n}\sum_j d_{\downarrow}(v_{det.}(o_i, c_{o_i}), v_{det.}(o_j, c_{o_j}))$,
  with $d_{\downarrow}(v_{det.}(o_i, c_{o_i}), v_{det.}(o_j, c_{o_j}))$

  $$= \begin{cases} d = \|v_{det.}(o_i, c_{o_i}) - v_{det.}(o_j, c_{o_j})\|, & \text{if } d < 0 \\ 0, & \text{otherwise} \end{cases}$$

  n denotes number of all object detections o_j in all classes c_{o_j} and $d_{\downarrow}(v_{det.}(o_i, c_{o_i}), v_{det.}(o_j, c_{o_j}))$ is computed $\forall c_{o_j} \neq c_{o_i}$. The higher $\mu_{\downarrow}(v_{det.}(o_i, c_{o_j}))$, the more unlikely o_i, especially if $v_{det.}(o_i, c_{o_i})$ is already small. (Based on $\boldsymbol{O_V}$).

To compute $\mu_d(c_{o_i})$ and $\mu_d(o_i, c_{o_i})$, the objects o_i of each object class c_{o_i} are represented as a (fully connected) graph $G = \{V, E\}$, as shown in figure 11.15. Each object o_i marks a node V and the connection between two objects and edge E. The distance between two objects is considered the edge weight. We can now compute the mean distance of all objects in c_{o_i} by computing the **Minimal Spanning Tree (MST)** via **Prim's algorithm** ([369]; [422]; [85]), described in appendix A.5. $\mu_d(o_i, c_{o_i})$ then is the mean of the sum of edge weights in the MST, as illustrated in figure 11.16. Additionally, we can compute the distance from each object o_i to all neighbors within c_{o_i} performing **Dijkstra's Shortest Path algorithm** ([315]; [422]; [85]) on the original (fully connected) graph G, also described in appendix A.5. The mean distance $\mu_d(o_i, c_{o_i})$ is the sum of each path (i.e. corresponding edge weights) divided by the number of neighbors. Note that the distance computation is performed between the closest points of the bounding rectangles of two objects instead of their centers. We therefore compute the intersection of the direct connection of both centers with the bounding rects using a **line-line intersection** approach [139]. This is crucial to allow marking overlapping objects with a zero edge weight. To enhance speed, all graph computation are performed in parallel, using OpenMP.

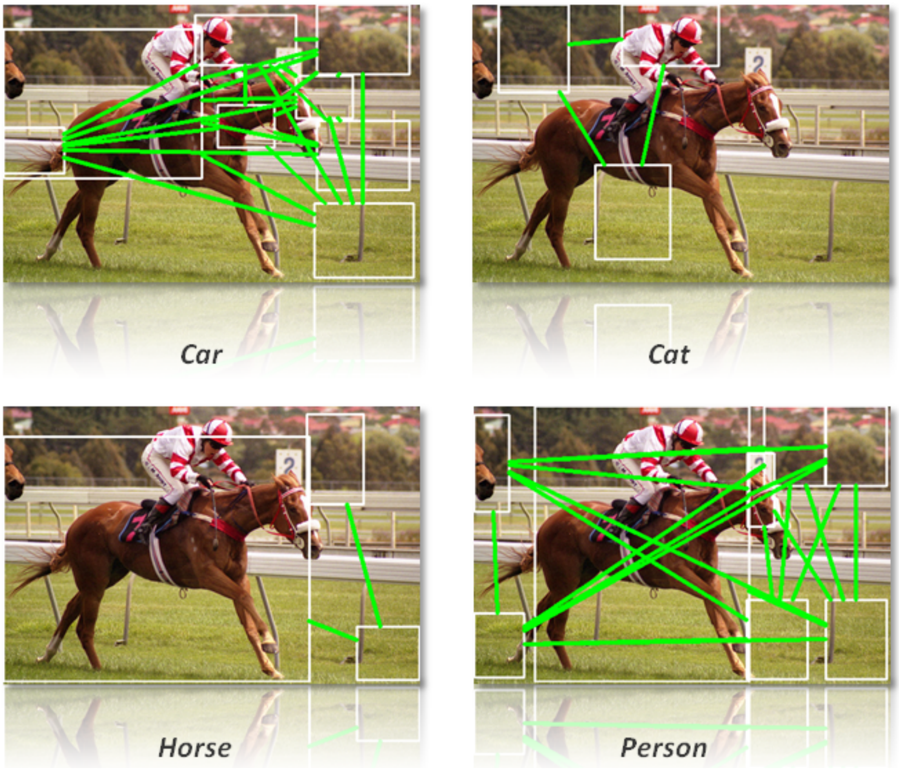


Figure 11.15: Fully connected graph representations of object detections on image 16 of the test set (see figure 11.19) for classes "car", "cat","horse" and "person". Note that no objects of the "airplane" class have been detected. Min. distances marked in green

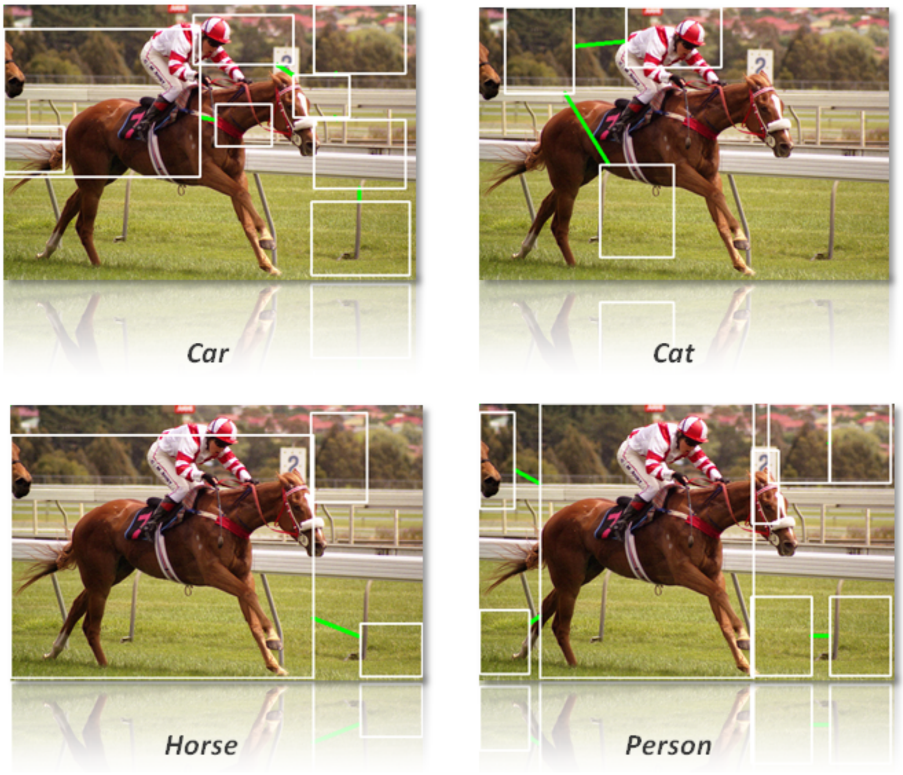


Figure 11.16: MST representations on image 16 of the test set for classes “car”, “cat”,“horse” and “person”

Principal Component Based Feature Set Transform

Due to the rather linear separable and correlated nature of the feature set, before training a classifier, we propose to perform a transformation of the feature set, using Principal Component Analysis (PCA) ([228]; [30]; [349]; [**?**]; [3]; [122]) , which projects each feature vector $\boldsymbol{fv}_i$ with $l = 16$ dimensions, onto a corresponding vector $\boldsymbol{fv'}_i$ in an orthogonal (sub)space:

$$\boldsymbol{fv'}_i = \boldsymbol{P} * (\boldsymbol{fv}_i - \bar{\boldsymbol{fv}}) \tag{11.9}$$

where $\bar{\boldsymbol{fv}}$ denotes the mean feature vector of a training set of feature vectors $\boldsymbol{fv}_i$, $\boldsymbol{P}$ the projection matrix, and $\boldsymbol{fv'}_i$ is the vector in the orthogonal subspace corresponding to each $\boldsymbol{fv}_i$.

Such an approach has several benefits:

- *PCA* transforms a set of observations of variables, which might exhibit correlations, into a set of linearly uncorrelated variables that are referred to as *principal components.*
- As the number of principal components needed to describe discriminances within the original data set is often less than the number of original variables, we additionally yield a reduction of the number of features needed for classification.
- The transformation is defined in a way that the first principal component refers to the direction of the largest variability within the data. Each subsequent principal component in turn accounts for the highest variance possible under the constraint that it is orthogonal, and therefore uncorrelated, to the preceding components. Such a projection of the feature set on a orthogonal subspace alleviates later classification, making linear classification approach suitable.

Generally the combination of data pre-processing based on PCA and subsequent classification using SVM has been studied ([284]; [75]) and employed in various fields, such as financial ([519]; [143]), medical ([405]; [516]; [395]; [521]; [405]) or biometric applications [278].

Hence, for a set of 30 images (shown in figure 11.18), we yielded a total number of 523 object detections and ,therefore, 523 feature vectors $\boldsymbol{fv}_i$ that were used to compute the projection matrix P.

To compute the projection matrix $\boldsymbol{P}$, we first compute the mean feature vector $\bar{\boldsymbol{fv}}$ over all $n = 523$ $\boldsymbol{fv}_i$:

$$\bar{\boldsymbol{fv}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{fv}_i$$

We then subtract the $\bar{\boldsymbol{fv}}$ from each feature vector $\boldsymbol{fv}_i$:

$$\boldsymbol{x}_i = \boldsymbol{fv}_i - \bar{\boldsymbol{fv}}$$

and define a matrix $\boldsymbol{X}$:

$$\boldsymbol{X} = (\boldsymbol{x}_0, \boldsymbol{x}_1, ..., \boldsymbol{x}_n) \in \mathbb{R}^{l \times n}$$

From that, we can set up the *Covariance Matrix* $\boldsymbol{C}$:

$$\boldsymbol{C} = \frac{1}{n}\boldsymbol{X}\ \boldsymbol{X}^T \in \mathbb{R}^{l \times l}$$

The *PCA* is based on the diagonalization of the covariance matrix $\boldsymbol{C}$:

$$\boldsymbol{C} = \boldsymbol{U}\ diag(\sigma_i^2)\ \boldsymbol{U}^T$$

$\boldsymbol{C}$ is symmetrical, hence the columns $\boldsymbol{u}_i$ of $\boldsymbol{U}$ form an orthogonal set of *eigenvectors*. σ_i are the standard deviations within the data along these *eigenvectors*. The diagonalization can be computed using *Singular Value Decomposition (SVD)* [368],[446].

Finally, $\boldsymbol{P}$ can be built, where each row p_j of $\boldsymbol{P}$ is a column $\boldsymbol{u}_i$ of $\boldsymbol{U}$, starting with that i corresponding to the highest σ_i.

The classification results (see table 11.4) show that the main of the feature set can be described using a reduced set of principal components of the transformed feature set. Our experiments were performed using 13 principal components, yielding $k = 15$ elements in $\boldsymbol{fv'}_i$, as the influence of further principal components were rather negligible concerning their σ_i. Hence the projection matrix $\boldsymbol{P}$ is of the dimensions:

$$\boldsymbol{P} \in \mathbb{R}^{k \times l}$$

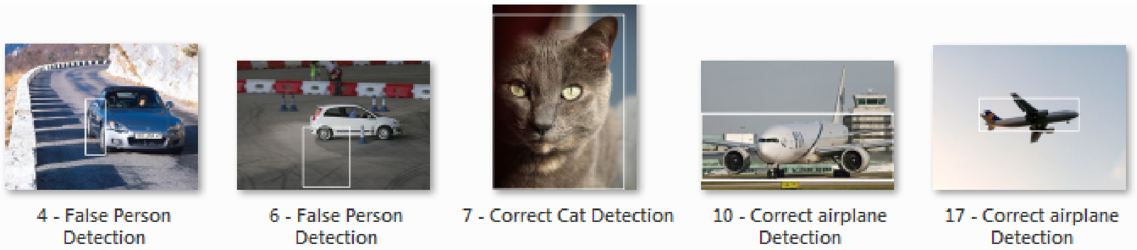
Results & Discussion

We choose classification based on *Support Vector Machines (SVM)*, discussed in appendix A.7, based on *libSVM* [71] and trained on a representative set of only of 85 projected feature vectors $\boldsymbol{fv'}_i$ of the training data set used for computing PCA.
The classifier is tested on an image set of 30 images (shown in figure 11.19), yielding a total number of 560 object detections. For tests, each detection is labeled manually as either 1 or -1, being a correct or incorrect detection. We then compared our algorithm with two simpler classification approaches. First, a basic thresholding approach BT_v, that classifies all detections with $v_{categ.,c_i} > 0$ and $v_{det.,i} > 0$ as a correct detection and as an incorrect detection otherwise. Second, a SVM based classifier SVM_v trained on $v_{categ.,c_i}$ and $v_{det.,i}$. SVM_v only. We trained a linear and non-linear classifier, both yielding equivalent results.

The results of our experiments in table 11.4 indicate our proposed algorithm to be very appropriate to be used within our application. Our algorithm is not only able to correctly falsify 100 percent of incorrect detection, it also outperformed the two other algorithms it was compared with when it comes to correct verification and incorrect falsification rate. Figure 11.17 illustrates some examples where our algorithm outperforms the comparison algorithms. It performs for each image in the test set in ≈ 3 **seconds** on an *Intel i5* 2.53GHz machine, in comparison to ≈ 6 seconds for categorization (of the 5 object classes) based on [93] and ≈ 30 seconds for detection based on [142]. Hence, it can be considered to be used in interactive applications.

Verification / Falsification Algorithm	**Correct Verifications**	**Correct Falsifications**	**Incorrect Verifications**	**Incorrect Falsifications**
Ground Truth	27	533	-	-
BT_v	13 (48.1%)	529 (99.2%)	4 (0.08%)	14 (51.9%)
SVM_v	18 (66.7%)	529 (99.2%)	4 (0.08%)	9 (33.3%)
our approach	**21(77.8%)**	**533(100%)**	**0 (0%)**	**6 (22.2%)**

Table 11.4: Evaluation of our proposed Verification / Falsification approach in comparison with a basic thresholding approach and a SVM trained on $v_{categ.,c_i}$ and $v_{det.,i}$



4 - False Person Detection

6 - False Person Detection

7 - Correct Cat Detection

10 - Correct airplane Detection

17 - Correct airplane Detection

Figure 11.17: Examples of our approach improving over the basic SVM approach. Plane detections in images 10 ($v_{categ.}(c_i) = -0.057$, $v_{det.}(i, c_i) = -0.027$) and 17 ($v_{categ.}(c_i) = 0.9438$, $v_{det.}(i, c_i) = -0.195$) as well as the cat detection in image 7 ($v_{categ.}(c_i) = 0.1756$, $v_{det.}(i, c_i) = -0.036$) are correctly verified by our approach opposed to the SVM approach. On the other hand as opposed to the SVM approach, false person detections in images 4 ($v_{categ.}(c_i) = 0.6005$, $v_{det.}(i, c_i) = 0.5979$) and 6 ($v_{categ.}(c_i) = 1.2143$, $v_{det.}(i, c_i) = -0.066$) are correctly falsified with our approach

Finally, correct detection are stored within the system as *audible object* $\boldsymbol{AO}_{plane,0}$ and each augmented visual pixel $\boldsymbol{v(x, y)}$ is equipped with a queue structure $v_{obj.-queue}$ that stores all the object class c_i of all object detections at (x, y) in decreasing order of $v_{det.}(i, c_i)$. Further, object detections in the image are stored within the system by their bounding rects $rect_i$, c_i and $v_{det.}(i, c_i)$ to be further processed for sonification.

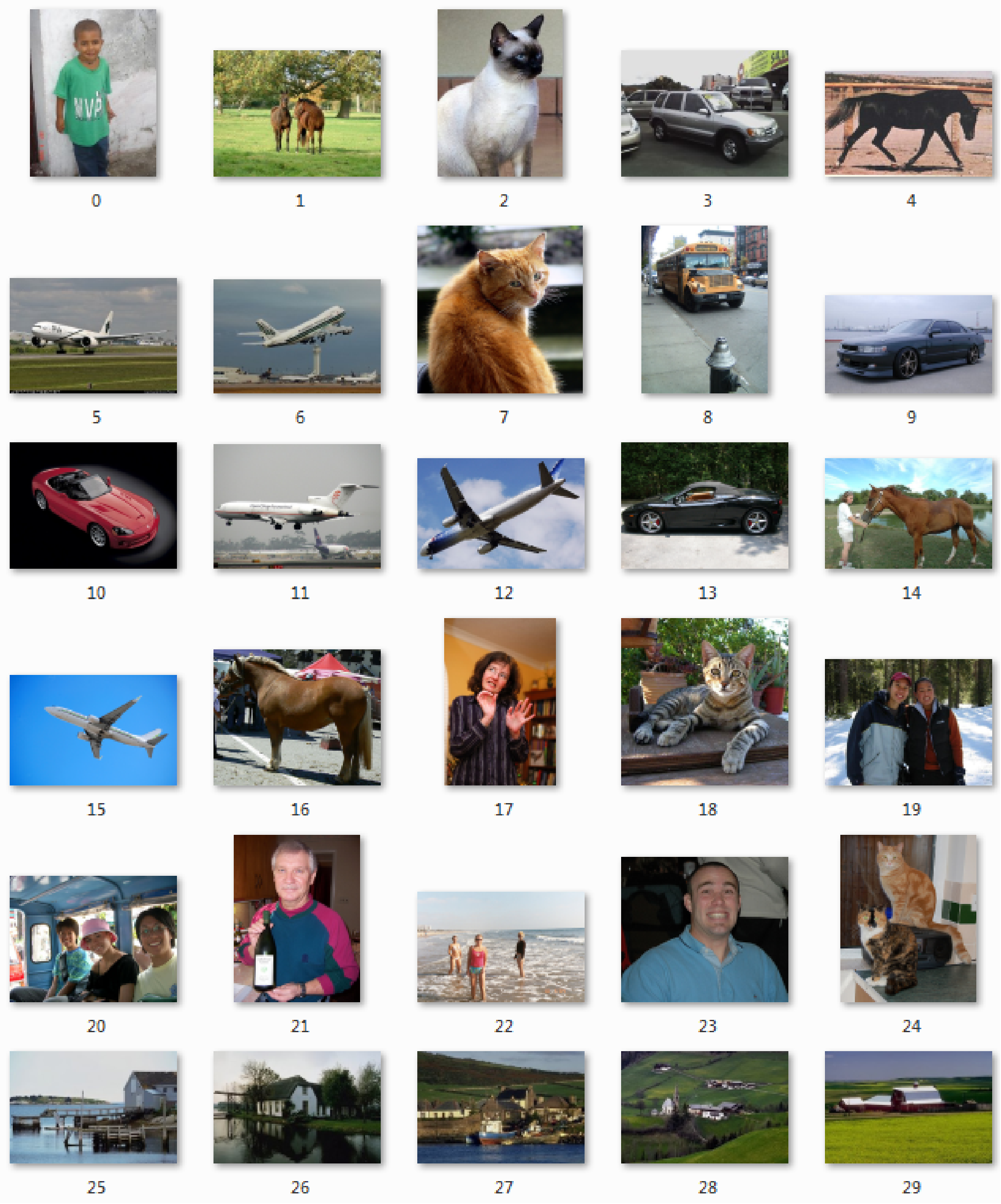


Figure 11.18: 30 images to perform PCA and compute the projection matrix U^T, taken from the Visual Object Classes Challenge 2010 (VOC2010) [135]

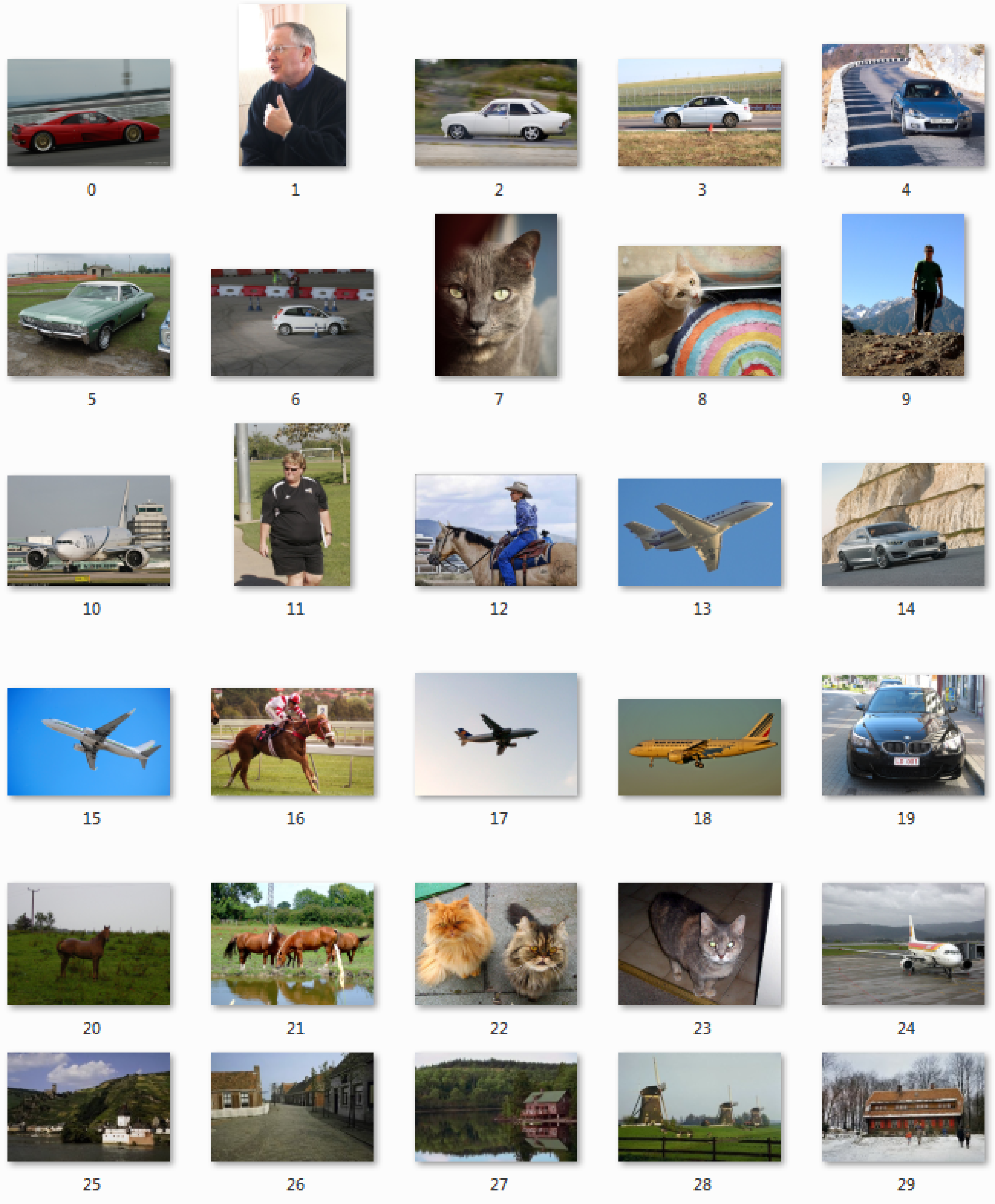


Figure 11.19: 30 images to test SVM classification on, taken from the *Visual Object Classes Challenge 2010 (VOC2010)* [135]

Extracting Objects

Foreground extraction algorithms, as already introduced and used in section 7.4, can be harnessed to enhance the exploration experience with object detections even more elegantly than in section 7.4. In contrast to the method used in section 7.4, we do not have to employ a region growing algorithm to gather enough pixel information for foreground extraction. Instead we can directly utilize the bounding rects $rect_i$ of the object detection i at (x, y). In case of multiple object detection at (x, y), for foreground extraction, the object detection i that exhibits the highest $v_{det.}(i, c_i)$ is selected. Again, the segmentation procedure is initialized by the user hitting an external additional "buzzer" button. Finally, all augmented visual pixels $\boldsymbol{v(x, y)}$ that do not belong to the segmented area have a specific flag $v_s(x, y)$ set to 0 (and 1 otherwise), which will signal the sonification module to not play any sound in these regions at all. As in section 7.4, the specific implementation of the modular sonification, presented in chapter 10, model is slightly modified, as illustrated in figure 11.20.

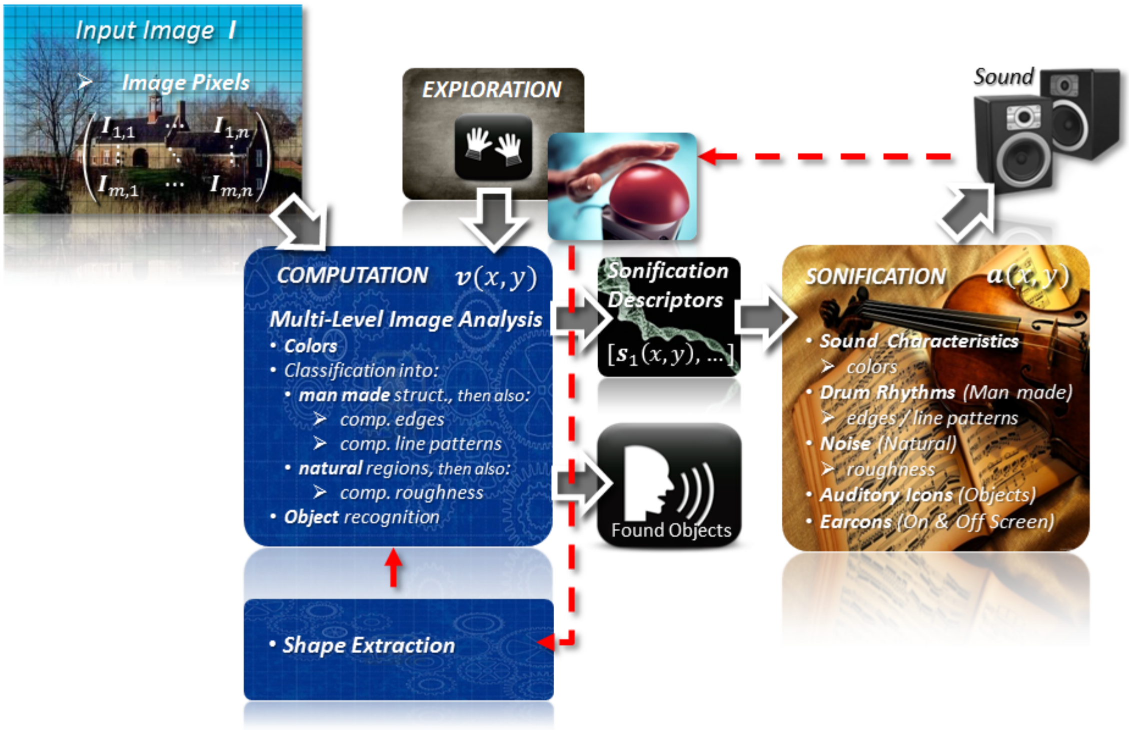


Figure 11.20: Illustration of the slightly modified version of modular sonification model implementation given in chapter 10. If the user hits the red "buzzer" the segmentation process is initiated starting from his current (x, y) position

Chapter 12

Sonification for Auditory Scene Understanding

12.1 An Audible Color Space Representation Inspired by Visual Color Perception

Color sonification in this part of the thesis is based on the ideas proposed in part III, except we exchanged the complementary instruments, discussed in chapter 8, with what we call **complementary sound characteristics** to represent the opponent color pairs red / green and blue / yellow. Basically, those are fundamental sound elements, such as tremolo, beats, harmonics etc. Using such fundamental sound characteristics has several benefits over common MIDI instruments:

- **Semantic Correctness**: Instruments, in general, do not give a decent representation of a color's visually perceived characteristic. Instruments are associated rather with certain objects, e.g. a choir, which in part III was assigned to red, with a cathedral. Hence, it is hard to intuitively connect this acoustical element to an object such as a tomato. In contrast, our fundamental sound characteristics might allow the user to perceive acoustically what corresponds to the visual perception of a seeing person.

- **Simplicity**: To describe every opponent color using a few, fundamental sound characteristics rather than instruments having complex sound spectra might closer to the idea of mixing elementary colors.

- **Discriminability**: Though instruments were chosen accordingly in part III, in general, they tend to interfere with one another making color and / or feature recognition unnecessary complicated. Mixing basic sound elements allows a more directed and, therefore, better separable combination of colors acoustically.

- **Loudness Independency**: The fundamental sound elements selected can be distinguished by their individual timbre, and therefore be mixed without and still be separated without the only criteria being volume. This alleviates the computation of control parameters considerably because no later compensation for a user's possible selection of too dominant instruments has to be made.

- **Usability**: Using high quality MIDI instruments requires usage of an external MIDI Synthesizer and additional linkage software as well as a certain level of expertise to connect the system with the synthesizer. While that might not be a problem for a seeing person, it definitely is to a visually impaired.

These fundamental sound elements mentioned, when mixed, create a **musical timbre**. Musical timbre is defined in [377] as:

> *That attribute of sensation in terms of which a listener can judge that two steady complex tones having the same loudness, pitch and duration are dissimilar*

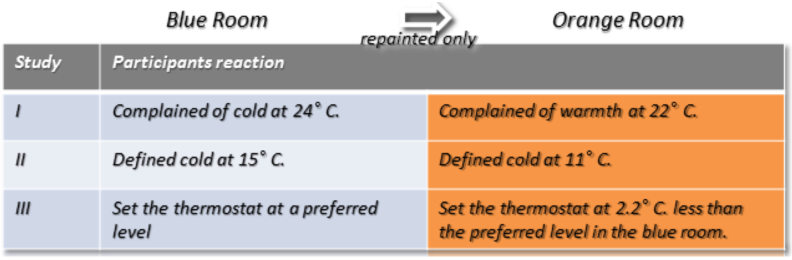
Rather than mixing instruments, which already contain a complex in time slightly changing timbre, combining fundamental sound characteristics allow for a more directed and predictable forming of specific timbres, which are, thus, easier to assign to a specific color.

Figure 12.1: Experience shows that the blue shadows on snow or ice represent coldness (right). "Warmth" perceptions of yellow and red alike colors mighte come from associations with sun, fire, wood (left)

The selection of specific sound elements to be assigned to a certain opponent color is inspired, or driven by how colors are perceived or “felt” visually. The complementary sound characteristics are chosen to convey emotions, a normal sighted person perceives, when looking at a specific color, through sound. Thus, the visual to audio transformation described in this part very closely follows the concept of **isomorphism** that Hofstadter describes in his famous work “Gödel, Escher, Bach: An Eternal Golden Braid” [211]:

> *The word “isomorphism” applies when two complex structures can be mapped onto each other, in such a way that to each part of one structure there is a corresponding part in the other structure, where ”corresponding” means that the two parts play similar roles in their respective structures. The usage of the word ”isomorphism” is derived from a more precise notion in mathematics. The perception of an isomorphism between two known structures is a significant advance in knowledge – and I claim that it is such perceptions of isomorphism which create meanings in the minds of people.*

According to **Color Theory** ([263]; [161]; [297]; [300]; [531]) colors are visually perceived individually, causing different emotional reactions. Wolfrath [297] found a significant increase in pulse and breathing frequency with stimuli of red and yellow colors, in contrast, a decrease on violet, blue color stimuli. He summarizes: *Colors are transformed, by the physiological process of vision, into feelings.* Furthermore, three separate studies confirmed the effect of color alone in determining thermal comfort levels, as illustrated in figure 12.2. A simple coat of paint varied the perception of temperature by as much as approximately 4 degrees Celsius [300]. Accordingly, there have been stated some tries of a psycho-physiological explanation for these phenomena. According to [263] and [531] it is assumed that the cool perception of cold colors is founded in associations of the human brain with blue-green ice and sea water or metal (see figure 12.1 (right)). Intuitively, the blue shadows on snow or ice represent coldness, while the complementary red nuances are perceived as warm colors. It might be further presumed that the ”warmth” perceptions of yellow and red colors come from associations with sun, fire, wood or blood (see figure 12.1 (left)).



Blue Room
repainted only
Orange Room

Study	*Participants reaction*	
I	*Complained of cold at 24° C.*	*Complained of warmth at 22° C.*
II	*Defined cold at 15° C.*	*Defined cold at 11° C.*
III	*Set the thermostat at a preferred level*	*Set the thermostat at 2.2° C. less than the preferred level in the blue room.*

Figure 12.2: Results of three studies on thermal comfort and color as described in [300]

A Physical Perspective on Color Temperature

From a physical perspective the color temperature of a specific light source is by definition the temperature of a **black body** [420] that radiates light of comparable hue to that of the light source. A black body is an idealized physical object defined in [361] as:

> *An ideal body is now defined, called a black body. A black body allows all incident radiation to pass into it (no reflected energy) and internally absorbs all the incident radiation (no energy transmitted through the body). This is true of radiation for all wavelengths and for all angles of incidence. Hence, the black body is a perfect absorber for all incident radiation.*

The color temperature of such light radiated by the black body is then defined as the black body's surface temperature in kelvins K ([501]; [420]) which permits the definition of some standard by which light sources can be compared. Thus, when talking of a lamp to have a color temperature of $\approx$ 3000 K, what is meant is that if a black body would be heated to 3000 K, it would give off the same light. This would be of course only valid if the light source in question behaves at least approximately as a black body ,which is true for the sun and most objects which simply give off light because they are hot. As an example, the Sun, has a surface temperature of $\approx$ 5500 K and its emitted radiation is mostly within the visible spectrum. Thus, light from the sun is considered to have a color temperature of $\approx$ 5500 K. Similarly, an incandescent lamp's light would be thermal radiation and the bulb approximates an ideal black body radiator. Thus, its color temperature is essentially the temperature of the filament (see figure 12.3 (right))

However, To the extent that a hot surface emits thermal radiation while not being an ideal black body radiator, the color temperature of the light does not refer to the actual temperature of its surface. A fluorescent lamp for instance, emits light primarily by processes other than thermal radiation, which means that the emitted radiation does not follow the form of a black body spectrum and thus, those light sources are assigned as a **correlated color temperature (CCT)**. Correlated color temperature is defined as the color temperature of a black body which to the human perception is most closely match the light from such a lamp (see figure 12.3 (right)).

Figure 12.3 (left) visualizes the spectral intensitiy distribution of Planck's black body radiation as a function of wavelength at different temperatures. Note that the maximum of the intensity shifts to shorter wavelengths as the black body temperature increases. Thus, from a physical point of view, colors, such as yellow and red, which are perceived as "warm" and respectively the corresponding light from lamps or fires are actually light sources having a relatively low physical color temperature.

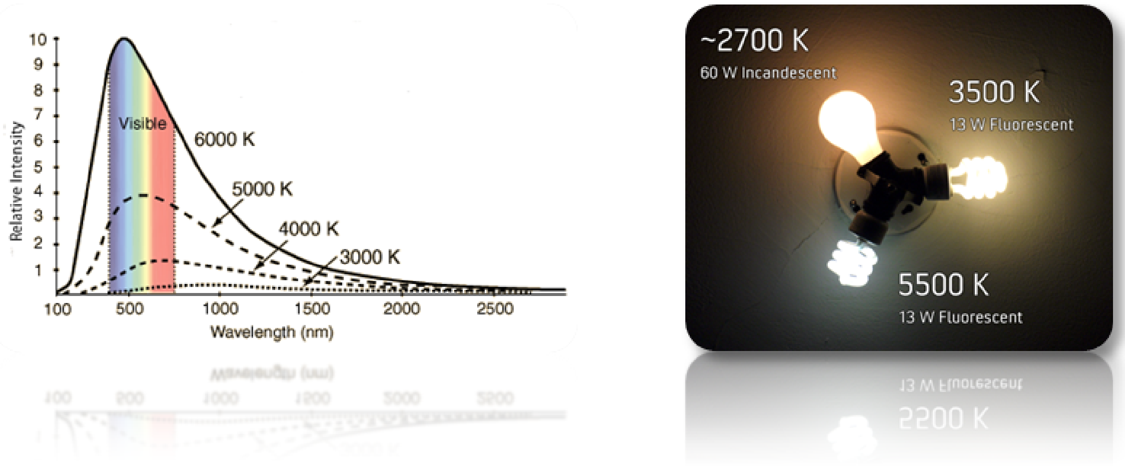


Figure 12.3: Left: The spectral intensitiy distribution of *Planck's black body* radiation as a function of wavelength at different temperatures. Picture modified from [420]. Right: Comparison of an incandescent lamp with a color temperature of approximately 2700 K, a 3500 K and a 5500 K fluorescent lamp. All lamps are all similar in light output (measured in **lumen**)

Chromesthesia

Further, some people experience the ability to taste, hear, feel or smell colors. Such a connection of the senses is known as Synesthesia ([95]; [96]). More specifically, the connection between colors and hearing is called Chromesthesia [96], illustrated in figure 12.4 (left). Although not been explained scientifically, it appears fairly frequently in history. Many great composers appear to have had this ability such as Franz Liszt or Nicolai Rimsky-Korsakov ([95]; [393]). Cytowic [96] gives an illustrative example of how especially Franz Liszt tended to express his views on music:

> *In 1842 when Liszt took over the post of Kapellmeister in Weimar, he astonished the orchestra by saying, 'Oh please, gentlemen, a little bluer, if you please! This tone type requires it!' Or, 'That is a deep violet, please, depend on it! Not so rose!' The orchestra eventually got used to the maestro seeing colors where they saw only notes.*

Vassily Kandinsky, famous artist and an accomplished musician, possessing Chromesthesia, accordingly said [238]:

> *Color is the keyboard, the eyes are the hammers, the soul is the piano with many strings. The artist is the hand that plays, touching one key or another purposely, to cause vibrations in the soul.*

However, individuals rarely agree on a mutual mapping of a specific color to a given sound [393], making a unifying sonification concept based on Chromesthesia rather infeasible.

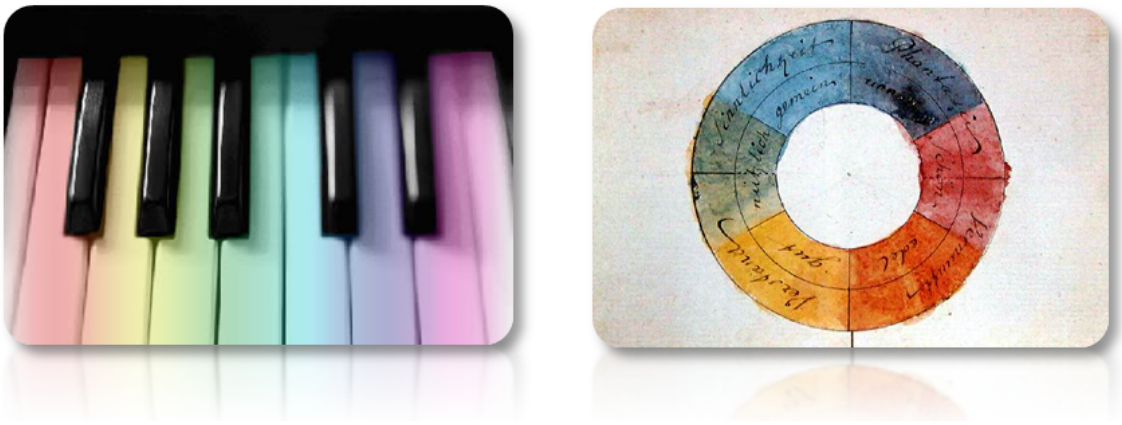


Figure 12.4: Left: An illustration of chromesthesia. Specific colors are trigged by different notes. Right: Goethe's color circle, illustrating his considerations about association of colors with certain psychological states. (Inner Ring: red - beautiful, orange - noble, yellow - good, green - useful, blue - mean and violet - unnecessary. Outer Ring: red/orange - reason, yellow/green - mind, green/blue - sensuality and violet/red - fantasy). Picture taken from [171]

Color Symbolism

Note that the verifiable color perceptions just discussed have nothing in common with what is mostly known as Color Symbolism, already considered by Goethe [171] (see figure 12.4 (right)), describing alleged associations of colors with certain psychological states or spiritual conditions. Consequentially, Gekeler [161] states:

> *Cause and effect are often not separable. Though, skepticism is recommended if specific colors are associated with certain spiritual conditions.*

Timbre Synthesis - the Creation of Colored Sounds

Timbre, as just described, refers to the “color” or quality of sounds and is typically differentiated conceptually from pitch and loudness. [504]. It is thus a multidimensional sensation that relies, among others, on the spectral energy distribution and temporal variation in this distribution ([419]; [182]). In order to better understand what the timbre feature refers to, numerous experiments have been performed ([200]; [262]; [258] ;[377]; [274]; [504]; [182] [309]). All of these experiments employ Multidimensional Scaling (MDS) analysis ([43]; [88]) to process dis-similarity judgments, partitioners were to make on pairs of sounds. Multidimensional Scaling (MDS) analysis also represents the sound stimuli in a low-dimensional space to uncover potential underlying attributes that listeners might employ. This low-dimensional representation is often referred to as a “Timbre Space” [209]. Investigations to describe musical timbres using common adjectives has been issue of considerate research for various purposes ([528]; [527]; [409]; [408]; [112]; [111]; [134]; [491]; [110]; [213]; [178]; [227]; [113]). Zacharakis et al. ([528]; [527]) conduct a study on the verbal attributes of musical timbre to identify the most significant semantic descriptors and to quantify the association between prominent timbral aspects. Factor and cluster analysis is performed on the subjective evaluation data in order to shed some light on the relationships between the proposed adjectives. Figure 12.5 shows the results of their studies in [528] as the results of a hierarchical cluster analysis based on squared Euclidean distances over 27 found common verbal descriptors.

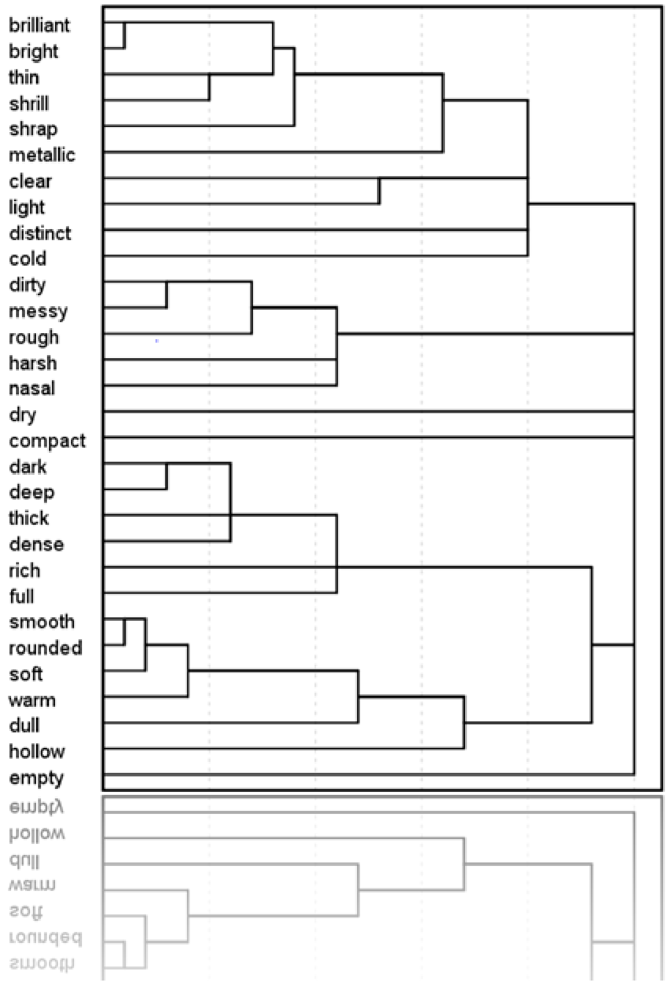


Figure 12.5: Dendrogram of the hierarchical cluster analysis over 27 found common verbal descriptors. Picture taken from [528]

Interestingly, user studies not only indicate the behaviour of assigning similar words to certain types of sounds, and therefore reveal a rather universal mapping between words and timbre ([409]; [408];). They further reveal, that it is reasonable to suggest that the description of musical sounds is not influenced by musical background and training and is rather an innate skill ([409]; [408]; [112]; [111]). These findings are quite encouraging for our approach to find timbres that are described by salient semantic descriptors, such as "warm", that match the visual descriptions for specific colors, as they ensure that our selected associations might be intuitive not only for a small group of trained participants.

Timbre Synthesis based on Complementary Sound Characteristics

Table 12.1 gives a brief overview over the visual perceptions associated with the opponent colors, according to ([263]; [297]; [300]; [531]), as well as the sound characteristics we assign as an acoustical representation.

Opponent Colors	**Visual Perception**	**Complementary Sound Characteristic**
black/white (gray-scale)	monotonous, sad	single sine wave
red	alive, vibrant	tremolo
green	calm	third + slow patterns of beats
yellow	warm	bass + even harmonics
blue	cold	flute model

Table 12.1: Our proposed visual to acoustical "isomorphism"

In words, all color sound synthesis starts off with a single (monotonous) **sine wave** for gray-scale values, called the "fundamental sine". With red, a **tremolo** effect [503] is created adding a second sine wave, equal in loudness to the fundamental sine, just a few Hertz apart. The superimposition of such two sine waves is known as a **first order beat** [392] and creates the tremolo effect if both frequencies are very close (differing by < 5 Hz). The more red the color turns, the smaller we tune the gap between both frequencies, increasing in speed of the perceived tremolo.

To create an opponent sound characteristic to such a vibrant red, we represent the rather calming green, as a calm motion of sound in time using an additional sine wave tuned to a classical third to the fundamental sine wave, forming a third chord, as well as two further sine waves, one tuned almost similar to the fundamental sine, the other almost similar to the second sine to form a pair of beats. In contrast, frequencies are set far enough apart to create not the vibrant tremolo effect but a smooth temporal varying sound pattern. All additional sine waves have the same loudness as the fundamental sine, which leads to the perception of a true chord, instead of harmonics, and therefore creates depth to the sound [199].

To simulate the visual perception of "warmth" with yellow, we increase the volume of bass, as encouraged in [199], created by two additional sine waves tuned 4 and 8 octaves below the fundamental sine wave, as well as the number and loudness of up to 8 additional sine waves, tuned to the frequencies of only the **even harmonics** of the fundamental sine wave, as motivated in [423]. "Warmth" in sound is a term that has been proven to be difficult to define. However, it is widely agreed that it deals largely with low and low mid-range frequencies ([505]; [320]; [321];), thus our selection. The resulting sound resembles a bit that of an organ. Note that additional sine waves are adjusted in loudness. For instance, the harmonics are played several times quieter than the fundamental sine to not be perceived as a complex chord with the fundamental sine. The possible loudness increase intervals of the two additional sine waves creating bass are chosen to avoid auditory masking as well as to not be perceived as a new fundamental tone. These considerations are implemented according to the experimental results of the equal loudness curves, discussed in section 2.1.

The coldness of blue was originally planned to be sonified adding only *odd harmonics* which leads to a **square wave**, creating a "cold" and mechanical sound. However, the sound so produced is too annoying to be used, so we applied one of the Synthesis Toolkit's pre-defined physical instrument models, which is able to synthesize the sound of a rough flute or wind. Interestingly, those instruments contain mostly odd harmonics. As discussed in section 3.1, a beneficial property of such models is that an increase in loudness goes along with as increase in timbral intensity and is, therefore, suitable to represent a change in color towards intense blue.

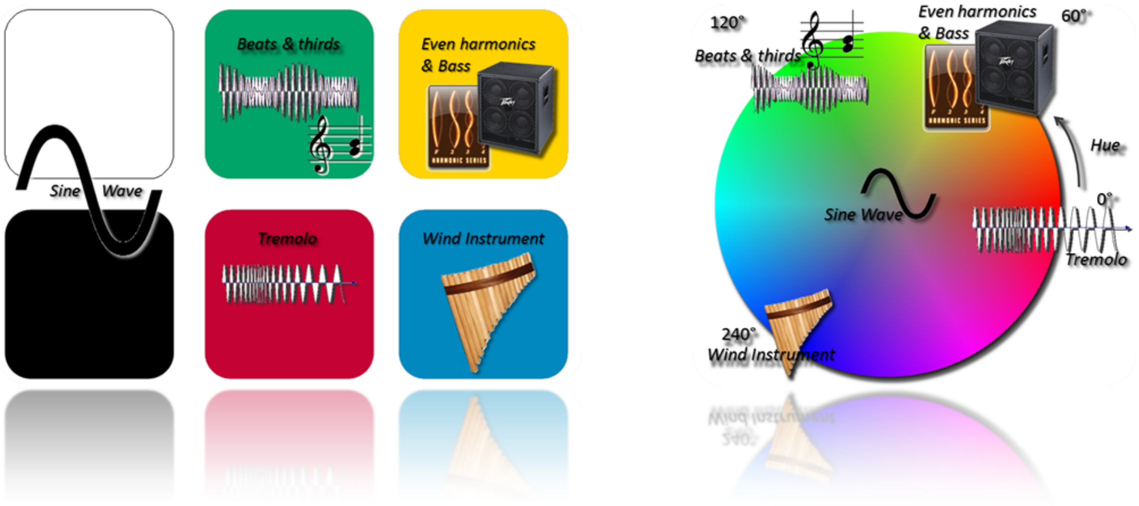


Figure 12.6: Left: The opponent colors are represented by complementary sound characteristics. Right: An audible color space inspired by visual perception

Note that our selection is inspired by the previously discussed observations on visual color perceptions. It does, however, not claim to be the only or most appropriate selection to convey such perceptions and it would be an interesting issue for further research, whether our or any other selection of sound characteristics can be verified to trigger an equivalent emotional response to the visual perception of a color.

Luminance Sonification

As motivated in part III, different luminances are represented by changes in pitch, in the form of a musical scale from $C_4 = 261.626$ Hz to $C_5 = 523.251$ Hz. Again, for harmonic reasons, we only utilize the whole tones of the octave and map each lightness value l between 0 and 255 to one of the eight tones of the scale, as discussed in section 8.1. However, we remove the thirds in between to not confuse with the thirds partially representing green.

Note that because we are no longer bound to the specifications of MIDI and synthesizing own sounds and acoustical timbres "from scratch", the usage of a continuous pitch instead of rather discrete notes to map lightness could be considered. However, as also already mentioned in section 8.1, we abstain from doing that to add another interesting semiotic element, emphasizing the sad monotony of gray scale values in contrast to the rather "joyful" colors.

Any time, a pixel value is considered to be gray scale, the sonification of its lightness is performed according to a rather "sad" **C natural minor scale** (figure 12.7 (right)), instead of the **C major scale** (figure 12.7 (left)) used with the lightness sonification of colorful pixel values.

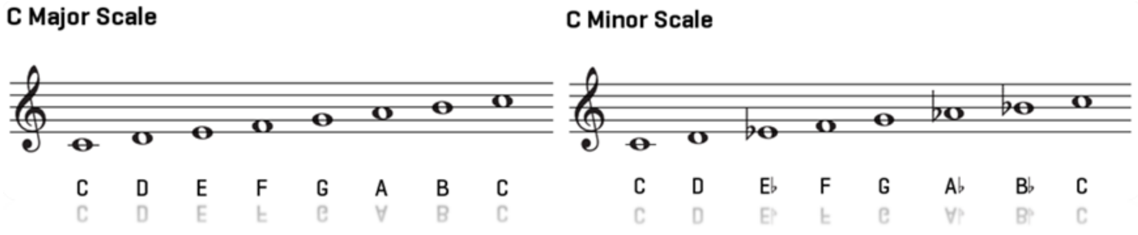


Figure 12.7: Left: C major scale. Right: C natural minor scale

Note that as timbre is typically differentiated conceptually from pitch and loudness [504], we can additionally use these dimensions to support the perception of color saturation, using loudness, as well as changes in lightness based on pitch, without the risk of unintended changes in the sound characteristic of a specific color.

Computation of the Audible Color Space

Calculating mixture relations between our sound elements makes use of the control entity structure, called *volume shape* $\vartheta(h, s)$,as presented in part III of the thesis. In part III a volume shape $\vartheta(h, s)$ for each instrument mapped a volume ϑ from 0 to 1 to each color (h, s), regardless of the lightness l.

We make use of this idea to control our sound parameters, except that we incorporate lightness now. As on can see in the enrolled *HSL* space (see figure), as lightness increases or decreases say above or below 50 percent it significantly looses its intensity not unlike the fade in saturation. These properties of the *HSL* model should be considered in its acoustical representation.

Further investigation in the HSL color space additionally revealed two major irregularities that have to be considered as well:

- Below 50 percent luminance around yellow ($h = 60°$) there is a certain region that would be visually perceived as “olive green” rather than dark yellow (see figure 12.8 (left)).
- Additionally, what should be a visual “deep blue” at $h = 240°$ increasing in luminance tends to fade into what is perceived visually as “violet” (see figure 12.8 (right)).

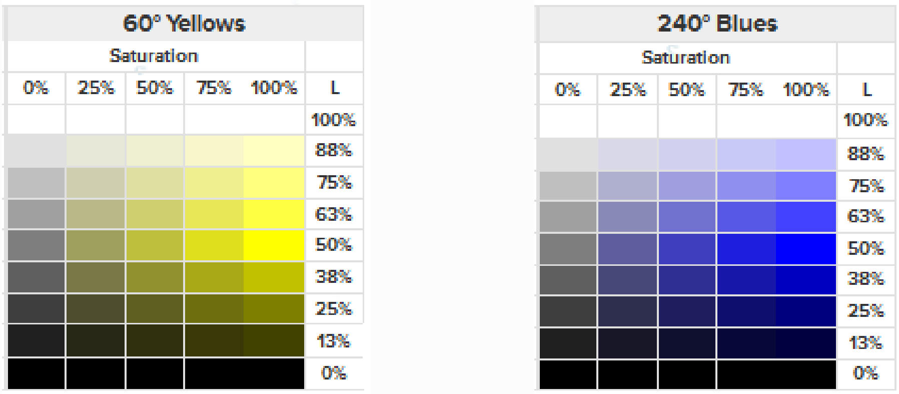


Figure 12.8: Irregularities within the HSL color model. Picture taken from [441]

Consequentially, we needed to adjust our acoustical color space representation to compensate for the audible perception to fit the visual perception.

A Timbre-Based Color Sound Synthesis Equation

To incorporate lightness as well as irregularities within the *HSL* model, we take the idea of volume shapes $\vartheta(h, s)$ a step further and pre-compute a control parameter value for each sound characteristic at every position within the HSL model, calling it *sound parameter volumetric* $\vartheta(h, s, l)$.
Before going into detail about adjusting the audible color space and the computation of $\vartheta(h, s, l)$, we can now formulate a new *color sound synthesis equation* that describes the mapping from color values (h, s, l) stored within each augmented visual pixel $\boldsymbol{v}(x, y)$ into a sound, referred to as the color sound attribute $a_{color}(x, y)$ of the audible pixel $\boldsymbol{a}(x, y)$, as a try to convey visual color perceptions and physiological effects through the sense of hearing:

$$\begin{aligned} a_{color}(x,y) &= \vartheta_{grey}(h,s,l) * Sine(\eta) \\ &+ \vartheta_{red}(h,s,l) * \boldsymbol{vibrant}(\eta, \vartheta_{red}) + \vartheta_{green}(h,s,l) * \boldsymbol{calm}(\eta, \vartheta_{red}) \\ &+ \vartheta_{yellow}(h,s,l) * \boldsymbol{warm}(\eta) + \vartheta_{blue}(h,s,l) * \boldsymbol{cold}(\eta) \end{aligned} \tag{12.1}$$

with:

$$h = s_h(x,y) = v_h(x,y), \quad s = s_s(x,y) = v_s(x,y), \quad l = s_l(x,y) = v_l(x,y)$$

and:

$$\boldsymbol{warm}(\eta) = \underbrace{Sine(\frac{\eta}{4}) + Sine(\frac{\eta}{8})}_{bass} + \underbrace{\sum_{i}^{8} Sine(\eta * 2 * i)}_{harmonics}$$

$$\boldsymbol{cold}(\eta) = \boldsymbol{flute}(\eta)$$

$$\boldsymbol{calm}(\eta, \vartheta_{red}) = \underbrace{Sine(\eta_{3rd})}_{third}$$

$$+ \underbrace{Sine(\eta + 30Hz * \vartheta_{red}(h,s,l)) + Sine(\eta_{3rd} + 30Hz * \vartheta_{red}(h,s,l))}_{Beats}$$

$$\boldsymbol{vibrant}(\eta, \vartheta_{red}) = \underbrace{Sine(\eta + 5Hz * \vartheta_{red}(h,s,l))}_{tremolo}$$

and:

$$\eta = \begin{cases} note_{major}(l), & \text{if } \vartheta_{gray}(h,s,l) = 1 \\ note_{minor}(l), & \text{otherwise} \end{cases}$$

The 1D lookup structures $note_{major}(l)$ and $note_{minor}(l)$ employs the same the mapping of l values to 1 of the eight tones, as used in section 8.1 and discussed in appendix B.2. However, $note_{major}(l)$ is the lookup table C major scale, $note_{minor}(l)$ that of a C natural minor scale, both containing corresponding frequencies (see table 12.2).

key $note_{major}(l)$	**key** $note_{minor}(l)$	**lightness range**
C (261.6 HZ)	C (261.6 HZ)	0 – 10
D (293.67 HZ)	D (263.67 HZ)	11 – 37
E (329.62 HZ)	Es (311.13 HZ)	38 – 63
F (349.23 HZ)	F (349.23 HZ)	64 – 101
G (392 HZ)	G (392 HZ)	102 – 153
A (440 HZ)	As (415.3 HZ)	154 – 179
H (493.88 HZ)	B (466.16 HZ)	180 – 242
C (523.25 HZ)	C (523.25 HZ)	242 – 255

Table 12.2: Lookup structures $note_{major}(l)$ and $note_{minor}(l)$

Note, that in contrast to the color sonification approach presented in part III, the fundamental sound characteristic that represents gray also builds up the foundation for 3 of the 4 opponent color sonifications and is, therefore, never switched off.
Further, the lookup table $note(l)$ depending on gray or not.. Based on a musical scale, as shown in figure 8.2 (right), black, as the lowest lightness value, is assigned to the tonic keynote, whereas white to its octave. In between there are six whole tones and 12 semitones. For harmonic reasons we only utilize the whole tones of a single octave and map each lightness value l between 0 and 255 to one of the eight tones, forming a $1D$ lookup structure $note(l)$. Note that the mapping of l values to specific tones, discussed in appendix B.2, was determined judging by our own visual perception of approximately equal lightness intervals.

The sound synthesis method employed would be additive synthesis for the sonification of red, yellow and green, as described in section 3.1, implemented using the *Synthesis Toolkit (STK)* library [83], briefly described in appendix A.9. The sonification of blue is based on a physical instrument model, which is also a part of the *Synthesis Toolkit (STK)*.

Computation of Sound Parameter Volumetrics

Although the general concept of *sound parameter volumetrics* $\vartheta(h, s, l)$ is based on the idea of volume shapes $\vartheta(h, s)$, their computation has been significantly varied. Opposed to color sonification in part III based on volume shapes, we quantize the acoustical color representation not only in l, but also along h. The sacrifice of continuity in mapping along h, comes with the crucial benefit to gain more control over the resulting sound, in other words to better match the visual perception to the audible perception.

The quantization along h is inspired by the human eye's inherent limitations to distinguish color differences [515]. Generally, *MacAdam ellipses* refer to the region on a *chromaticity diagram* [299], [514]. Such an ellipse, illustrated in figure 12.9 (left), contains all colors that are indistinguishable, to the (average) human eye, from the color at the center of the ellipse. The contour of the ellipse, therefore, represents the *just noticeable differences (JND)* of chromaticity. Figure 12.9 (right) shows the application of JND ellipses in the *Cielab* color space.

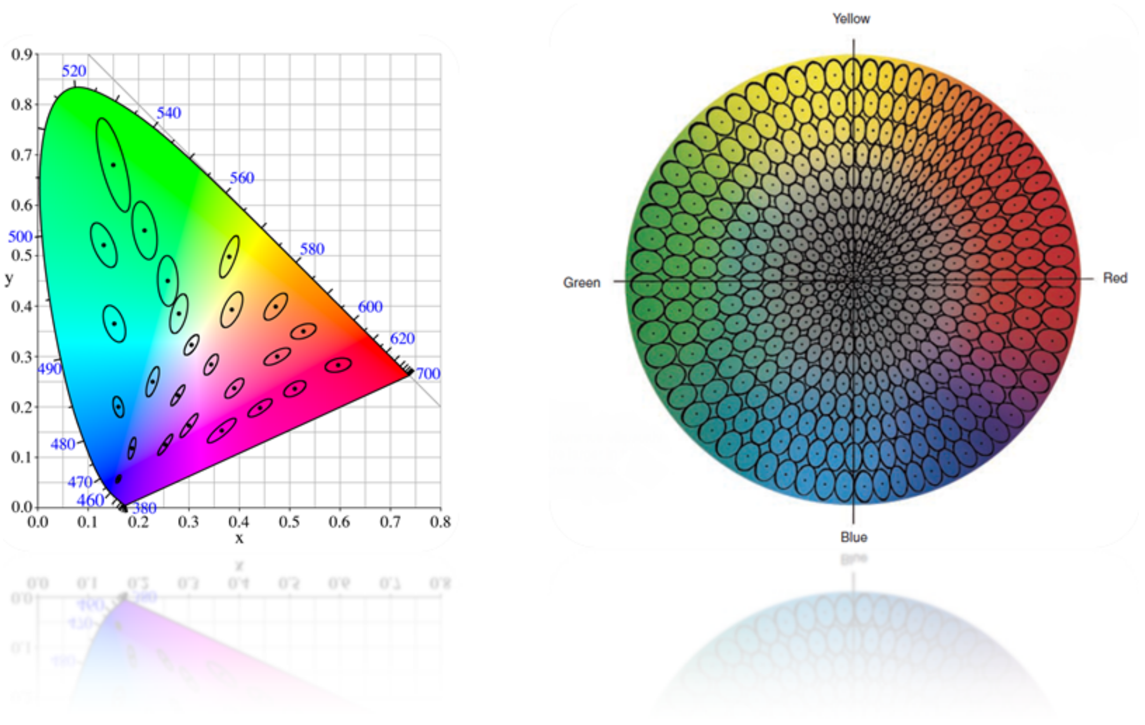


Figure 12.9: Left: MacAdam ellipses for one of MacAdam's test participants. Illustration build based on [299]. Right: ellipsoid of "equally" perceived colors in the *CieLab* color space. Picture modified from [515]

Hence, we define regions of approximately "equally perceived" color values within our audible color space, though those regions will be significantly greater than the JND of visual color perception. Note that such regions are chosen by judging by our own visual observations of approximately equal hue intervals. However, there has been some research on "human perception based color segmentation" that might corroborate our selection ([479]; [426];[370]; [241]; [6]; [80]). To avoid unnatural jumps at borders we define a linear transition between regions. Table 12.3 shows our defined hue intervals of approximately "equally perceived" color values, and transitions in between, within our audible color space, at 50 percent luminance and 100 percent saturation.

Perceived Color (mixture)	**hue range**	ϑ_{red}	ϑ_{yellow}	ϑ_{green}	ϑ_{blue}
red	0° – 14°	255	0	0	0
transition	15° – 19°	0	0	0	0
red / orange	20° – 29°	255	64	0	0
transition	30° – 34°	0	0	0	0
orange	35° – 40°	255	255	0	0
transition	41° – 47°	0	0	0	0
yellow / orange	48° – 49°	64	255	0	0
transition	50° – 56°	0	0	0	0
yellow	57° – 64°	0	255	0	0
transition	65° – 69°	0	0	0	0
yellow / green (olive)	70° – 84°	0	64	127	0
transition	85° – 89°	0	0	0	0
green / yellow	90° – 91°	0	64	255	0
transition	92° – 99°	0	0	0	0
green	100° – 159°	0	0	255	0
transition	160° – 169°	0	0	0	0
cyan	170° – 189°	0	0	64	127
transition	190° – 194°	0	0	0	0
light blue	195° – 219°	0	0	0	127
transition	220° – 225°	0	0	0	0
dark blue	226° – 264°	0	0	0	255
transition	265° – 269°	0	0	0	0
blue / purple	269° – 269°	64	0	0	255
transition	270° – 274°	0	0	0	0
purple	275° – 284°	255	0	0	255
transition	285° – 289°	0	0	0	0
red / purple (magenta)	290° – 334°	255	0	0	64
transition	335° – 339°	0	0	0	0
red	340° – 360°	255	0	0	0

Table 12.3: Our defined hue intervals of approximately "equally perceived" color values

Note that in figure 12.9 (right) the "equally perceived color" ellipses within the orange area are longer and narrower than the broad and rounder ones in the green area. As one can see in table 12.3, such property naturally flew into our hue intervals of approximately "equally perceived" color.

Based on table 12.3 all sound parameter volumetric values $\vartheta(h,s,l)$ at $l_{50\%} = 127$ can now be computed (see figures 12.10 and 12.11). For all $\vartheta_{(r)ed}$, $\vartheta_{(y)ellow}$, $\vartheta_{(g)reen}$, $\vartheta_{(b)lue}$, a decrease in saturation causes a linear decrease in $\vartheta(h,s,l)$:

$$\vartheta_{r,y,g,b}(h,s,l_{50\%}) = \begin{cases} \vartheta_{r,y,g,b}(h,s_{100\%},l_{50\%}) * \frac{s-s_{min}(l_{50\%})}{s_{100\%}-s_{min}(l_{50\%})}, & \text{if } s > s_{min}(l_{50\%}) \\ 0, & \text{otherwise} \end{cases}$$

with

$$s_{100\%} = 255, \quad l_{50\%} = 127$$

Note that any point where $\vartheta(h,s,l) = 0$ is, therefore, not necessarily the center of the (h,s,l) color space (along l), but the value of the minimum saturation lookup structure $s_{min}(l)$ computed in part III.

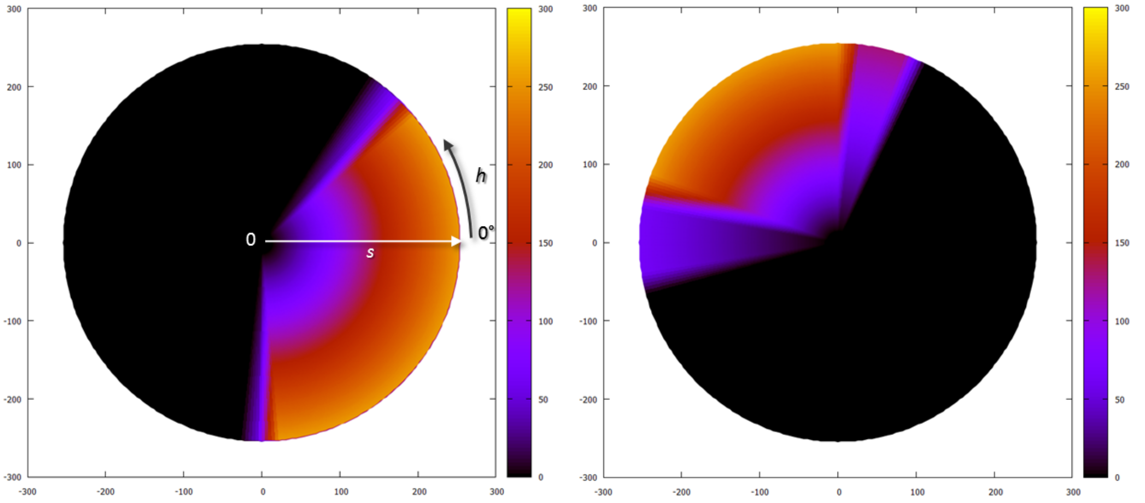


Figure 12.10: $\vartheta_{red}(h,s_{100\%},l_{50\%})$ & $\vartheta_{green}(h,s_{100\%},l_{50\%})$. Note that the range of values for each $\vartheta(h,s)$ is from 0 (black) to max. 255 (orange-yellow)

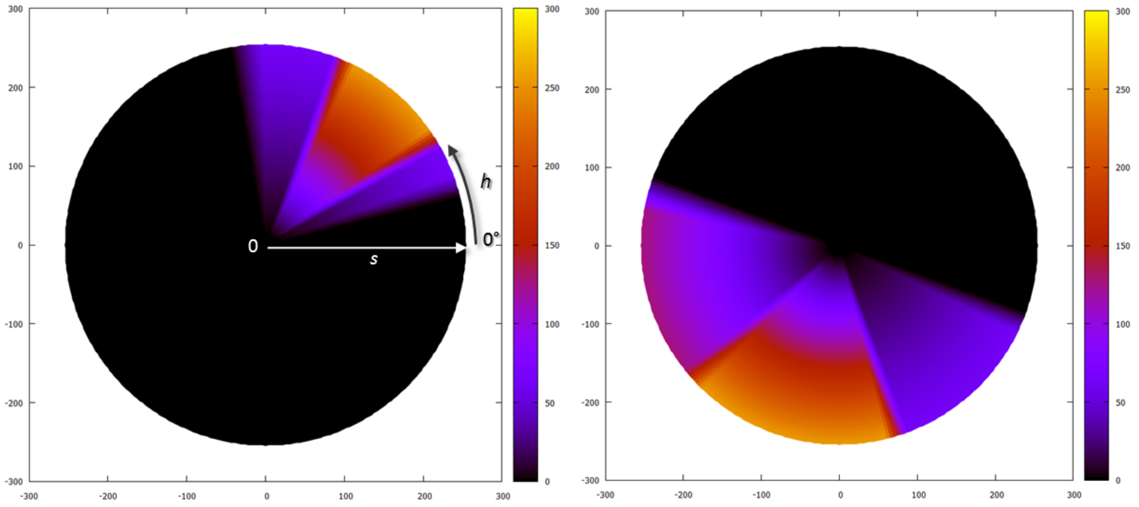


Figure 12.11: $\vartheta_{yellow}(h, s_{100\%}, l_{50\%})$ & $\vartheta_{blue}(h, s_{100\%}, l_{50\%})$. Note that the range of values for each $\vartheta(h, s)$ is from 0 (black) to max. 255 (orange-yellow)

The computation using the non-linear *Thin Plate Splines (TPS)* interpolation method as in part III is inappropriate given many control points that describe linear intervals and transitions. Furthermore, the fundamental sound elements selected can be distinguished by their individual timbre, and therefore be mixed and still be separated without the only criteria being volume. Hence, no later compensation for a user's possible selection of too dominant instruments has to be made, which was another main motivation in part III to work with TPS. However, the TPS method will still be exploited to deal with some of the previously mentioned visually perceived irregularities. In order to not disturb the textual flow at this point the explanation of how we deal with these irregularities has been shifted to section 2.1.

To support the perception of changes in a color's perceived "colorfulness" as a color gets darker or lighter than $l = 50\%$ values of $\vartheta(h, s, l)$ decrease, starting from $l = 50\%$, according to:

$$\vartheta_{r,y,g,b}(h, s, l) = \begin{cases} \vartheta_{r,y,g,b}(h, s, l_{50\%}) * (1 - \frac{\|l - l_{50\%}\|}{l_{50\%}}), & \text{if } s > s_{min}(l_{50\%}) \\ 0, & \text{otherwise} \end{cases}$$

with

$$l_{50\%} = 127$$

Amplitude Envelope

To avoid nasty “cracks” in the output signal, it is crucial to prevent color sonification from “jumps” in the final volume. In part III, this was internally solved by the *MIDI* synthesizer. Each *MIDI* instrument followed a general amplitude envelope structure [114] as illustrated in figure 12.12(left). Basically, such an amplitude contains 4 phases:

- *Attack*: Describes the time an instrument needs to run from zero volume to its peak value, when a note is initially triggered, such as a key on a piano is first pressed.
- *Decay*: Denotes the time taken for the subsequent decay from the attack level to a designated sustain level.
- *Sustain*: describes the amplitude level during the main sequence of the sound's duration, until the note is released.
- *Release*: describes the time necessary for the run down from the sustain level to zero after the note is released.

In the color sound synthesis presented in this part, the final volume is directly depended on the *sound parameter volumetric* $\vartheta(h, s, l)$ and the queue of sonification descriptors, as discussed in chapter 5.4, offers a very elegant way to create our own amplitude envelope. Thus, we compare all ϑ of the current and the next sonification descriptors $\boldsymbol{s}(x, y)_{curr.}$, $\boldsymbol{s}(x, y)_{next}$ to be converted into sound and check whether their difference $\vartheta_{next} - \vartheta_{curr.}$ would be greater than a specific threshold $\triangle_{max}\vartheta$. The sign of the difference indicates whether the volume of the sound increases or decreases.

If $|\vartheta_{next} - \vartheta_{curr.}| > \triangle_{max}\vartheta$ we create several copies $\boldsymbol{s}_i(x, y)_{curr.}$ of $\boldsymbol{s}(x, y)_{curr.}$ with intermediate equidistant (increasing or decreasing) ϑ_i:

$$\triangle\vartheta = \frac{\vartheta_{next} - \vartheta_{curr.}}{n} \tag{12.2}$$

with

$$n = \lceil \frac{\vartheta_{next} - \vartheta_{curr.}}{\triangle_{max}\vartheta} \rceil \tag{12.3}$$

n denotes the number of intermediate sonification descriptors $\boldsymbol{s}_i(x, y)_{curr.}$ needed so that each ϑ_i is below $\triangle_{max}\vartheta$. Finally:

$$\vartheta_{i+1} = \vartheta_i \pm \triangle\vartheta \tag{12.4}$$

Consecutively, the created intermediate sonification descriptors $\boldsymbol{s}_i(x,y)_{curr.}$, $\boldsymbol{s}_{i+1}(x,y)_{curr.}$, ..., $\boldsymbol{s}_n(x,y)_{curr.}$ are positioned within the sonification queue between $\boldsymbol{s}(x,y)_{curr.}$, $\boldsymbol{s}(x,y)_{next}$. The so created smooth amplitude transition between different pixel positions shapes an amplitude envelope of the form as illustrated in figure 12.12(right). Note that the decay phase is not needed within our color sound synthesis.

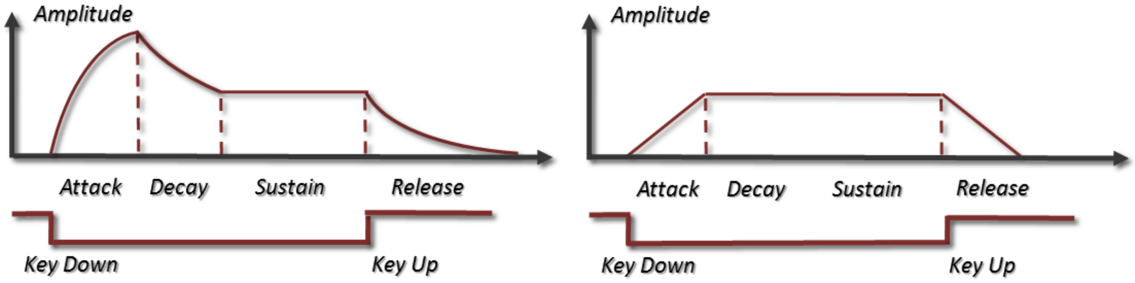


Figure 12.12: Left: The general amplitude envelope structure [114]. Right: The amplitude envelope for smooth transitions between pixel positions

12.2 Classified-Region Dependent Feature Sonification

The sonification of complex features, such as roughness, highest gradients, line patterns and objects, is processed using pre-computed wave-files with the *irrklang* Audio Engine [8] (see appendix A.6). It allows for post-processing of pre-computed wave-files with sound effects such as looping, volume change, playback-speed change, pitch change, reverberation, echo, *stereo panning* and $3D$ sound positioning. We harness such possibilities to convey complex features audibly along with colors without additional computational effort to synthesize such sounds additionally. Further, the usage of external audio files allows for an easy exchange of sounds.

Natural Regions Represented by Brown Noise

Natural regions are sonified using pre-recorded *brown noise* as an intuitive acoustical roughness representation, as discussed in section 3.2. Its spectral density is inversely proportional to f^2, meaning it has more energy at lower frequencies, which gives brown noise a "damped" or "soft" sound, in contrast to *white* and *pink noise*. This makes it more comfortable to work with and it is less likely tends to mask other sonifications. It sounds like a low roar resembling a waterfall or heavy rainfall. Additionally, we make use of stereo panning to support localization as well as the estimation spatial propagation of natural structures within the image easier.

The following equation shows the mapping of $v_{FD}(x, y)$ into the final sound, i.e. the audible pixel's $a_{natural}(x, y)$, with $2D$ spatialization already incorporated:

$$a_{natural}(x, y) = (vol_{left}(x, y) + vol_{right}(x, y)) * vol_{natural}(x, y) * BrownNoise() \quad (12.5)$$

with

$$vol_{natural}(x, y) = \begin{cases} vol_{max,natural} * v_{FD}(x, y), & \text{if (x,y)} \in \text{natural region} \\ 0, & \text{otherwise} \end{cases}$$

$vol_{max,natural}$ was been experimentally chosen to be low enough to not mask other sonifications.

Man Made Structures Acoustically Encoded in Rhythms

Regions belonging to man made structures are acoustically represented using two different rhythms, say Ω_{lp} and Ω_{hg}, played by two different percussion instruments, again prerecorded as sound files.

The perception and analysis of rhythm and rhythmic dissimilarities, in general, has been topic of thorough research ([470]; [468]; [425]; [424]; [469]; [188]). The selection of a rhythmic representation for structure sonification is motivated by two major benefits:

- As discussed in section 2.1, the brain is very sensitive to temporal and spectral variations, which makes the application of specific rhythms an ideal instrument to convey information audibly.
- Rhythms that vary primarily in temporal rather than spectral dimensions tend to less likely interfere with, e.g., the color sonification.

The last observation is partly corroborated by Cullen and Coyle [94], who employ rhythmic parsing for the sonification of DNA and RNA sequences. They state:

> *It is often the case that sonified audio has little or no rhythmic component, and it is felt that as rhythm is such an important part of the musical analysis process it should be given far more serious consideration when representing mathematical data as audio.*

Other than that, in our context, the usage of rhythm can be motivated that by intuition the occurrence of edges as well as repeating line patterns, as singular peaks within a continuous lightness level, might be very intuitively conveyed into a regular repeating pattern of impact sounds.

Buildings & Line Patterns

The first rhythm Ω_{lp}, played on a *wooden bongo drum*, see figure 12.13(left), represents the presence of man made structures in general. It is heard as soon as the user moves into an area that is classified to be man made.

This first rhythm is additionally utilized to emphasize any additional occurrence of line patterns, that is what *lp* in Ω_{lp} stands for. For this purpose, we employ *reverberation* [440], [476]. It is the persistence of sound in a particular space after the original sound is produced. A reverberation, or reverb, is created when a sound is produced in an enclosed space causing a large number of echoes to build up and then slowly decay as the sound is absorbed by the walls and air.

The following equation shows the mapping of line patterns to sound, i.e. the audible pixel's $a_{man\,made\,1}(x,y)$, with 2D spatialization already incorporated:

$$a_{man\,made\,1}(x,y) = (vol_{left}(x,y) + vol_{right}(x,y)) * vol_{man\,made\,1} * rev(\Omega_{lp},(x,y)) \quad (12.6)$$

with

$$rev(\Omega_{lp},(x,y)) = \begin{cases} \text{reverberation with intensity of } v_{n_{\|}}(x,y) \text{ on } \Omega_{lp}, & \text{if } v_{n_{\|}}(x,y) > 1 \\ \Omega_{lp}, & \text{otherwise} \end{cases}$$

and

$$vol_{man\,made\,1} = \begin{cases} vol_{max,man\,made\,1}, & \text{if (x,y)} \in \text{man made structure} \\ 0, & \text{otherwise} \end{cases}$$

Note that the specific orientation of the line patterns is not additionally separately as the user can guess such orientations by assuming the same as such of the highest gradient which will be sonified using a different drum rhythm, described below. This is due to keep level of confusion, caused by too many acoustical signals as low as possible and to transport as much as possible information with as little as possible acoustical entities.

Highest Gradient

The second rhythm Ω_{hg} is to represent the pixel element referring to the mapped orientation α_1 of the highest gradient. The rhythm is performed by the opening and closing of a *hi hat* percussion instrument, see figure 12.13 (right). It is altered in pitch and speed depending on whether $v_{\varphi\nabla_1}(x,y)$ is more 0° or 90°. Hence, if an edge at (x,y) is more horizontal the rhythm will be played slow and deep. If it is more of a vertical edge, the rhythm is adapted to be played quite fast and high. If no edge is present at (x,y) at all, Ω_{hg} is set mute.

$a_{man\,made\,2}(x, y) =$

$$(vol_{left}(x, y) + vol_{right}(x, y)) * vol_{man\,made\,2} * (vel(\Omega_{hg}, (x, y)), p(\Omega_{hg}, (x, y))) \qquad (12.7)$$

with

$$vel(\Omega_{hg}, (x, y)) = \begin{cases} \frac{v_{\varphi_{\nabla_1}}(x,y) - vel_{min}}{vel_{max} - vel_{min}}, & \text{if } v_{\varphi_{\nabla_1}}(x, y) \neq -1 \\ 0, & \text{otherwise} \end{cases}$$

and

$$p(\Omega_{hg}, (x, y)) = \begin{cases} \frac{v_{\varphi_{\nabla_1}}(x,y) - p_{min}}{p_{max} - p_{min}}, & \text{if } v_{\varphi_{\nabla_1}}(x, y) \neq -1 \\ 0, & \text{otherwise} \end{cases}$$

and

$$vol_{man\,made\,2} = \begin{cases} vol_{max,man\,made\,2}, & \text{if (x,y)} \in \text{man made structure} \\ 0, & \text{otherwise} \end{cases}$$

vel_{max}, vel_{min}, p_{max} and p_{min} denote the maximum or respectively minimal values in speed (vel) and pitch (p) that $v_{\alpha}(x, y)$ will be mapped to.

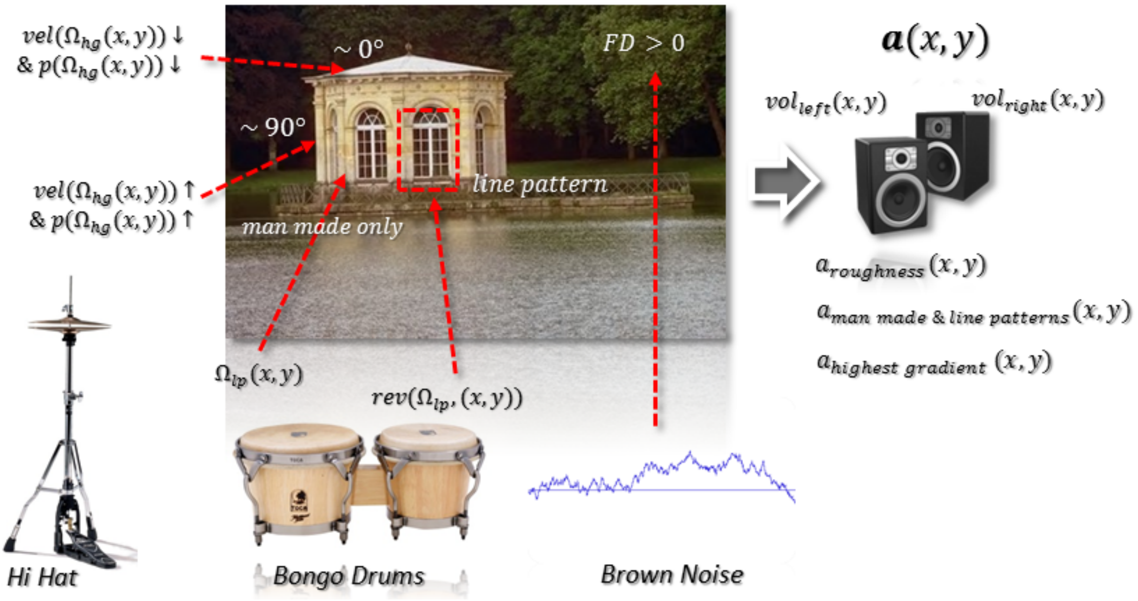


Figure 12.13: Man made and natural region sonification “in action”

12.3 Audible Object Detections Based on Auditory Icons

The objects i that are found and verified are sonified using familiar *auditory icons*, such as the "meow" produced by a cat or the barking of a dog, so no abstract memorization is required. The icon is played whenever the user moves over a pixel region referring to a specific object. Additionally, we use 2D sound positioning to support spatial localization. The short sounds of about $1-2$ seconds length are repeatedly played quietly as long as the user moves within the object region (i.e. its bounding rectangle $rect_i$), so he can shift attention to color or texture features. Our system prioritizes the detection of objects over the more general "man made structure" and "natural region" discrimination. This means, the sonification of man made or natural regions will both be at first suspended within areas of detected objects. However, besides the sonification of the general occurrence of the object i within a certain region in the image, given by its bounding rectangle $rect_i$, we will activate the sonification of man made and natural low-level features again individually, depending on whether the current object i at (x, y) belongs to a more "natural" or rather "man made" object class c_i:

$$\text{activate sonification of} = \begin{cases} \text{man made features,} & \text{if } c_i \in \{car,\ airplane\} \\ \text{natural features,} & \text{if } c_i \in \{car,\ person,\ horse\} \end{cases}$$

In case of multiple object detections at (x, y), the class c_i of that object detection i that exhibits the highest $v_{det.}(i, c_i)$ is selected.

12.4 On & Off Screen Indication Based on Earcons

When working with a touch screen as an exploration device, presented in section 5.3, we faced the problem that the user's finger position tended to be not registered sometimes. Hence sonification stopped, giving the user the impression that he was maneuvering outside the image. Thus, we implemented an additional sonification that simply signals the user whether contact with the screen is lost or connected. The approach makes use of *Earcons* which are already familiar to the user in a different context. We apply the typical sound generally used to signalize that a peripheral device, such as a mouse, has been connected or disconnected to the usb plug. Figure 12.14 shows the typical structure of this on screen / off screen sound.

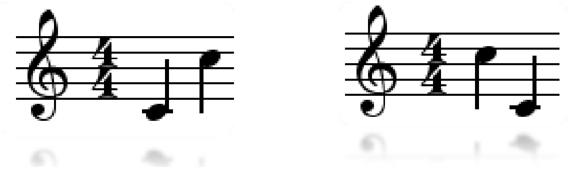


Figure 12.14: Left: On Screen Earcon. Right: Off Screen Earcon

Chapter 13

User Studies & Discussion

After 9 months we consulted the same group of participants as in part III for further studies with the refined and enhanced framework. Unfortunately, because of illness, we were deprived of one of the teenage participants. Note that the participants had no further encounter or possibility to work or train with our framework in between. They received only a 10 minutes summary of the changes in the sonification in the current system and almost no training time (approximately 5 minutes of personal interactive exploration) before we started with the experiments. As in part III, we utilized a *Touch Screen* for all tests.

13.1 Experiments

Experiment I - Object Recognition by Color Only

The goal of the first experiment is to verify that the new color sonification concept is as useful and informative as the one presented in part III. Hence, the setup is kept similar to the one in chapter 9, except for the number of trials, which is dropped from 60 to 40 this time. The task is to identify objects by color only, while all other sonification is deactivated. The stimuli are 40 photographs that show one out of 4 elements (orange, tomato, apple and lemon) in different positions (see figure 9.2). In each of 40 trials, one image is selected at random and displayed at an arbitrary position on the touch screen. The task of the participant is to find and name the object. In the evaluation (table 13.1 and figure 13.1), we focus on the time between the moment when the participant finds the object (which depends on where he starts and is, therefore, not very informative), and the moment when he names the object verbally to the experimenter. Chance level (pure guessing) is 25 % in this experiment. The results in table 13.1 state that the advanced color sonification approach is as appropriate as the one presented in part III is. Further, all participants reported that the advanced color sonification approach is more comfortable, intuitive and discriminable, especially in combination with the other sonifications.

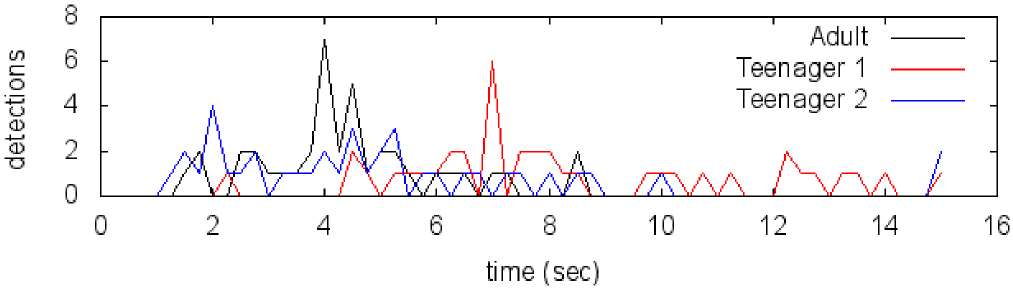


Figure 13.1: Histogram of experiment I. N elements (y axis) recognized in how many seconds (x axis) each

Participant	**Hitrate (% , N)**	**$\tilde{X}$**	μ	σ
Adult	97.5 % (39/40)	4.5 s	4.4 s	1.6 s
Teenager 1	97.5 % ,(39/40)	7.5 s	8.3 s	2.9 s
Teenager 2	97.5 % ,(39/40)	4.5 s	5.1 s	3.7 s

Table 13.1: Experiment I. Hit rates and times (median $\tilde{X}$, mean μ, and standard deviation σ in seconds), for each trial and participant

Experiment II - Finding Basic Scene Elements

The second experiment is about finding a set of scene elements named by the experimenter. Table 13.2 shows the elements to find and the cumulative times per participant per trial for images (stimulus 1 - 5 in figure 13.2). Only the sonification of color and "man made" regions are activated. We only sonify the existence of man made structures, i.e. no additional sonification for oriented lines and line gratings. Stereo panning is deactivated.

Img	Scene Elements to Find	Participant	Time
1	red building, sky, snow line on mountain top	Adult	12.1 s
		Teenager 1	19.0 s
		Teenager 2	25.0 s
2	building, green lawn, light blue sky, dark blue water	Adult	17.3 s
		Teenager 1	31.0 s
		Teenager 2	23.6 s
3	building, water, sky	Adult	10.6 s
		Teenager 1	21.0 s
		Teenager 2	20.2 s
4	buildings, lawn, trees, blue roof, white sky	Adult	45.0 s
		Teenager 1	12.5 s
		Teenager 2	10.7 s
5	dark red part of building	Adult	16.9 s
		Teenager 1	6.5 s
		Teenager 2	8.5 s

Table 13.2: Experiment II: Accumulated times for finding all announced scene elements per image

Experiment III - Understanding Scenes Audibly

This time, all Participants are given 2 minutes for each of the test images (stimulus 6 - 8 in figure 13.2) for exploration, without further information. Thereafter, they are to report what they have found in the image and what their interpretation of the scene is. Sonification is as in experiment II. A qualitative evaluation can be found in Table 13.3.

Img	Part.	Verbally Described Scene Estimation
6	Adult	*Lots of green parts, some small buildings within. At the top left is some kind of dark (uncolored) region, maybe belonging to sky or some sort of rock-structure.*
	Teen. 1	*There seems to be no sky visible in the image, but lots of green natural regions, into which a few small buildings are embedded.*
	Teen. 2	*There is a lot of green throughout the whole image, which is presumably a meadow or forest. Then there are some small buildings surrounded by meadows. Sky could not be found in any part of the image.*
7	Adult	*Green regions in the lower image part, probably some natural areas followed by a broad section of different colored building structures. In the mid-section of the image there is some red building block with that is surpassing the other building structures, presumably some sort of tower. The tower is surrounded by light blue and white, which might be the sky.*
	Teen. 1	*There is a meadow in the lower part of the image followed by a building or buildings of various colors. Those buildings are rather flat except for some sort of tower. The main upper part is covered in light blue, supposedly sky.*
	Teen. 2	*There is some sort of meadow in the lower image part and blue sky in the upper part. In between there is a different colored building section.*
8	Adult	*There is a small building on the mid-right, which is yellow. A bit to the left above the yellow building there is another building. Both buildings are surrounded by various colored non man made structures, which could be a meadow with various bushes or trees illuminated by the sun. On the top left there is a glimpse of light, maybe representing the sky*
	Teen. 1	*There is a yellow building. A green area beneath the building would presumably by some sort of meadow. The different colored spots surrounding the meadow and the building might be colored trees.*
	Teen. 2	*There is a yellow building and another more white one. Below the white building is a green area, presumably a meadow. There are yellow areas around and above the buildings, which could be trees.*

Table 13.3: Experiment III: Verbal descriptions of the scene estimates given by participants

Figure 13.2: Image set used in experiment II and III

Experiment IV - Understanding Scenes Audibly

Experiment IV is performed by the adult participant only. The setup was identical to Experiment III except that sonification of natural regions is additionally turned on. This time, the participant was given 10 images (stimulus 9 - 18 in figure 13.3) to explore. A qualitative evaluation can be found in Tables 13.4 and 13.5. The participant was able to detect and interpret all important scene content for 8 out of 10 images. With the other 20 percent, image 10 and 15, he only mistook the water for sky, which especially with picture 15 is hard to avoid.

Img	Verbally Described Scene Estimation
9	*There is a rather small, in parts yellowish, building in the midst of the top of an overgrown hill or meadow. The grown region is colored in green, with yellow and red stains. The upper part is light blue, supposedly sky.*
10	*A big block-like slightly red building in the mid-section of the image. Below that building is some green stripe, which might be a lawn. To the right the building seems to be embedded in some ascending rough natural green region with yellow elements. Could be some hilly, sun-illuminated lawn, or trees, reflecting sunlight.*
11	*The lower part of the image from left to right is some intensive green area. There is a strong contrast in roughness on the right from the smooth green area to a coarser green area in the mid-section. There is some light blue spot, which will be sky, on the top right corner and salient red building on the left.*
12	*There is a smaller band of light blue at the top across the image, supposedly sky. Then there are a few rather small buildings. The rest seams to be natural regions, which besides green and yellow include also some red elements. There is a dark blue spot in the lower left corner of the image, which will be some sort of water, such as a lake.*
13	*There are to buildings in the upper part of the image, one more to the left, the other more to the right. Both buildings are separated by a more white region. This white region also surrounds the upper parts of both buildings, so it is supposed to be sky. The left building has a slightly reddish roof. The whole lower part of the image is covered by some green-yellowish natural regions, such as lawn or forests.*
14	*There are to separate or a whole building complex at the center part of the image. The complex seems to be embedded in some sort of green-yellowish natural environment. The lower part is very dark and the the upper part of the image is covered from left to right by some light blue, which will be the sky.*

Table 13.4: Experiment IV: Verbal descriptions of the scene estimates given by participant

Img	**Verbally Described Scene Estimation**
15	*In the center of the image is some tower-alike building and a smaller one propagating to the right. The area below the building seems to be green natural environment. The tower is surrounded by blue and white of varying intensities, supposedly sky.*
16	*The lower part of the image is covered by some yellow-greenish area, supposedly meadows. From the left to the center within the mid-section there are some red buildings with blue roofs. Directly below these buildings there is some yellow band underlying such buildings from left to center. Right to the center building is some intensive green area, presumably a forest. The upper part, completely covered in light blue, should be sky.*
17	*The mid-section of the image is covered by some building complex. The building is partly yellow, and green on top. Below is a green and yellow region, probably lawn, and above and surrounding the building is blue and white, presumably sky.*
18	*The lower part of the image from left to right is smooth green, such as a lawn. Then there is a deep blue stripe which is supposedly some sort of water, such as a river. Above the river is a very flat band of buildings, followed by some green natural section. The top region is blue, presumably sky.*

Table 13.5: Experiment IV (continuation): Verbal descriptions of the scene estimates given by participant

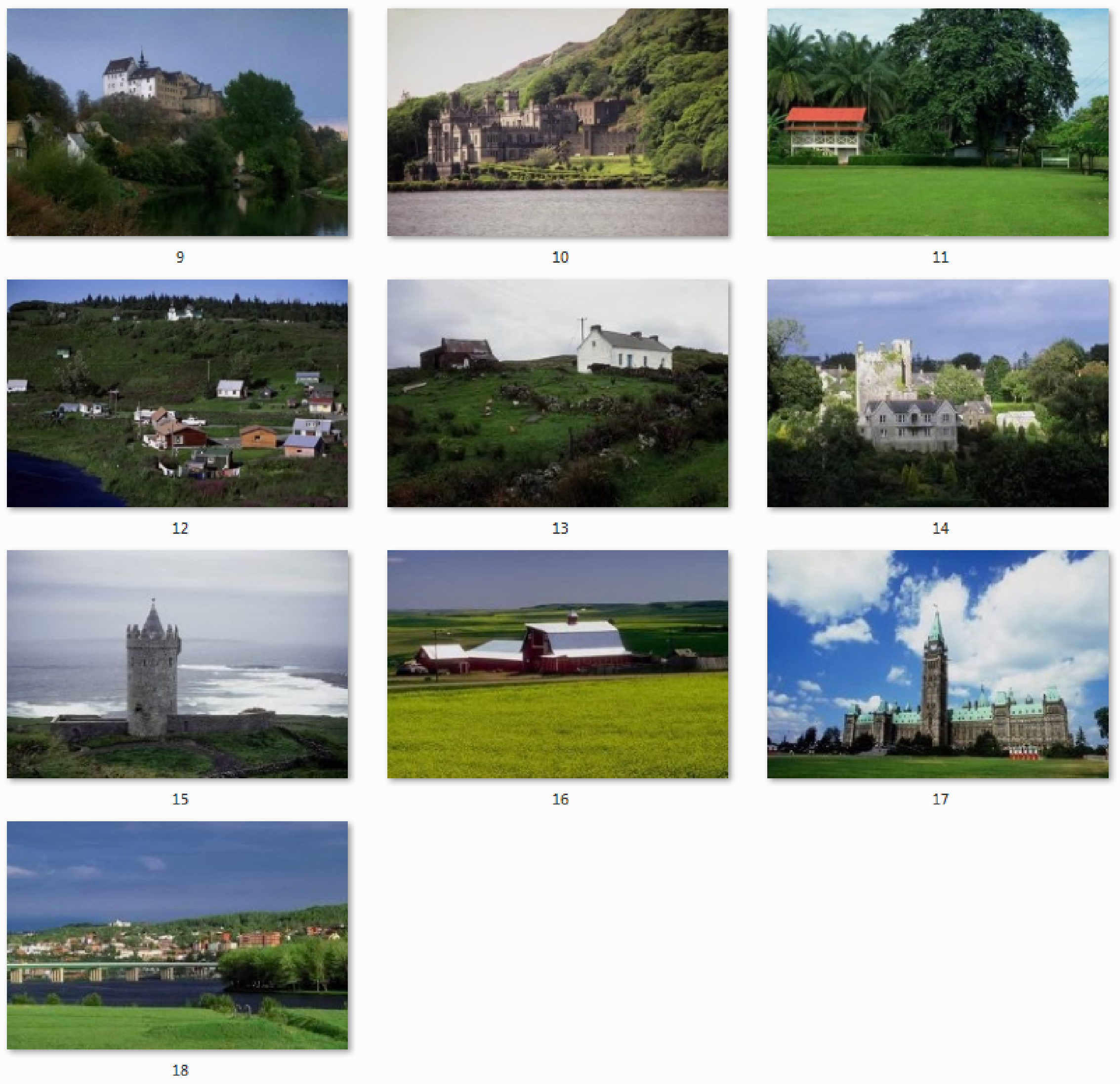


Figure 13.3: Image set for experiment IV

Experiment V - Categorization of Man Made Structures

Experiment V is performed by the adult participant only. All sonification modalities (color, roughness, man made structure (including oriented lines and line gratings),*Stereo Panning*) except object recognition are enabled. The participant is given 2.5 minutes for each image (stimulus 19 - 28 in figures 13.4) to explore the man made structures and to give an estimation what kind of building type it might be. A qualitative evaluation and comparison to what a seeing person might estimate is given in table 13.6. The participant was able to interpret the types of 7 out of 10 buildings correctly.

Img	Categ.	Verbally Described Type Categorization
19	Fortress or Church	*The flat compact building complex with tower in the lower right corner could be some sort of* ***fortress*** *.*
20	Temple or Church	*A bigger compact upper part on some sort of pillars or windows. The upper part has some sort of bevel or graded slope. Definitely a kind of* ***temple*** *or* ***gallery****.*
21	Light-house	*My first impression is a very small* ***tower*** *in the upper left corner. Might also be a small* ***cabin*** *on top of massive rock.*
22	Hotel	*Definitely a very big sort of manor. Many windows or pillars below the flat orange roof. Maybe some sort of* ***gallery*** *or* ***castle****.*
23	Fortress	*A building complex flat to the right, with a tower on the left. Could be a* ***church****.*
24	Tower or Church	*Seems to be a delicate bright* ***tower*** *opened to its right.*
25	Hotel	*A flat red building from left to right and equal in height. Above deep blue sky and below deep blue water. Could be some sort of* ***hotel*** *or* ***holiday resort****.*
26	Temple	*Small, very flat, bright and many windows. Maybe some sort of* ***bungalow****.*
27	Cabin	*Small, in the right corner. Blue water below and woods to the left. A* ***red cabin in the woods****.*
28	Light-house	*Seems to be the* ***lighthouse*** *again, on first impression. Could be a* ***cabin*** *also. On the left their is some sand-colored structure.*

Table 13.6: Experiment V: Comparison of visual and audible building type estimates

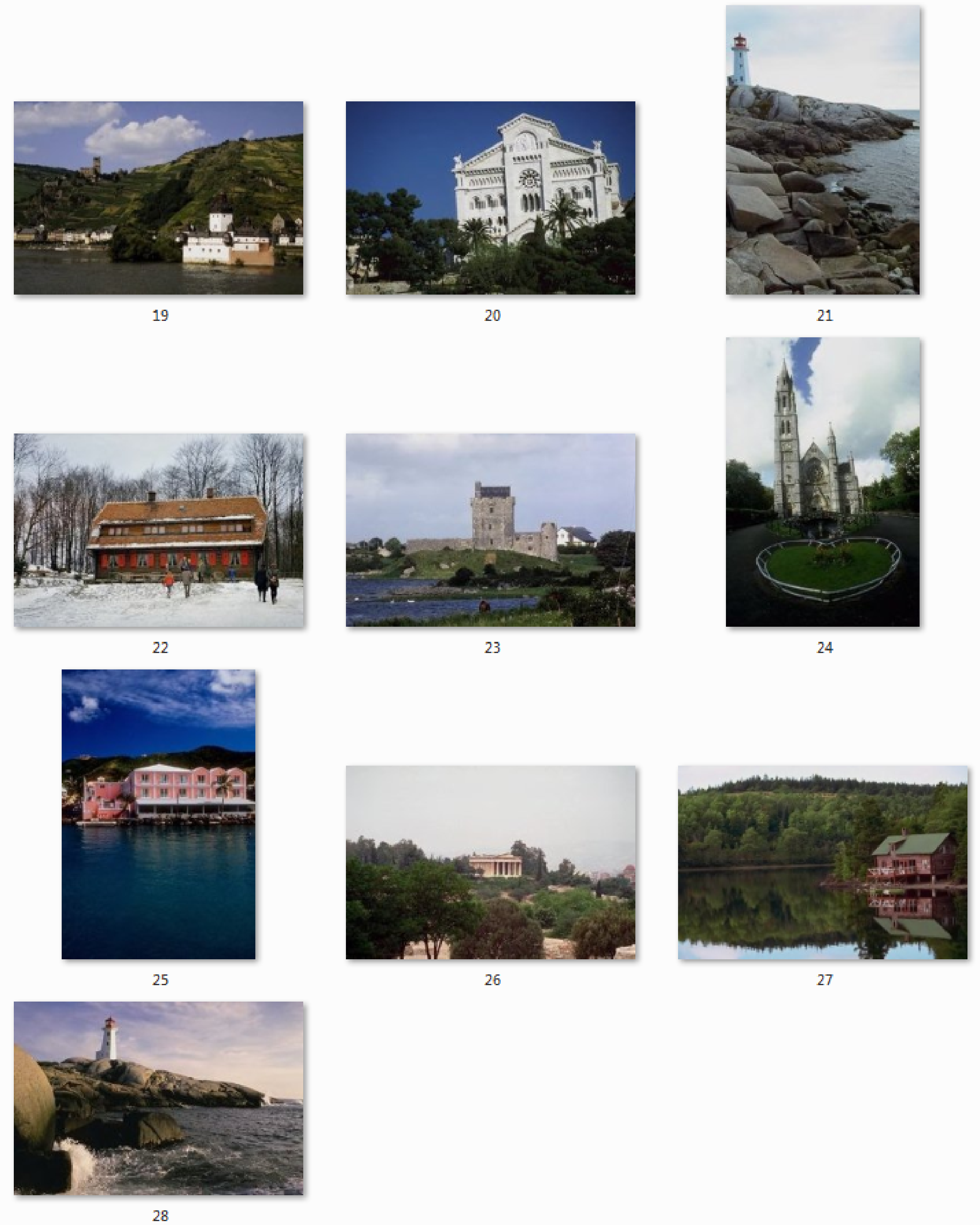


Figure 13.4: Image set for experiment V

13.2 Discussion

All participants appreciated the system to be very intuitive, easy to understand and quick to learn, and they enjoyed using it. The experimental results in experiment I indicate that the new "visual perception" based color sonification approach our system is at least as applicable for color recognition as the one presented in part III. However, participants stated it to be more intuitive and pleasant to work with. Furthermore, the experiments II - V indicate that the system raises the hope to be capable of giving visually impaired persons access to image content. Within a reasonable span of time, they were able to get an overview of "what is where" in the image, and to identify objects, given some context information about the scene. It is now a realistic application scenario that blind people can explore personal photos, perhaps together with a friend, and share memories about, e.g., their vacation. This is due to our paradigm of direct perception and interactive exploration using a very general tool. Expressed in the words of our adult participant:

> *What amazes me is that I start to develop some sort of a spatial imagination of the scene within my mind which really corresponds with what is shown in the image.*

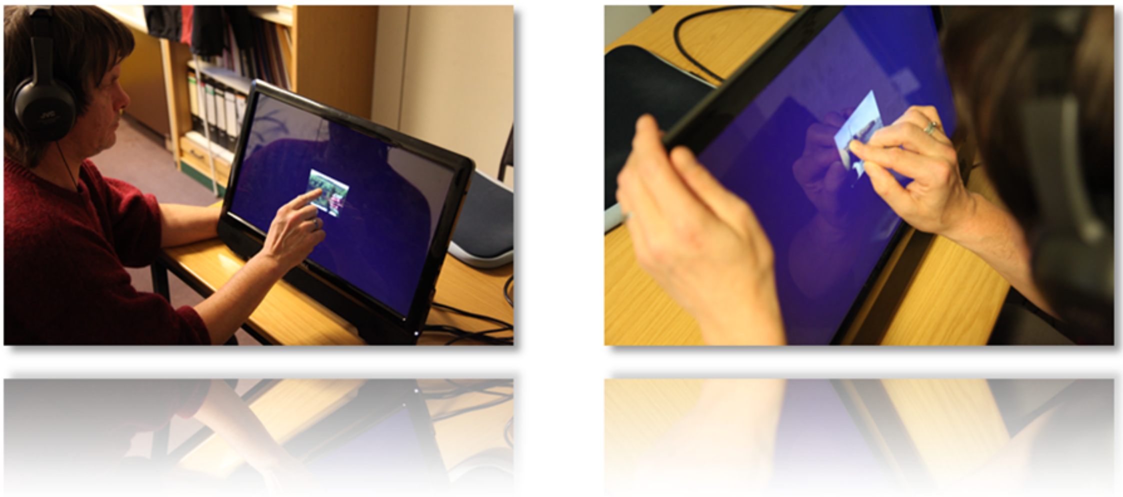


Figure 13.5: Our adult participant exploring a scene

Chapter 14

System Design of the Explorative Image Sonifyer Software

Finally, the work on “auditory image understanding” in this part IV of the work has been developed into a stand-alone application that is designed to run on *Microsoft* Windows 7 and 8 based tablet PCs. This chapter we give a brief overview over the system design of this application, called *Explorative Image Sonifyer (EIS)*.

14.1 Usability

Designed for visually impaired and especially blind users, the program can be controlled by only 3 touch-screen gestures, illustrated in figure 14.1:

- **Open Gesture**: Load and pre-process an image from a specific image folder.
- **Double Tab**: Load and pre-process an image from the tablet PC’s integrated camera. This feature makes the EIS software system ideally suited to be taken along by visually impaired people on, e.g., hiking tours.
- **Close Gesture**: Exit the program.

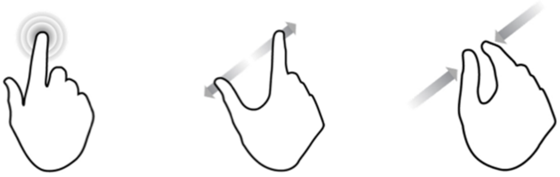


Figure 14.1: Left: Double tab (load image from camera). Middle: Open gesture (load image from folder). Right: Close gesture (close program)

The user is informed when an image is pre-processed and ready to be explored. At the same time, the image is rendered in the midst of the screen. Further all found man-made structures as well as detected objects are highlighted, as illustrated in figure 14.2. Although the blind user would not be able to utilize such visual information, it might be important in the context of sharing the acoustical image information with a normal sighted person. As visualized in figure 14.2, the "color to sound" mapping is permanently present from top to bottom at the left part of the screen. As the user moves across these regions a certain color or color-combination is played, while the name of the color is announced verbally. This feature was inspired by one of our congenital blind users to help memorize colors and to quickly compare what is heard in the scene to the color table.

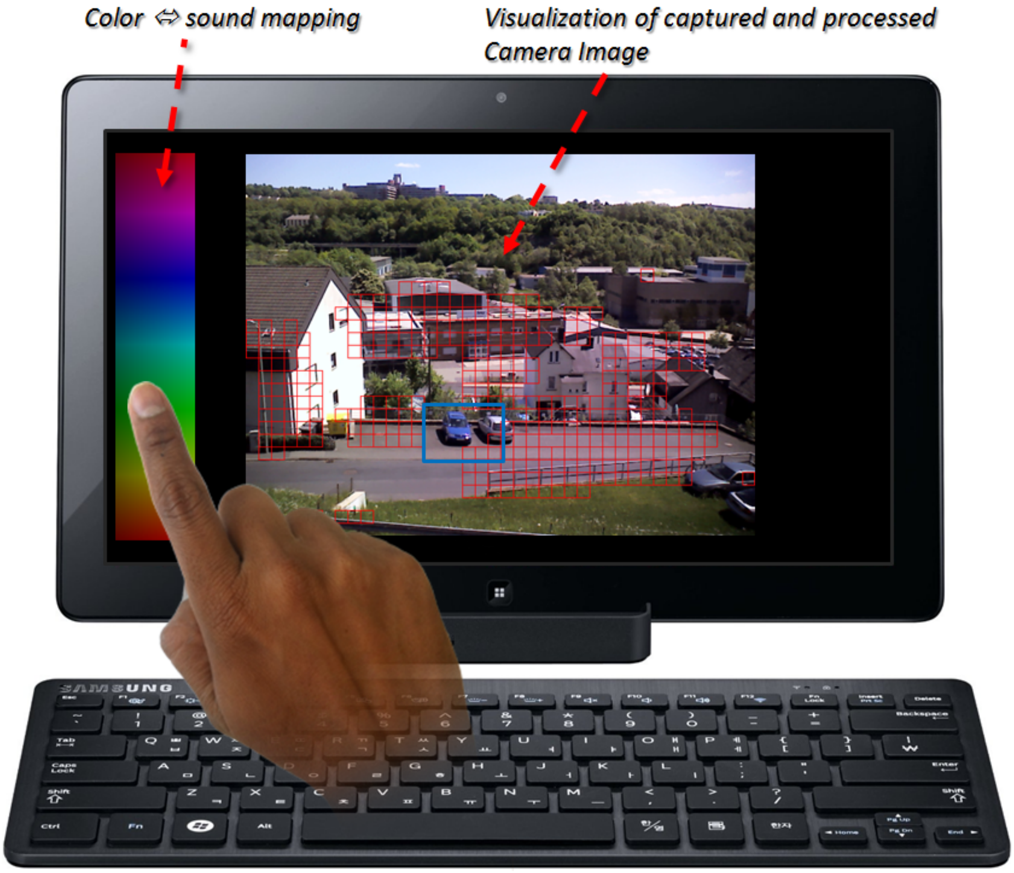


Figure 14.2: Illustration of a captured and pre-processed image on the Samsung Slate 7 tablet Pc. The color to sound mapping is visualized in the left part of the screen

14.2 System Architecture

The EIS software is internally designed as a **finite state machine (FSM)** ([288]; [72]). A finite state machine denotes a mathematical model of computation that can be used to design both logic circuits as well as computer programs. It is conceived as an abstract machine that can be in one of a finite number of states and the machine is in only one state at a time. A change of states can be triggered by an input event or condition, called a **transition**. However, transitions can be unconditional too. Thus, transitions map some state-event pairs to other states. Our FSM implementation, as illustrated in figure 14.3, contains 3 states:

- **Stand by**
- **Computation**
- **Exploration & Sonification**

While running, the program stays in stand-by state until the user accesses the screen using a double tab or open gesture. It then tries to load and pre-process (computation state) either the next image found in the image folder or from the integrated camera. Finally, it switches (unconditionally) to sonification state, allowing the user to explore the image interactively. A further double tab or open gesture causes the program to switch back to computation state and prepare the next image from the folder or camera, respectively.

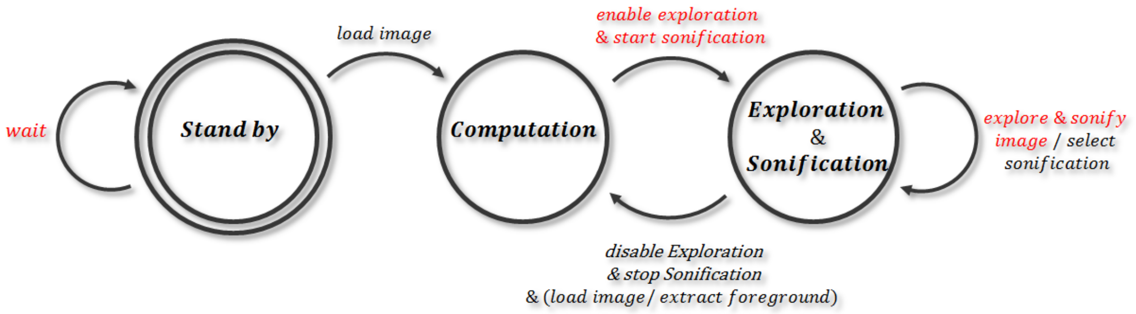


Figure 14.3: Our system internally realized as a finite state machine. Unconditional transitions are illustrated in red

Note that although not enabled within the current version of the framework, hitting an additional button in sonification state would cause the program to jump back to computation state and perform a foreground segmentation at the user's current pixel position. In section 11.4 we referred to such a button as the external "buzzer" like button.

14.3 Text-to-Speech Output

As mentioned in section 4.1 the user is kept up to date about the program's current status using *text-to-speech* output. We, therefore, make use the **Microsoft Speech API (SAPI)** [402]. To enhance diversity, in a separate file, we define a number of varying expressions describing the same process within the program and a random generator that selects any of those to communicate to the user. As an example, as the user hits a button to load an image, the program could randomly choose between, e.g., *Ok, I will load the next image now!* or *I try to load an image now!*.

14.4 Non-blocking Realtime Audio Programming

Parallel real-time exploration and sonification in general is crucial to allow the system to continuously write and process variables at the same time. Other than in part III, where such real-time parallelism is handled by the external MIDI synthesizer, it has to be implemented from scratch in part IV and, therefore, an essential component of the EIS software. However, the implementation of a stand-alone sonification system that is capable of gathering data, due to explorative interaction, while simultaneously turning such data into sound in real-time is a pretty challenging task. As Bencina [28] writes in his technical report on real time audio processing:

> *Writing real-time audio software for general purpose operating systems requires adherence to principles that may not be obvious if you are used to writing "normal" non real-time code. Some of these principles apply to all real-time programming, while others are specific to getting stable real-time audio behavior on systems that are not specifically designed or configured for real time operation ... you do not want your softwares audio to glitch and real time waits for nothing.*

As discussed in section 4.1 the appropriate elements from the augmented visual pixels $\boldsymbol{v}(x, y)$ have to be evaluated by the sonification module in real-time, during interactive exploration, based on the user's current position (x, y) and his selection which features he wants to be sonified. Subsequentially, adjustments in the play-back of features sounds as well as rather complex alterations in the color sound synthesis have to be performed at the same time while new data is gathered, due to interactive exploration. Hence, exploration and sonification have to run in separate threads and share mutual data and this is where the usage of an intermediate queue structure, as proposed in section 5.4, that contains the necessary control parameters really pays off.

Generally, when working with threads that share data, they have to be prevented from one thread writing data while another one wants to read it. Hence, mutual exclusion techniques, known as **mutexes** ([449]; [33]) are employed to prevent that from happening. Thus, the thread writes something into the shared data firstly **locks** a mutex that encapsulates the access to this data. When finished, he releases the mutex and the second thread is free to read it, himself locking the mutex.

While this technique works well under non-real-time purposes it has a significant drawback when applied to real-time applications. While the mutex is locked, the second thread is stalled, which leads to a delay in the further processing. Hence, the concept has been extended to use a **trylock** approach. A thread attempts to acquire the mutex. If it is available, the call returns with the mutex locked and returns true and the thread can access the data. Otherwise, if the mutex is locked by another thread, the call returns false and the thread can process with any other operation.

However, in our application the first thread is assigned to the exploration module and the second to the sonification module. Both threads now try to access the same shared data. The exploration module tries to select the appropriate information for passing it to the sonification module, while this module tries to constantly transform this data into sounds and, therefore, both modules can not afford to wait upon each other. Other than that, both modules process data at different speeds, meaning the transformation of data into sounds occurs at a different speed than the gathering of data through the exploration module.

Hence, the queue structure in between, in combination with mutexes, can be harnessed as a non-blocking buffer. The whole queue contains the shared data. While the exploration module appends sonification descriptors $\boldsymbol{s}(x, y)$ at the end of the queue, the sonification module sequentially process the first one. Trylock techniques are employed to prevent especially the sonification module from waiting upon the exploration module to unlock the queue. In case the queue is locked when the sonification module wants to access it, it simply continues to sonify the current given data and tries again after a specific period of time.

Threading and mutual-exclusion is implemented based on the QT4 library ([33]; [136]). While the timing of the exploration module to access the queue (via tryLock) is left to *QT4*'s general interface policies, the sonification module is timed to try to access the queue every ≈ 50 milliseconds.

14.5 Portability

The EIS software is designed to be easily ported to several other systems and platforms, using few and platform independent libraries, such as QT4 for threading, exploration interfaces and the (minimal) graphical user interface. Further we employ the irrklang audio engine and the Synthesis Toolkit (STK),which is only a set of C/C++ classes for sound generation. The rest of the project would be a set of object oriented C++ classes only.

Part V

Conclusion

If the brain were so simple we could understand it,
we would be so simple we couldn't .
Lyall Watson

The modular computer vision sonification model presented in part II with the two sample implementations presented in part III and IV, propose a promising general framework of some device that can support blind and visually impaired persons in exploring images or scenes. All of our congenital participants appreciated the system to be very intuitive, easy to understand and quick to learn, and they enjoyed using it. The experimental results indicate that our system is successful in giving visually impaired persons access to image content: Within a reasonable span of time, they were able to perform both, "auditory object recognition" and "auditory image understanding". Thus, they able to obtain an overview of "what is where in the image", and to identify objects, given some context information about the scene. As mentioned in chapter 13, it is now a realistic application scenario that blind people can explore personal photos, perhaps together with a friend, and share memories about, say, their vacation. This is due to our paradigm of direct perception and interactive exploration using a very general tool. In contrast, many everyday tasks, such as navigation, are more likely to be the domain of special-purpose tools and a faster, more automated procedure to derive specific relevant information.

Our experiments give hope that the proposed color sonification approach that proves to be intuitive enough to be understood and applied by 4 congenital blind people of different backgrounds in very little time will prove also successful on a larger group of congenital blind people. As a fascinating insight, within the experiments on "auditory image understanding" in chapter 13, participants are "incorporated " in the image understanding process. Although pre-classification is only into natural and man made regions, during exploration, participants utilize detected man made structures or specific natural regions as reference points to classify other natural regions by their individual location, color and texture.

For image pre-evaluation, we have presented two novel algorithms, especially for the visually impaired. Robust scene classification is performed based on a novel type of Conditional Random Fields, called Dual Support Vector Fields (DSVF), that harness the high discriminative power of non-linear support vector machines for both, unary and pairwise potentials. As shown in section 11.2, DSVF, in combination with an advanced feature set, provide a valuable alternative to existing models for man made structure detection. DSVF crucially reduce parameter learning, in both time and complexity, and are, therefore, highly suitable, given an arbitrary feature set, for "rapid prototyping" of classification problems with spatial dependencies.

A second algorithm has been proposed in section 11.4 as a subsequent step of object recognition to verify or falsify results, which becomes significantly important in the context of image evaluation for the visually impaired. Great benefit of the proposed algorithm is that both, the algorithm itself as well as the integrated feature set can be applied to the results of any common recognition algorithm.

Due to their design, both algorithms can be also employed in other applications than “auditory image understanding”, e.g., for fully-automated computer vision systems. Integrated in the image sonification implementation in part IV, these approaches deliver a complete powerful system that helps visually impaired users to explore image material.

Furthermore, it is also important to us to introduce the problem setting of employing algorithms to provide an exploratory visual substitution system, rather than a complete verbal description, to the computer vision community, as it sheds new light on the understanding of vision in general in terms of what might be the “intermediate description level” below a complete semantic image description, or what features, categories and mechanisms need to be integrated for scene understanding, both in computer vision and in the human visual system.

For further research, it might be interesting to extend the computation and sonification module in part IV to incorporate depth information, as there have been recent interesting developments to create a depth perception from single still images ([292]; [412]; [411]; [467]).

As described in section 14.5, the system proposed in part IV is designed to be easily ported to several other systems and platforms. Therefore, it would be interesting not only to port the system but also provide some sort of “training” mode, that blind users can obtain along with the system, that replaces a sighted person to introduce the program on a larger scale.

Incorporating the System at the "Internat des Rheinischen Blindenfürsorgeverein 1886 Düren"

In all our experiments, the group of 3 congenital blind 14 year old teenagers are residents at a residential school for the visually impaired, the Internat des Rheinischen Blindenfürsorgeverein 1886 Düren. This residential school offers accommodations for about 84 students, who live in 10 family-like residential-groups. In close cooperation with parents and the Louis-Braille-school for the visually impaired, the residential school contributes to nurture all students. Therefore, it provides various additional educational workshops, that are incorporated in the student's daily routines. These workshops deal with various areas, such as gross and fine motor skills, orientation, mobility, perception, communication, social skills or recreation. As mentioned, in January 2013, preparations commenced to incorporate our system permanently within some of these workshops as a method to support students in some of these areas on a regular basis.

Figure 14.4: The residential school for the visually impaired in Dueren, Germany (Internat des Rheinischen Blindenfürsorgeverein 1886 Düren)

Part VI

Appendix

Appendix A

Mathematical & Algorithmic Concepts

A.1 Edge Preserving Filtering

Anisotropic Diffusion

Anisotropic diffusion ([353]; [32]; [125]) is inspired by interpreting Gaussian smoothing as a heat conduction partial differential equation (PDE): $\frac{\partial I}{\partial t} = -\triangle I$. Thus, the intensity I of each pixel denotes heat that is propagated to its neighbors according to the heat spatial variation. An "edge-stopping" function g has been introduced by Perona and Malik [353] that varies the "conductance" according to the image gradient and therefore prevents heat flow across edges:

$$\frac{\partial I}{\partial t} = div[g(\|\nabla I\|)\nabla I]$$

Perona and Malik [353] propose two expressions for such an edge-stopping function g:

$$g_1(x) = \frac{1}{1+\frac{x^2}{\sigma^2}} \qquad\qquad g_2(x) = e^{-\frac{x^2}{\sigma^2}}$$

where a scale parameter σ in the intensity domain specifies the gradient intensity which should stop diffusion. The discrete Perona-Malik diffusion equation governing the value I_i at pixel i would then be

$$I_i^{t+1} = I_i^t + \frac{\lambda}{4} \sum_{j \in N_4(i)} g(I_j^t - I_i^t)(I_j^t - I_i^t)$$

Discrete time steps are described based on t, and $N_4(i)$ denotes the first order neighborhood of pixel i. One further scalar λ determines the rate of diffusion. Generally, anisotropic diffusion is a rather slow process due to its discrete diffusion nature and the results depend on the stopping time, since the diffusion converges to a uniform image.

Bilateral Filtering

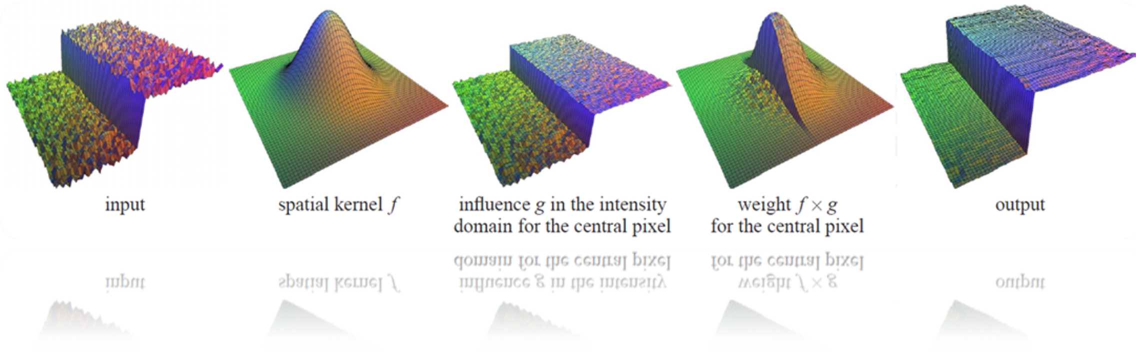

input
spatial kernel f
influence g in the intensity domain for the central pixel
weight f × g for the central pixel
output

Figure A.1: Bilateral filtering. Colors are used only to convey shape. Picture taken from [125]

Bilateral filtering has been developed by Tomasi and Manduchi as an alternative to anisotropic diffusion ([464]; [21]; [125]). It describes a non-linear filter with the output being a weighted average of the input. Bilateral filtering starts with Gaussian smoothing based on a spatial kernel f (see figure A.1). Further, the weight of a pixel additionally depends on a function g in the intensity domain that decreases the weight of pixels with large intensity differences. g can be seen as an edge-stopping function similar to that of Perona and Malik [353]. The final result B_i of the bilateral filter for a pixel i would then be:

$$B_i = \frac{1}{k(i)} \sum_{j \in \Omega} f(j-i)\, g(I_j - I_i)\, I_j$$

with k(s) being a normalization term:

$$k(i) = \sum_{j \in \Omega} f(j-i)\, g(I_j - I_i)$$

In practice, a Gaussian for f in the spatial domain as well as a Gaussian for g within the intensity domain are used. Therefore, each pixel value I_i is influenced mainly by spatially close pixels that have similar intensities, as illustrated in figure A.1. Thus, bilateral filtering can easily be extended to color images, and any metric g on pixels can be used, such as in CIELab Color Space.

On the Distortion of Colors

For gray-scale images, intensities between gray levels are still gray levels. Thus, Gaussian smoothing on gray-scale images produces intermediate levels of gray across edges, resulting in blurred images. Smoothing color images is therefore more complicated, as between any two colors there are, mostly, rather different colors. Disturbing color bands can arise when smoothing across color edges, resulting in a smoothed image which does not only look blurred, but also contains strangely colored auras around objects.

To compensate for, edge-preserving smoothing could be applied to the red, green, and blue channels of an image separately, altough intensity profiles in the three color bands are generally different across an edge, rendering such an approach infeasible. In contrast, bilateral filtering allows to combine the three color bands appropriately, and to measure photo-metric distances between pixels in such a combined space. using Using a Euclidean distance measure in the CIELab color space [286], such a combined distance closely corresponds to perceived dissimilarity. Therefore, as Tomasi and Manduchi [464] emphasize:

> *In a sense, bilateral filtering in the CIElab color space would describe the most natural type of filtering for color images, as only perceptually similar colors are averaged together, and only perceptually important edges are preserved.*

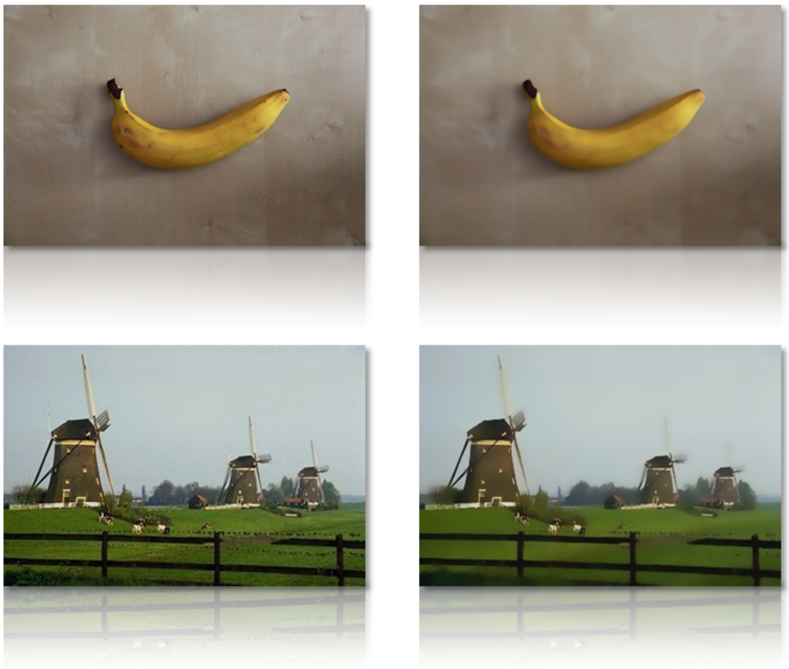


Figure A.2: Bilateral filtering example. Left: Original images, Right: Bilateral filtered images

A.2 Fractal Geometry & Fractal Dimension

The idea of "fractured" dimensions has a long history in mathematics ([126]; [140]) but the term itself was first introduced by Benoit Mandelbrot based on his famous paper on self-similarity, in which he discusses **fractional dimensions (FD)** [302]. In [302], Mandelbrot cited previous work by Lewis Fry Richardson describing the counter-intuitive idea that a coastline's measured length changes with the length of the measure used [385].
Therefore, the fractal dimension of a coastline quantifies how the number of scaled measures, such as boxes, necessary to evaluate the coastline, changes with different box scales [303], illustrated in figure A.3.

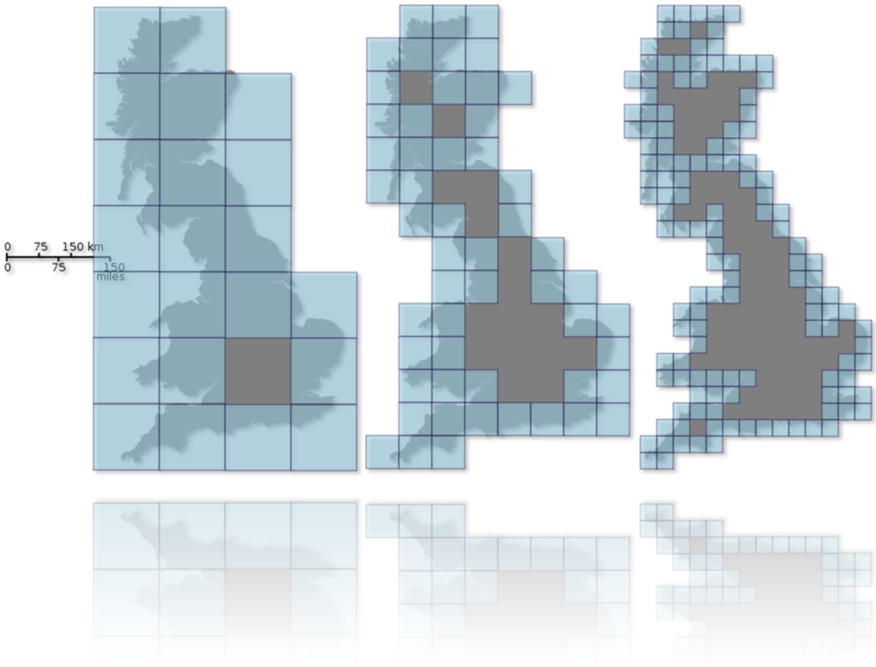


Figure A.3: Estimating the box-counting dimension of the coast of Great Britain

There are various formal mathematical definitions of fractal dimension that build on this concept of change in detail with change in scale, one which would be the **Hausdorff dimension** ([115]; [459]), also known as **box-counting dimension**:

$$FD = \lim_{r \to 0} \frac{\log N_r}{\log(1/r)}$$

with N_r being the least number of boxes of side length r needed to cover the entire structure under examination.

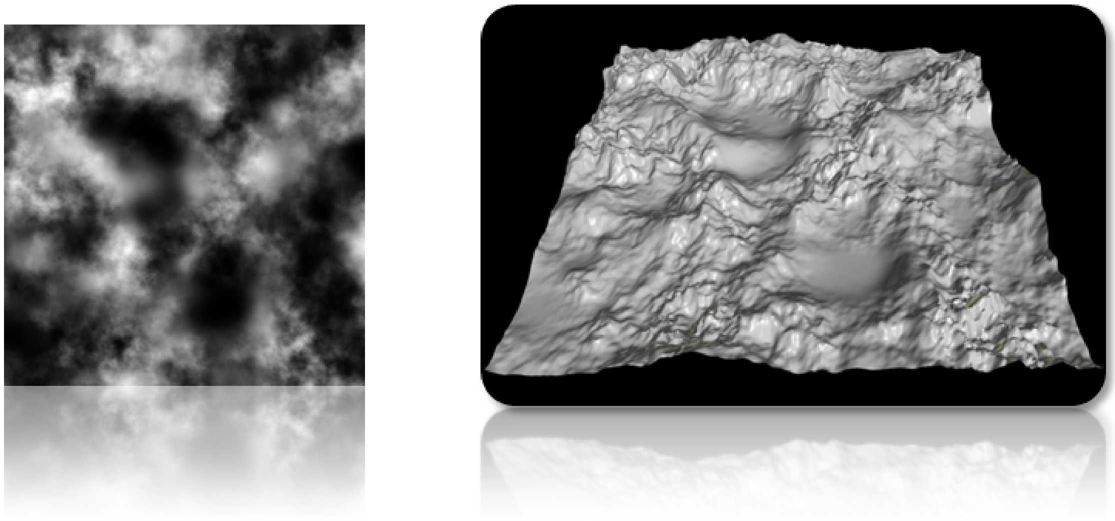


Figure A.4: Left: A 2D gray-scale texture. Right: The corresponding 3D representation

The box-counting method or its variation, the **differential box-counting (DBC)**, can easily extended to evaluate 3D structures or surfaces [410]. Pentland [351] showed, that the fractal dimension of a surface corresponds quite closely to our intuitive notion or roughness.

The DBC method represents a gray level 2D image $\boldsymbol{I}_{2D}$, see figure A.4 (left), as a 2D surface within a euclidean 3D space with the the gray level denoting a z along with each (x, y) pixel position within the image, see figure A.4 (right), and estimates the fractal dimension by dividing the number of boxes, needed to cover the overall 3D area, by their diameter. However, due to the a series of problems, outlined in [289], the accuracy of the original DBC method is limited. Li et al. [289] present three main modifications in their box-counting estimation method to compensate for these problems, which is why we implement their method to compute the fractal dimension of image sites in chapter 11.3. Generally, fractal dimension approaches are employed in Computer Vision in segmentation and classification problems ([484]; [242]; [74]; [214]; [76]).

A.3 Gabor Wavelet Transform

Gabor wavelets have been introduced to image analysis due to their biological relevance and computational properties ([230]; [99]). They have been widely adopted for feature extraction in various areas, such as face recognition ([535]; [534]; [282]), texture segmentation ([123]; [124]; [223]) or iris recognition ([99]; [100]). A Gabor function, first proposed by Dennis Gabor in 1946 [156], is a 2D Gaussian function multiplied by a 2D harmonic function. In [156], Gabor showed that a sort of "quantum principle" exists for information as well. Hence, it is crucial for the conjoint time-frequency domain for 1D signals to be quantized so that no signal or filter occupies less than a minimum area in it. Similar to Heisenbergs uncertainty principle ([244]; [301]), such a minimal area, reflecting the inevitable trade-off between time resolution and frequency resolution, has a lower bound in their product, and, according to Gabor's discoveries, Gaussian-modulated complex exponentials provide the best trade-off. Generally, a harmonic function is a Fourier basis function. Especially, in a 2D Gabor kernel it is a sinusoidally modulated function, in a form of complex exponential function. Elements of a family mutually similar Gabor functions are called **Gabor wavelets** if they are created by dilation and shift from a single elementary Gabor function. The Gaussian function varies in dilation and the harmonic function varies in rotation and frequency ([535]; [534]; [282]). Gabor wavelets capture local structure corresponding to spatial frequency, i.e., scale, spatial localisation (coordinates), and orientation selectivity. A Gabor wavelet ([100]; [123]) is defined as

$$\psi_{\varphi,\nu}(z) = g_{\varphi,\nu,\sigma}(z)\Big[e^{i\,k_{\varphi,\nu}\,z} - e^{-\frac{\sigma^2}{2}}\Big] \tag{A.1}$$

with the Gaussian envelope:

$$g_{\varphi,\nu,\sigma}(z) = \frac{||k_{\varphi,\nu}||^2}{\sigma^2}\, e^{-\frac{||k_{\varphi,\nu}||^2\,||z||^2}{2\,\sigma^2}} \tag{A.2}$$

where $z = (x, y)$ indicates a point with x, the horizontal coordinate and y, the vertical coordinate. The parameters φ and ν define the angular orientation and the spatial frequency of the Gabor kernel. The spatial frequency in equation (A.1) modulates the size of the 2D discrete Gabor kernel. Thus, ν also determines the scale of kernel. $||.||$ denotes the norm operator. The parameter σ is the standard deviation of Gaussian window in the kernel.

The Gaussian window shape in the 2D Gabor function provides the best time-frequency localization window in a sense of the Heisenberg uncertainty principle [301]. The wave vector $k_{\varphi,\nu}$ is defined as

$$k_{\varphi,\nu} = k_\nu \, e^{i\,\phi_\varphi} \tag{A.3}$$

with $k_\nu = \frac{k_{max}}{f^\nu}$ and $\phi_\varphi = \frac{\pi\varphi}{n}$. n denotes the number of different orientations chosen, k_{max} the maximal frequency, and f denotes a spatial factor between kernels within the frequency domain.

Since Gabor kernels in equation (A.1) are generated from one kernel by dilation and rotation via the wave vector $k_{\varphi,\nu}$ they are all self-similar. Each kernel is a product of a **Gaussian envelope** $g_{\varphi,\nu,\sigma}(z)$ formulated in equation (A.2) and a complex plane wave $e^{i\,k_{\varphi,\nu}\,z}$. The complex wave determines the oscillatory part of the kernel. The term $-e^{-\frac{\sigma^2}{2}}$ is defined to compensate for the Disparity Compensated (DC) value that makes the kernel **DC-free** [260]. In other words, the term ensures the wavelets do not lose any generality, i.e., there is no minimal energy loss when images are reconstructed by the wavelets. Thus, DC-free wavelet representations do compensate for global illumination changes. In case the parameter σ that determines the ratio of the Gaussian window width to wavelength, is sufficiently large, the effect of the DC term becomes negligible.

Generally, five different scales and eight orientations of Gabor wavelets are used, e.g., for face recognition in [535], with $\nu \in \{1, ..., 3\}$ and $\varphi \in \{0, ..., 7\}$. In our application we only use a single scale, depending in size upon the specific task in part III and IV, and 32 orientations. We choose the maximum frequency to be $k_{max} = \pi/2$, and the factor $f = \sqrt{2}$. Gabor wavelets are modulated by a Gaussian envelope function with relative width $\sigma = 2\pi$. These parameters are chosen according to previous findings ([510]; [535]; [293]). The kernels contain useful characteristics of spatial frequency, orientation selectivity or spatial locality.

The Gabor kernel is defined as the product of a Gaussian and a complex plane wave with real, called “even”, and imaginary parts, called “odd”. The equation (A.1) can, thus, be separated into real part

$$\frac{||k_\nu||^2}{\sigma^2}\, e^{-\frac{||k_\nu||^2\,||z||^2}{2\,\sigma^2}} \left\{ \cos(k_\nu cos(\phi_\varphi)x + k_\nu \sin(\phi_\varphi)y) - e^{-\frac{\sigma^2}{2}} \right\} \tag{A.4}$$

and the imaginary part

$$\frac{||k_\nu||^2}{\sigma^2}\, e^{-\frac{||k_\nu||^2\,||z||^2}{2\,\sigma^2}} \sin(k_\nu cos(\phi_\varphi)x + k_\nu \sin(\phi_\varphi)y) \tag{A.5}$$

Figure A.5 illustrates the real and imaginary parts of an ensemble of Gabor wavelets of various scales and orientations. A discrete ensemble of the 1.5 octave bandwidth family

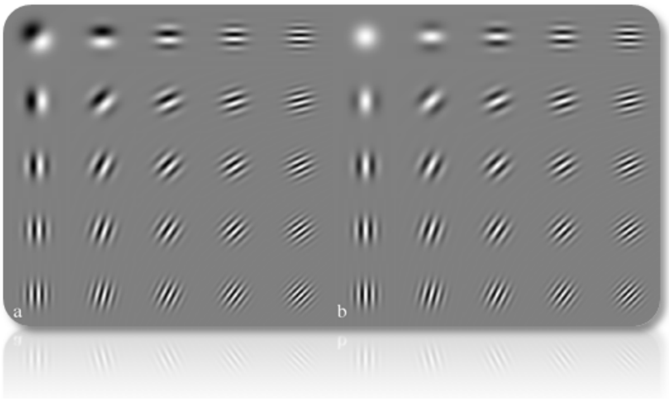


Figure A.5: An ensemble of odd (a) and even (b) Gabor filters. Picture taken from [282]

of Gabor wavelets along with their coverage of the spatial frequency plane is illustrated in figure A.6.

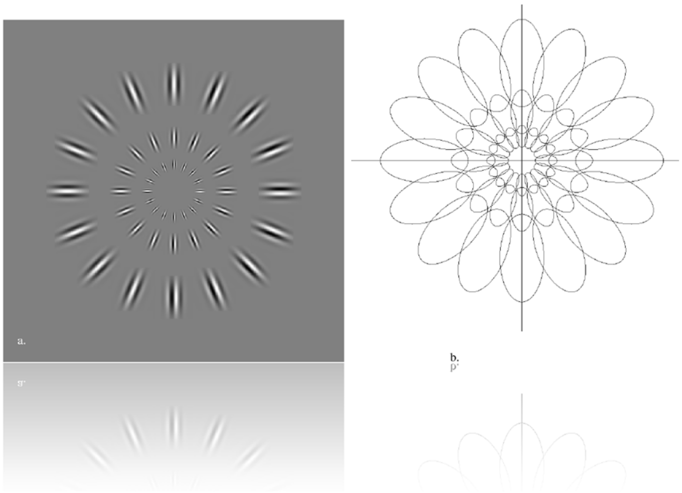


Figure A.6: An ensemble of Gabor wavelets (1.5 octave bandwidth) (a) and their coverage of the spatial frequency plane (b). Each ellipse shows the half-amplitude bandwidth contour dilated by a factor of 2, covering almost the complete support of a wavelet. Picture taken from [282]

Gabor Responses

The Gabor wavelet transform of an image can be performed by convolving all Gabor wavelets with the image. This convolution of an image $\boldsymbol{I}$ and a Gabor kernel $\psi_{\varphi,\nu}$ is defined as

$$\boldsymbol{O}_{\varphi,\nu}(z) = \boldsymbol{I}(z) * \psi_{\varphi,\nu}(z)$$

where $*$ denotes the convolution operator, and $\boldsymbol{O}_{\varphi,\nu}(z)$ is the convolution result corresponding to the Gabor kernel at the orientation φ and the spatial frequency ν. $z = (x, y)$ defines the position in the image. Since Gabor wavelets are of complex form, convolution results contain a real response and imaginary response as follow

$$\boldsymbol{O}_{\varphi,\nu}(z) = \Re\{\boldsymbol{O}_{\varphi,\nu}(z)\} + i\,\Im\{\boldsymbol{O}_{\varphi,\nu}(z)\}$$

where $\Re$ represents the real response and $\Im$ represents the imaginary response. The real response of Gabor filtering is an image $\boldsymbol{I}(z)$ convolved with the real part of the Gabor kernel in (A.4) . The real response of Gabor filtering is than defined as

$$\Re\{\boldsymbol{O}_{\varphi,\nu}(z)\} = \boldsymbol{I}(z) * \Re\{\psi_{\varphi,\nu}\}$$

The imaginary response is the image convolved with the imaginary part of the Gabor kernel in (A.5). It is stated as

$$\Im\{\boldsymbol{O}_{\varphi,\nu}(z)\} = \boldsymbol{I}(z) * \Im\{\psi_{\varphi,\nu}\}$$

The magnitude response of Gabor filtering, widely employed in Computer Vision and as well in our application in part III and IV, is the square root of the sum of the squared real response and imaginary response, such as

$$||\boldsymbol{O}_{\varphi,\nu}(z)|| = \sqrt{\Re^2\{\boldsymbol{O}_{\varphi,\nu}(z)\} + \Im^2\{\boldsymbol{O}_{\varphi,\nu}(z)\}}$$

As discussed in [22], an even Gabor function, a cosine function, can be seen as a partial differential operator of an even order, see figure A.7 (right), while an odd Gabor function, a sine function, can be looked at as a partial differential operator of an odd order, see figure A.7 (left). To compute the even and odd derivatives together with a single complex Gabor wavelet, one determines the even derivative from a real part of the function $\Re$ and the odd derivative from an imaginary part of the function $\Im$.

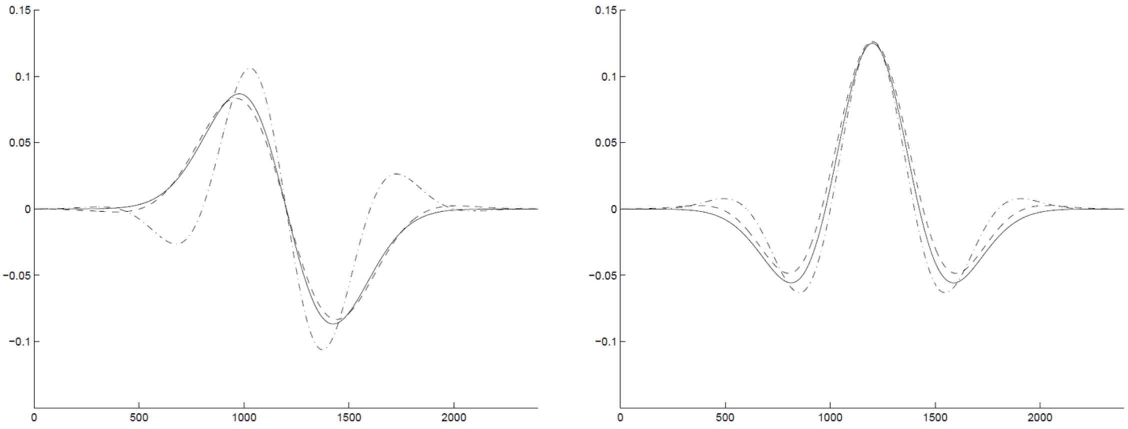


Figure A.7: Left: Graphs of the odd Gabor functions with different frequencies (dash-dot, dashed) and the first derivative of the Gaussian function (solid). Right: The even Gabor functions with different frequencies (dash-dot, dashed) and the second derivative of the Gaussian function (solid). Picture taken from [22]

A.4 Gaussian Image Pyramids

Gaussian Image Pyramids ([222]; [64]; [12]) serve as a multi-scale representation of an input image for image processing (illustrated in A.8). It consists of a set of low-pass copies of the input image, each representing pattern information of a different scale. Gaussian pyramids have been applied in various areas, such as image data compression [64] or mosaic-ing [63] just to name a few. In the implementation in section 7.1 (part III) we deal with rather coarse structures. Hence, we apply sub-scaling of the original input image using Gaussian pyramids to perform filtering on a lower resoluted copy. Thus, we can efficiently reduce computation time. Subsampling with Gaussian pyramids is not just e.g. taking every second pixel in every second line as such a procedure would violate the **Nyquist-Shannon Sampling Theorem** ([428]; [226]; [298]). Given a structure that is sampled three times per wavelength within the original image would only be sampled one and a half times in the sub-sampled image and therefore appear as an aliased pattern. Hence, one must ensure that all structures that are sampled less than four times per wavelength would be suppressed by an appropriate smoothing filter to ensure a proper subsampled image. The combined smoothing and size reduction can be expressed in a single operator using the following notation to compute the $q + 1^{th}$ level of the Gaussian pyramid from the q^{th} level:

$$\boldsymbol{G_0} = \boldsymbol{G}, \;\; \boldsymbol{G_{q+1}} = \boldsymbol{B_{\downarrow 2} G_q}$$

The number behind the $\downarrow$ in the index denotes the subsampling rate. The 0^{th} level of the pyramid is the original image. If we repeat the smoothing and sub-sampling operations iteratively, we obtain a series of images, which would be the actual Gaussian pyramid. From level to level, the resolution and size decreases by a factor of four. Consequently, one gets series of images forming the shape of a pyramid as illustrated in figure A.8. However, in section 7.1, after several experiments to find the most appropriate low-level image representation in terms of computational speed and edge detection quality, we finally selected the first down-scaled image $\boldsymbol{G_1}$.

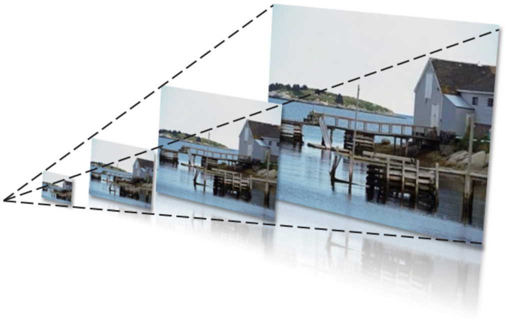


Figure A.8: From level to level, the resolution and size of the input image decreases by a factor of four. The result is series of images forming the shape of a pyramid

A.5 Algorithms in Graph Theory

Dijkstra's Single-Source Shortest-Paths Algorithm

Given a weighted graph $G_o = (V, E)$ ([73]; [473]; [85]), Dijkstra's algorithm, as used in section 11.4 (part IV), finds the connections of minimum costs from a specific source vertex $s \in V$ to each other vertex $v \in V$, known as the "Single-Source Shortest-Paths problem" ([315]; [422]; [85]). It keeps a set S of vertices whose final shortest-path weights from the source s have already been determined. The algorithm repeatedly selects the vertex $u \in V - S$ with the minimum shortest-path estimate, adds u to S, and scrutinizes all edges leaving u, as illustrated in A.9.

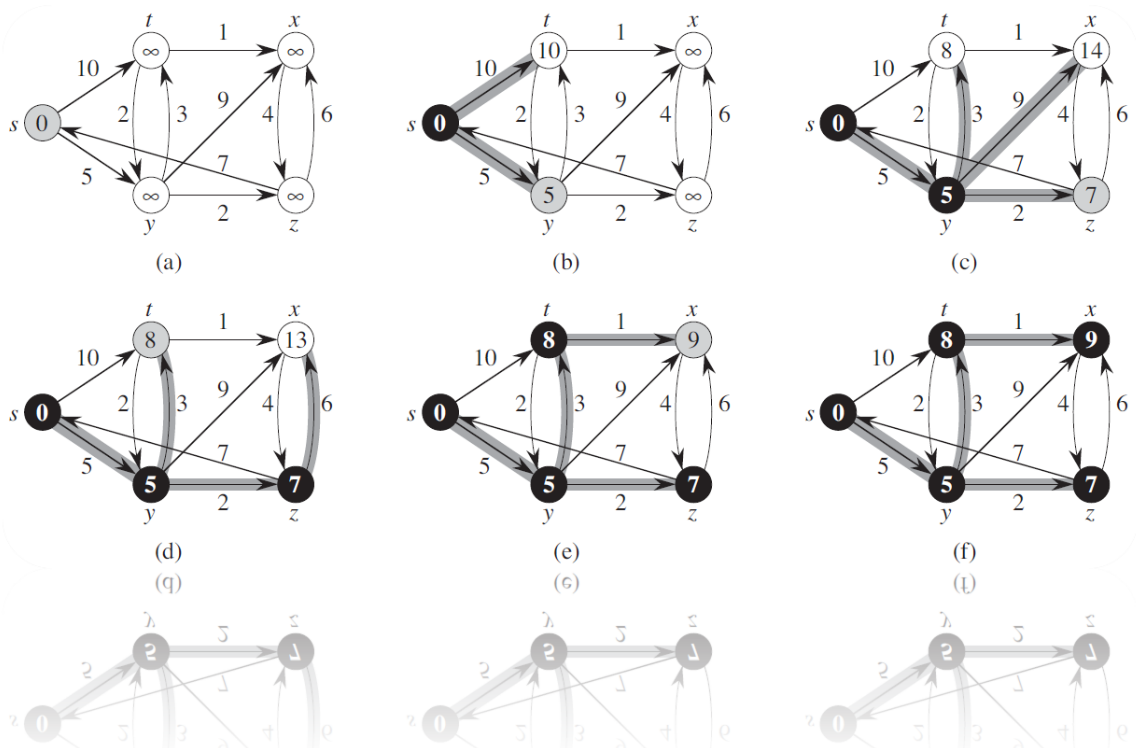


Figure A.9: An illustrative example of Dijkstra's Shortest Path algorithm. Shaded edges indicate potential predecessor values. The set S of vertices whose final shortest-path weights from the source s have already been determined are marked in black. Picture taken from [85]

Prim's Minimum-Spanning-Tree Algorithm

Given a weighted graph $G_o = (V, E)$, a **Spanning Tree** of that graph is a sub-graph which is a tree that connects all the vertices together. Thus, a specific graph might have various of such spanning trees. A **Minimum Spanning Tree (MST)** ([73]; [473]; [85]) would be that spanning tree where the sum of weights of all edges involved is less than the sum of weights of every other spanning tree. Prim's algorithm ([369]; [422]; [85]), as used in section 11.4 (part IV), is one algorithm that computes such a MST. It starts from an arbitrary vertex $s \in V$ and grows until the tree spans all vertices in V. In each step it adds an edge to the tree that connects the tree to a vertex $v \in V$ on which no edge yet exists. As there will be more than one potential edge, the edge with the minimal weight is chosen. The algorithm is illustrated in figure A.10.

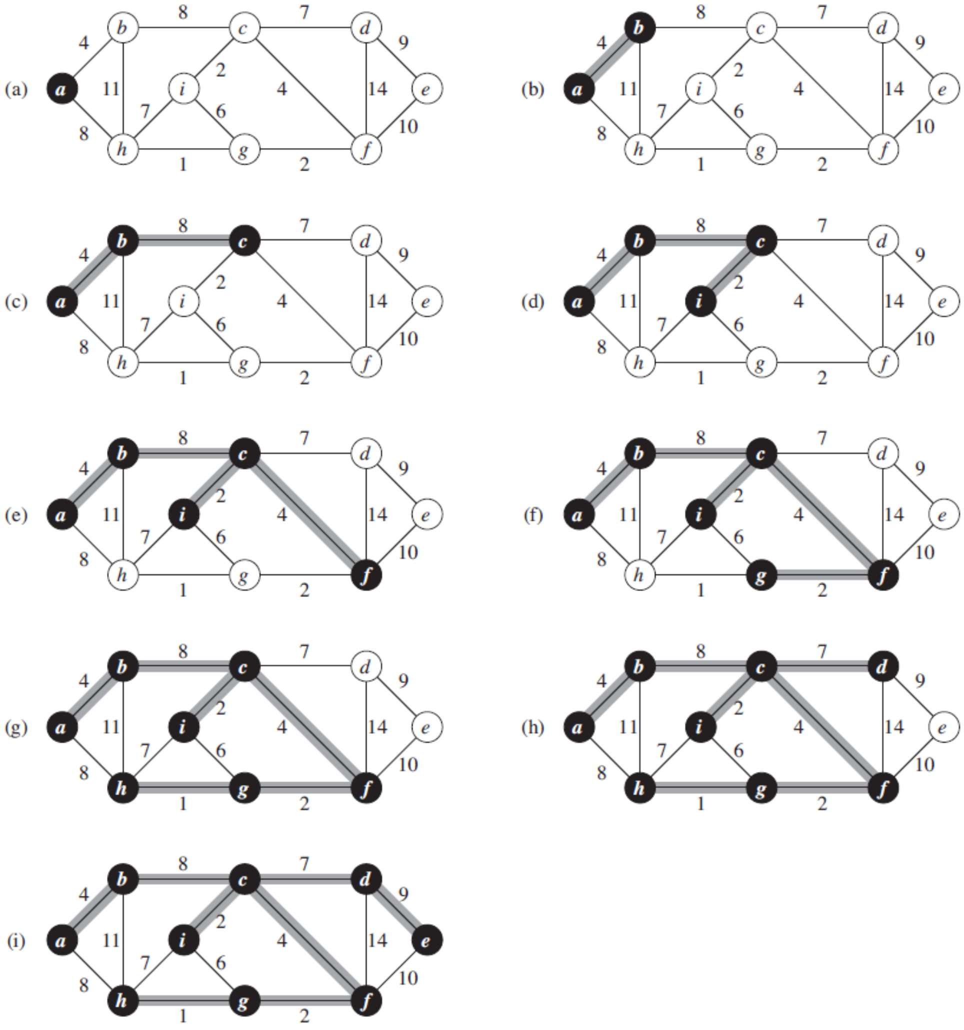


Figure A.10: An illustrative example of Prim's algorithm to form a Minimum Spanning Tree. Starting from vertex "a", shaded edges are added to the tree until all vertices have been connected, visualized in black. Picture taken from [85]

Graph Cuts and the Min-Cut / Max-Flow Problem

Given a weighted graph $G_o = (V, E)$, with V being a set of nodes and E a set of directed edges in between. The set of nodes now contains two additional specific terminal nodes $V = \{s, t\}$, called source (s) and sink (t), as illustrated in figure A.11. Each edge in G is assigned a non-negative weight $w(p, q)$ and $w(p, q)$ might differ from $w(q, p)$. An edge connecting a non-terminal with a terminal node is called a "t-link". Edges connecting two non-terminal nodes are called "n-links".

The Min-Cut / Max-Flow Problem

An s/t cut C is a partitioning of the nodes in the graph into two disjoint subsets S and T such that the source s would be in S and the sink t is in T. The cost of a cut $C = \{S, T\}$ is the sum of weights of all "boundary" edges (p, q) such that $p \in S$ and $q \in T$. If (p, q) is a boundary edge, then one can say that cut C severs edge (p, q). The **minimum cut** problem would then be to find a cut that has minimal costs among all possible cuts.

As a fundamental result in combinatorial optimization ([84]; [281]), the minimum s/t cut problem can be solved by finding a **maximum flow** from the source s to the sink t. The maximum flow problem can be illustrated as finding these edges, visualized as "pipes" of certain capacity, i.e., edge weights, that allow a "maximum amount of water" from the source to the sink. Thus, the theorem of Ford and Fulkerson [152] states that such a maximum flow from s to t saturates a set of edges in the graph dividing the nodes into two disjoint parts $\{S, T\}$ corresponding to a minimum cut. Therefore, the min-cut and max-flow problem formulations are equivalent. In fact, the maximum flow value is equal to the cost of the minimum cut. An early use of graph cuts for energy minimization in vision is due to Geig et al. in [181], which considers the problem of binary image restoration of binary images corrupted by noise. As min-cut/max-flow algorithms are principally binary techniques, binary problems constitute the most fundamental case for graph cuts.

Standard polynomial time algorithms for min-cut/max-flow exist ([84]; [281]; [45]; [46]), which are be divided into two main groups: approaches known as "push-relabel" methods ([172]; [45]) and algorithms based on augmenting paths [45]. In practice the push-relabel algorithms perform better for general graphs. In vision applications, however, common types of a graph are two or a higher dimensional grids. For the grid graphs, Boykov and Kolmogorov [45] developed a fast augmenting path algorithm which often significantly outperforms the push relabel algorithm with linear running time.

In general, graph construction as in figure A.11 can be used for arbitrary binary "labeling" problems. In section 7.4 (part III) of our work it is the essential element of the utilized approach to image segmentation by [401]. In section 11.2 (part IV) it is employed to do image labeling considering neighbor interactions. Suppose we are given a penalty $D_p(l)$ that pixel p incurs when assigned label $l \in L = \{0, 1\}$ and we need to find a spatially coherent binary labeling of the whole image. We can then state a spacial regularization via some global energy function:

$$E(f) = \sum_{p \in P} D_p(f_p) + \sum_{(p,q) \in N} V_{pq}(f_p, f_q) \tag{A.6}$$

The question is would then be whether one can find a globally optimal labeling f using some graph cuts construction and a definitive answer to this question exists for the case of binary labelings. According to Kolmogorov and Zabih [254], a globally optimal binary labeling for (A.6) can be found via graph cuts if and only if the pairwise interaction potential V_{pq} satisfies

$$V_{pq}(0, 0) + V_{pq}(1, 1) \leq V_{pq}(0, 1) + V_{pq}(1, 0) \tag{A.7}$$

(A.7) is called the regularity condition.

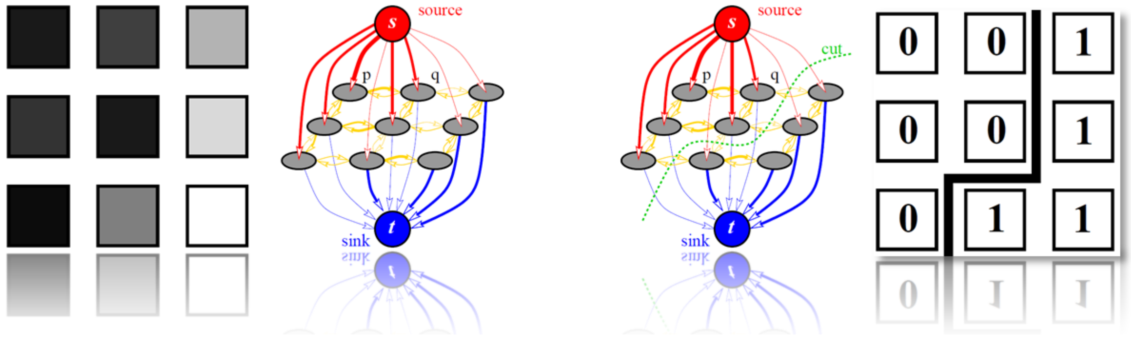


Figure A.11: Left: An example image and corresponding graph representation. Right: The minimum cut and final image labeling. Graph construction is equivalent to the one by Greig et al. in [181]. Picture modified from [45]

A.6 The Irrklang Audio Engine

The irrKlang audio engine ([8]; [2]) is a high level 2D and 3D sound engine and audio library that can play and post-process almost any common file format, such as WAV, MP3, OGG, just to name a few. irrKlang is cross platform, currently supporting Microsoft Windows, Mac OS X and Linux platforms. Within Microsoft Windows, irrKlang makes use of Direct Sound [233], the audio component of the Microsoft DirectX library [233], which provides an interface to the sound card driver. irrKlang offers several interesting sound effects to post-process audio files, such as, e.g., Volume, Chorus, Distortion, Echo, Flanger, Compressor, Reverb, Stereo Panning, 3D Doppler Effect and so forth. As it is possible to enable or disable these effects during playback for every single sound, as well as to adjust parameters of the effects if it is already active, they are very convenient to be harnessed for sonification, as employed in section 12.2. The usage of such a library therefore very easily allows further research and even sonify additional features with the same number of sounds only by altering parameters of effects applied. The following basic example shows how to start up the engine and play an MP3 file in C++:

```
#include <iostream>
#include <irrKlang.h>

using namespace irrklang;

int main(int argc, const char** argv)
{
  // start irrKlang with default parameters
  ISoundEngine* engine = createIrrKlangDevice();

  if (!engine)
    return 0; // error starting up the engine

  // play some sound stream, looped
  engine->play2D("somefile.mp3", true);

  char i = 0;
  std::cin >> i; // wait for user to press a key

  engine->drop(); // delete engine

  return 0;
}
```

A.7 Learning Theory & Support Vector Classifiers

The main goal of statistical learning theory ([483]; [482]; [417]; [61]; [86]; [430]) is to provide a framework for studying the problem of making predictions or decisions or constructing models from a set of data. Assumptions can be made of the statistical nature about the underlying phenomena. One of the original problems to model using learning theory is that of (binary) pattern recognition. Thus, given two classes of entities and being faced with a new unknown object, one has to assign this new unclassified object to one of the two classes. The problem can be formalized as follows. Suppose we are given m observations. Each observation consists of a pair: a vector $\boldsymbol{x}_i \in \mathcal{R}^n, i = 1, ..., m$ and the associated "ground truth" labeling $y_i \in \{-1, 1\}$. Given the empirical data

$$(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_m, y_m) \in \mathcal{X} \times \{-1, 1\} \tag{A.8}$$

one wants to estimate a **decision function** $f \to \mathcal{X} \times \{-1, 1\}$. An appropriate decision function will have the property to generalize to unseen data points, achieving a rather small value of the **actual** or **expected risk**:

$$R(\alpha) = \int \frac{1}{2} \mid y - f(\boldsymbol{x}, \alpha) \mid dP(\boldsymbol{x}, y) \tag{A.9}$$

Being a statistical model, it is assumed that some unknown probability distribution $P(y, \boldsymbol{x})$ from which these data are drawn exists, i.e., all data is assumed to be independently drawn and identically distributed (iid). If a certain density $p(y, \boldsymbol{x})$ exists, $dP(y, \boldsymbol{x})$ might be written $p(\boldsymbol{x}, y)d\boldsymbol{x}dy$, as a way of writing the true mean error rate. However, unless $P(\boldsymbol{x}, y)$ is known it is of no use at all. Statistical learning theory shows that it is crucial to restrict the set of functions from which f is chosen to one that has a capacity suitable for the number of available training data. It provides some bounds on the test error, depending on the capacity of the function class and the empirical risk. Subsequently, minimizing of such bounds leads to the principle of **structural risk minimization** ([482]; [417]; [61]; [86]). The **empirical risk** $R_{emp}(\alpha)$ is defined to be the measured mean error rate on the training set, for a finite number of observations:

$$R_{emp}(\alpha) = \frac{1}{2m} \sum_{i=1}^{m} \mid y_i - f(\boldsymbol{x}_i, \alpha) \mid \tag{A.10}$$

$R_{emp}(\alpha)$ is a fixed number for a particular choice of α and a particular training set $\{\boldsymbol{x}_i, y_i\}$, without any probability distribution attached. The quantity $\frac{1}{2} \mid y_i - f(\boldsymbol{x}_i, \alpha) \mid$ is known as the **loss function**.

For binary classification, it can only take the values 0 and 1. Thus, choosing some η such that $0 \leq \eta \leq 1$ yields the following bound for previously mentioned loss functions, with probability 1 - η ([417]; [61]; [86]):

$$R(\alpha) \leq R_{emp}(\alpha) + \sqrt{(\frac{h(\log(2m/h)+1) - \log(\eta/4)}{m})} \tag{A.11}$$

h denotes a non-negative integer called the **Vapnik Chervonenkis (VC) dimension** and describes a property of a set of functions $\{f(\alpha)\}$. It can be defined for various classes of function f. For functions, corresponding to the two-class pattern recognition case, if a given set of m observations can be labeled in all possible $2m$ ways, and for each labeling, a member of the set $\{f(\alpha)\}$ can be found that correctly assigns those labels, this set of observations is described to be "shattered" by that set of functions. Hence, the VC dimension for such a set of functions $\{f(\alpha)\}$ is then defined as the maximum number of training points that can be "shattered" by $\{f(\alpha)\}$. Note that, if the VC dimension would be h, then there is at least one particular set of h points that can be shattered. In general, it will, however, not be true that each set of h points can be shattered.

The second term on the right hand side of (A.11) is, therefore, called the "VC confidence" and the whole right hand side of (A.11) the "risk bound", which is independent of $P(\boldsymbol{x}, y)$. The risk bound only presumes that both, the training and test data, are drawn independently according to $P(\boldsymbol{x}, y)$. Generally, it is not possible to compute the left hand side, however, if h is known, one can compute the right hand side.

Structural Risk Minimization

Thus, given several different families of functions $f(\boldsymbol{x}, \alpha)$, called "learning machines" and choosing a fixed, sufficiently small η, by then taking that machine which minimizes the right hand side, one can choose that machine which provides the lowest upper bound on the actual risk. This provides a method for selecting a learning machine for a given task and is the fundamental idea of structural risk minimization. The VC confidence is described as a monotonic increasing function of h (see in figure A.12) for any number of observations m and given a selection of learning machines whose empirical risk would be equal to zero, one wants to select a learning machine whose set of functions, associated with it, has minimal VC dimension, which then leads to a better upper bound on the actual error, as illustrated in figure A.12. In general, for non zero empirical risk, one selects that specific learning machine that will minimize the right hand side of (A.11).

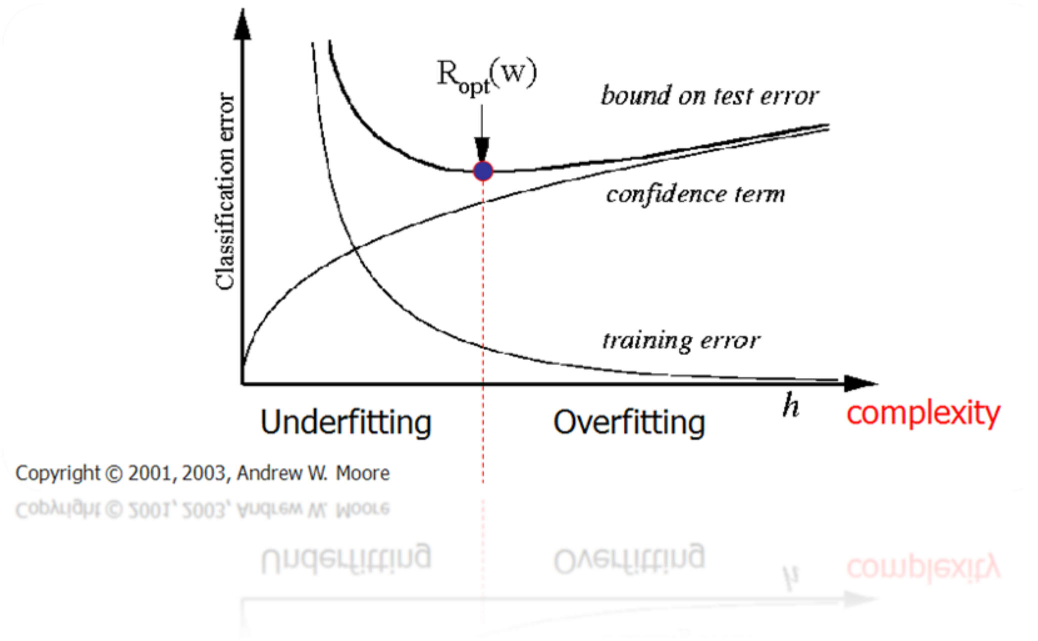


Figure A.12: Structural risk minimization: Choosing the best learning machine model

Support Vector Machines

Support Vector Machines (SVM) can be considered an approximate implementation of the principle of structural risk minimization, as they try to minimize a combination of the empirical risk in (A.10), and a capacity term derived for the class of **hyper-planes** in a dot product space $\mathcal{H}$ ([482]; [483]; [417]; [61]; [86]),

$$\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = 0 \quad \text{with} \quad \boldsymbol{w} \in \mathcal{H}, b \in \mathbb{R} \tag{A.12}$$

corresponding to decision functions

$$f(x) = sgn(\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b).$$

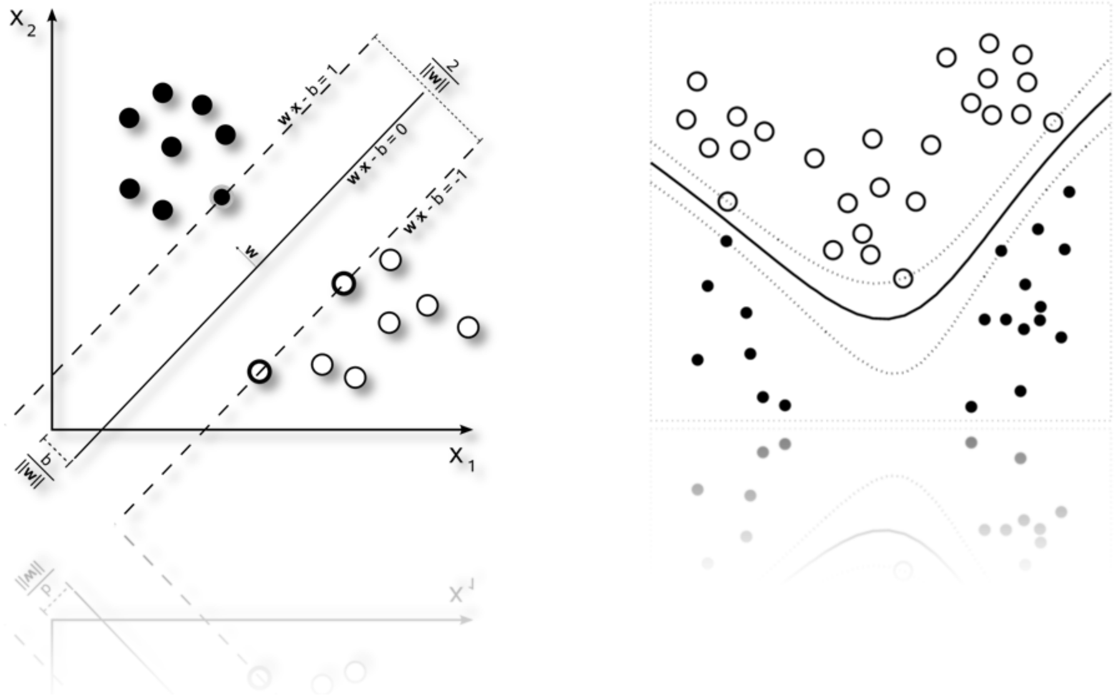


Figure A.13: Left: Linear separable classification. Optimal hyperplane is shown as a solid line. Weight vector $\boldsymbol{w}$ and a threshold b yield $y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) > 0 \quad \forall i = 1, ..., m$. Support Vectors lie on the borders of the margin (dashed lines). Right: Examples of a non-linear separation surface found using a radial basis function kernel $k(\boldsymbol{x}, \boldsymbol{x'}) = e^{\|\boldsymbol{x}-\boldsymbol{x'}\|^2}$

Hard and Soft Margin Solutions

In case of a linearly separable set of observations a unique optimal hyper-plane exists, differentiated by the maximal margin of separation between any observation point $\boldsymbol{x}_i$ and the hyper-plane, as visualized in figure A.13 (left). Such a case is called a hard margin solution. Such an optimal hyper-plane would be the solution of

$$\underset{\boldsymbol{w}\in\mathcal{H},b\in\mathbb{R}}{maximize} \min\{\|\boldsymbol{x}-\boldsymbol{x}_i\| \mid \boldsymbol{x}\in\mathcal{H}, \langle\boldsymbol{w},\boldsymbol{x}\rangle + b = 0, i = 1,...,m\} \tag{A.13}$$

Moreover, the capacity of the class of separating hyper-planes decreases with increasing margin. As illustrated in figure A.13 (left), in order to construct the optimal hyper-plane, on needs to solve the following quadratic programming problem

$$\underset{\boldsymbol{w}\in\mathcal{H},b\in\mathbb{R}}{minimize} \frac{1}{2}\|\boldsymbol{w}\|^2 \quad \text{subject to} \quad y_i(\langle\boldsymbol{w},\boldsymbol{x}_i\rangle + b) \geq 1 \quad \forall i = 1,...,m \tag{A.14}$$

The constraints ensure that $f(\boldsymbol{x}_i)$ will be +1 for y_i = +1, and -1 for y_i = -1. This constrained optimization problem in (A.14) is computed based on **Lagrange multipliers** ([391]; [61]) $\alpha_i \geq 0$ ($\boldsymbol{\alpha} := (\alpha_1,...,\alpha_m)$) and a **Lagrangian**

$$L(\boldsymbol{w},b,\boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{m}\alpha_i(y_i(\langle\boldsymbol{w},\boldsymbol{x}_i\rangle + b) - 1) \tag{A.15}$$

L has some saddle point in $\boldsymbol{w}$, b and $\boldsymbol{\alpha}$ at the optimal solution of the primal optimization problem. Therefore, it is minimized with respect to the **primal variables** $\boldsymbol{w}$ and b and maximized with respect to the **dual variables** α_i. Moreover, the product between constraints and Lagrange multipliers in L diminishes at optimality, i.e.,

$$\alpha_i(y_i(\langle\boldsymbol{w},\boldsymbol{x}_i\rangle + b) - 1) = 0 \quad \forall i = 1,...,m \tag{A.16}$$

,which is known in optimization theory [148] as **Karush-Kuhn-Tucker conditions** ([148]; [61]). Minimization with respect to the primal variables requires

$$\frac{\partial}{\partial\, b}L(\boldsymbol{w},b,\boldsymbol{\alpha}) = -\sum_{i=1}^{m}\alpha_i\, y_i = 0 \tag{A.17}$$

$$\frac{\partial}{\partial\, \boldsymbol{w}}L(\boldsymbol{w},b,\boldsymbol{\alpha}) = \boldsymbol{w} - \sum_{i=1}^{m}\alpha_i\, y_i\, \boldsymbol{x}_i = 0 \tag{A.18}$$

The solution has some expansion (A.18) in terms of a subset of the observations, with non-zero α_i. This subset of observations are called **Support Vectors (SVs)**. Most often, only a fraction of the training examples actually end up being Support Vectors and due to the Karush-Kuhn-Tucker conditions, Support Vectors lie on the margin (see A.13 (left)). Thus, once the α_i have been found this can be harnessed to compute b . All remaining training examples $(\boldsymbol{x}_j, y_j)$ turn out to be irrelevant as their constraints $y_j(\langle\boldsymbol{w},\boldsymbol{x}_j\rangle + b) \geq 1$ can be discarded. Therefore, the hyper-plane is completely determined by the observations closest to it. Substitution of (A.17) and (A.18) into the Lagrangian (A.15) eliminates the primal variables $\boldsymbol{w}$ and b, yielding a problem, which is usually solved in practice, known as **dual optimization problem**:

$$\underset{\alpha \in \mathbb{R}^m}{maximize} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \, \alpha_j \, y_i \, y_j \, K_{ij} \quad \text{subject to} \quad \alpha_i \geq 0 \quad \forall i = 1, ..., m \wedge \sum_{i=1}^{m} \alpha_i \, y_i = 0 \tag{A.19}$$

with $K_{ij} := \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$. Based on (A.18), the decision function in (A.7) is written as

$$f(\boldsymbol{x}) = sgn \left(\sum_{i=1}^{m} y_i \, \alpha_i \, \langle \boldsymbol{x}, \boldsymbol{x}_i \rangle + b \right) \tag{A.20}$$

b can be computed using (A.16). Usually, a separating hyper-plane may not exist, for instance, if noise within the training data causes a large overlap of the classes. To allow for this case, **slack variables** $\xi_i \geq 0 \, \forall i = 1, ..., m$ are introduced in order to relax the constraints of (A.14) to

$$y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geq 1 - \xi_i \quad \forall i = 1, ..., m \tag{A.21}$$

Thus, a learning machine that generalizes appropriately is found by controlling both, the classifier capacity, based on $\|\boldsymbol{w}\|$, and the sum of the slack variables $\sum_{i=1}^{m} \xi_i$, which provides an upper bound on the number of training errors. Such a classifier, known as **soft margin solutions**, is obtained by minimizing the objective function

$$\frac{1}{2} \|\boldsymbol{w}\|^2 + C \sum_{i=1}^{m} \xi_i \tag{A.22}$$

subject to the constraints on ξ_i and (A.21). The constant $C > 0$ determines the trade-off between margin maximization and training error minimization. Again, this leads to the problem of maximizing (A.19), subject to modified constraint with the only difference from the separable case being an upper bound C on the Lagrange multipliers α_i. Another realization replaces the parameter C by a parameter $\nu \in (0, 1]$ that provides upper and lower bounds for the subset of examples which become Support Vectors and those which will have non-zero slack variables, respectively ([482]; [483]; [417]; [61]; [86]).

Non Linear Support Vector Machines

If the decision funtion f (A.7) is not linear, all above methods have to be generalized to these cases. Boser et al. [44], proved that a rather old method [7], referred to as **kernel trick**, can be used to accomplish this in a straightforward manner. Thereby, symmetric similarity measures of the form $k : \mathcal{H} \times \mathcal{H} \to \mathcal{R}$, with $(\boldsymbol{x}, \boldsymbol{x'}) \to k(\boldsymbol{x}, \boldsymbol{x'})$, are considered. These functions, given two observations x and x' return a real number value denoting their similarity. To allow for a variety of similarity measures and learning algorithms, the observations are represented as vectors in an arbitrary selected **feature space** $\boldsymbol{\Phi}$, due to the mapping: $\boldsymbol{\Phi} = \mathcal{X} \to \mathcal{H}$ with $\boldsymbol{x} \to \boldsymbol{\Phi}(\boldsymbol{x})$. The function k is often referred to as a **kernel**. Some popular kernel choice are **Gaussian**, $k(\boldsymbol{x}, \boldsymbol{x'}) = e^{-\frac{\|\boldsymbol{x}-\boldsymbol{x'}\|^2}{2\sigma^2}}$, or **Radial Basis Functions (RBF)**, $k(\boldsymbol{x}, \boldsymbol{x'}) = e^{\gamma\|\boldsymbol{x}-\boldsymbol{x'}\|^2}$, as shown in figure A.13 (right). Finally, f can be rewritten as

$$f(x) = sgn\left(\sum_{i=1}^{m} y_i\, \alpha_i\, k(\boldsymbol{x}, \boldsymbol{x_i}) + b\right) \tag{A.23}$$

Furthermore, in the quadratic optimization problem (A.19) the definition of K_{ij} becomes $K_{ij} = k(\boldsymbol{x_i}, \boldsymbol{x_j})$.

Probability Estimates

As discussed, SVM predicts only class labels without probability information. However, in section 11.2 (part IV) of our work we need probability estimates in the context of building a graphical model. The libSVM library provides an extension to provide probability estimates based on the approach described in [512]. Briefly, given n classes of data, for any observation $\boldsymbol{x}$, the goal would be to estimate

$$p_i = P(y = i|\boldsymbol{x}),\ i = 1, ..., n \tag{A.24}$$

Following the setting of the one-against-one (i.e., pairwise) approach for multi-class classification, we first estimate pairwise class probabilities

$$r_{ij} \approx P(y = i|y = i\ \wedge\ j, \boldsymbol{x}) \tag{A.25}$$

using an improved implementation ([362] ; [291]). If $\widehat{f}$ is the decision value at $\boldsymbol{x}$, then we assume

$$r_{ij} \approx \frac{1}{1 + e^{A\,\widehat{f}+B}} \tag{A.26}$$

where A and B are estimated by minimizing the negative log likelihood of training data (using their labels and decision values). It has been observed that decision values from training may overfit the model in (A.26), so five-fold cross-validation is conducted to obtain decision values before minimizing the negative log likelihood. After collecting all r_{ij} values, Wu et al. [512] propose various possible approaches to obtain p_i, $\forall i$.

Parameter Estimation

In both our application contexts in sections 11.2 and 11.4 (part IV), we check for the most appropriate parameters C (III), for linear SVM, and (C, γ) IV, for RBF based non-linear SVM, via grid search based (five-fold) **cross validation (CV)**, provided by the libSVM library.

A.8 Solid of Rotation

A Solid of Rotation ([193]; [247]) denotes a solid figure obtained by rotating a curve in an Euclidean plane around a straight line (i.e. an axis) lying on the same plane.

Figure A.14 illustrates several stages in the progression of rotating the following function around its rotational axis, creating some sort of a "vase" alike $3D$ structure:

$$f(r, \phi, z) = \{(2 + \tan(z)\ \cos(\phi, (2 + \cos(z))\ sin(\phi), z\}$$

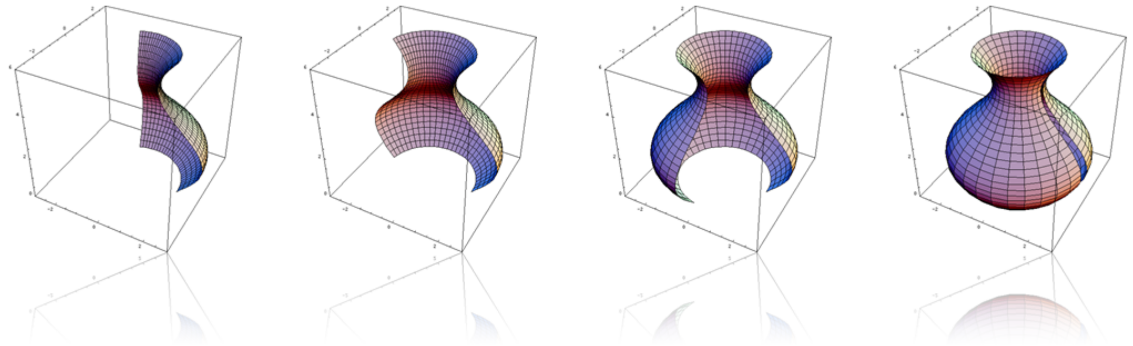


Figure A.14: Illustration of several stages in the progression of rotating the function $f(r, \phi, z)$ around its rotational axis done in Mathematica [384]. Parameters are $z \in \{0, 2\pi\}$, $\phi \in \{0, \alpha\}$ and $\alpha \in \{0.1, 2\pi\}$ with $\pi/12$ step-size. The plotting range is $x = \{-3, 3\}$, $y = \{-3, 3\}$ and $z = \{0, 6\}$

A.9 The Synthesis Toolkit

The Synthesis ToolKit (STK) ([82]; [83]) was developed as a set of C++ audio signal processing and algorithmic synthesis classes. The toolkit allows rapid development of music synthesis and audio processing software, with an emphasis on cross-platform functionality and real-time control. The STK currently supports audio and MIDI real-time processing on Linux, Mac OS X as well as Microsoft Windows computer platforms. It further has been ported to Symbian OS as well. It contains both, low-level synthesis and signal processing classes (oscillators, filters and so forth), as well as higher-level instrument classes. These instrument classes contain examples of state of the art physical modeling algorithms. The following example, taken from the example section from [82], illustrates the basic use of the SDK. In the example a stream "RtAudio dac" is initialized and "opened" for a callback function (called "tick()" function) to be processed and to write single audio frames to the sound device with a sample rate of 44100 Hz. In the example, the audio signal is a simple sine wave at 440 Hz.

```
#include "SineWave.h"
#include "RtAudio.h"
using namespace stk;

// This tick() function handles sample computation only.  It will be
// called automatically when the system needs a new buffer of audio
// samples.

int tick( void *outputBuffer, void *inputBuffer, unsigned int nBufferFrames,
         double streamTime, RtAudioStreamStatus status, void *dataPointer )
{
  SineWave *sine = (SineWave *) dataPointer;
  register StkFloat *samples = (StkFloat *) outputBuffer;

  for (unsigned int i=0; i<nBufferFrames; i++)
    *samples++ = sine->tick();

  return 0;
}
```

```
int main()
{
  // Set the global sample rate before creating class instances.
  Stk::setSampleRate(44100.0);

  SineWave sine;
  RtAudio dac;

  // Figure out how many bytes in an StkFloat and setup the RtAudio stream.
  RtAudio::StreamParameters parameters;
  parameters.deviceId = dac.getDefaultOutputDevice();
  parameters.nChannels = 1;
  RtAudioFormat format = (sizeof(StkFloat) == 8) ? RTAUDIO_FLOAT64
                                                : RTAUDIO_FLOAT32;
  unsigned int bufferFrames = RT_BUFFER_SIZE;

  try {
    dac.openStream(&parameters,NULL,format,(unsigned int)Stk::sampleRate(),
                   &bufferFrames,&tick,(void *)&sine);
  }
  catch (RtError &error) {error.printMessage(); goto cleanup; }

  sine.setFrequency(440.0);

  try {dac.startStream();}
  catch (RtError &error){error.printMessage();goto cleanup;}

  // Block waiting here.
  char keyhit;
  std::cout << "\nPlaying ... press <enter> to quit.\n";
  std::cin.get(keyhit);

  // Shut down the output stream.
  try {dac.closeStream();}
  catch (RtError &error) {error.printMessage(); }

 cleanup:
  return 0;
}
```

A.10 Thin Plate Splines

Thin Plate Splines (TPS) ([117]; [41]; [121]; [497]; [444]; [394]), introduced in 1976 by Duchon [121], describe a specific form of **radial basis functions (RBF)** ([60];[367]; [117]; [41]) $\varphi(r)$ (see figure A.15) of the form:

$$\varphi(r) = r^2 \, \log r \tag{A.27}$$

Typically, linear combinations of radial basis functions in general, or TPS in particular are employed to approximate other functions $f(r)$:

$$f(r) = \sum_{i \in N} \lambda_i \| \varphi(r) \| \tag{A.28}$$

Thereby, N denotes the number of applied radial basis functions, and λ_i are the coefficients, which weight each used RBF's contribution. The formulation given in (A.27), replaces the more hat-alike structure of, e.g., the Gaussian RBF (see figure A.15 (left)) by bowl-alike introducing smoother deviations (see figure A.15 (right)). TPS are commonly employed for representing coordinate mappings. As an example, Bookstein [41] and Davis et al. [102], scrutinize their application to the problem of modeling changes in biological shapes. However, the TPS interpolation model is inappropriate for the approximation of rather linear functions, as it itself is non-linear.

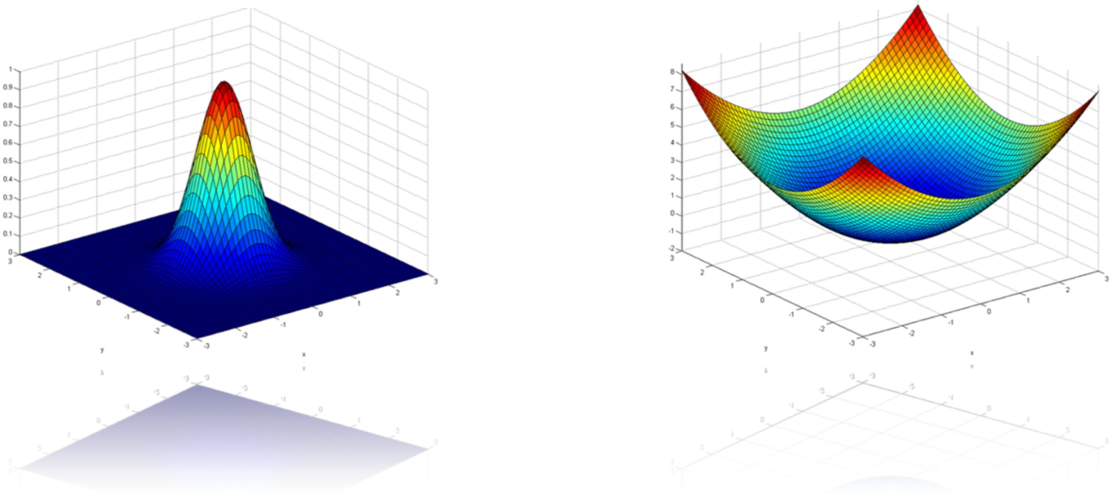


Figure A.15: Left: A Gaussian RBF. Right: The specific form of RBF, the Thin Plate Spline

3D Interpolation

The name Thin Plate Spline refers to a physical analogy of a flat thin medal plate (see figure A.16(left)) that is deformed by a few punctual strains, which we will call control values $\boldsymbol{c}$. The plate is than forced into a new form that minimizes the deformation energy (see figure A.16 (right)).

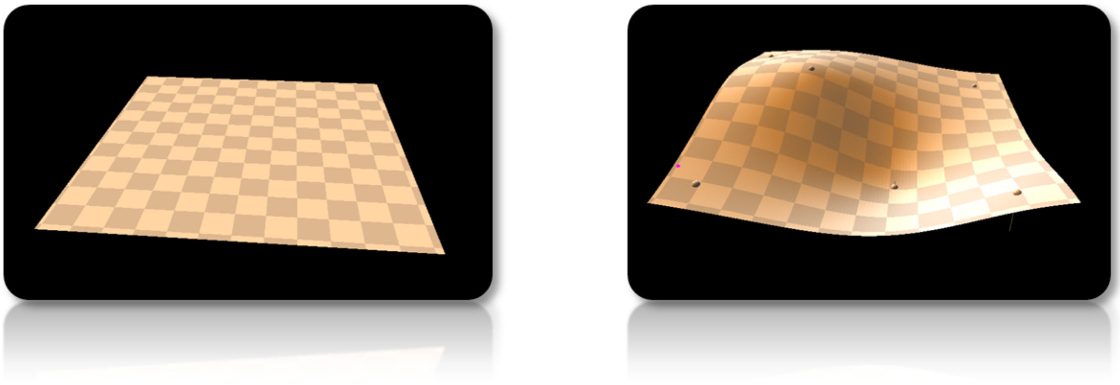


Figure A.16: Visualization of the Thin Plate Spline physical analogy. Left: A flat thin medal plate is deformed by a few punctual strains, called control values $\boldsymbol{c}$. Right: The plate is forced into a new form that minimizes the deformation energy

Given N control points $\boldsymbol{c}$, then z_{c_i} denotes the target function value at (x_{c_i}, y_{c_i}) of $\boldsymbol{c}_i$. It is assumed that the locations (x_{c_i}, y_{c_i}) are all different and not collinear. The Thin Plate Spline interpolant $f(x, y)$, therefore, takes the form

$$f(x,y) = a_0 + a_x x + a_y y + \sum_{i=1}^{N} \lambda_i \varphi(\|(x_{c_i}, y_{c_i}) - (x,y)\|) \tag{A.29}$$

with $\varphi(r)$ being the radial basis functions employed. In order for $f(x, y)$ to provide square integrable second derivatives, it is required that

$$\sum_{i=1}^{N} \lambda_i = 0 \ \wedge \ \sum_{i=1}^{N} \lambda_i x_{c_i} = \sum_{i=1}^{N} \lambda_i y_{c_i} = 0 \tag{A.30}$$

The weights λ_i can be found by minimizing a cost function E_f, that involves the sum of squared distances to each control value $\boldsymbol{c}_i$, as well as the integral of the square of the second derivatives of $f(x, y)$. Such a term, describing the bending energy, serves as smoothing term

$$E_f = \iint \Big[\left(\frac{\partial^2 f}{\partial x^2}\right)^2 + 2\left(\frac{\partial^2 f}{\partial xy}\right)^2 + \left(\frac{\partial^2 f}{\partial y^2}\right)^2 \Big] dx\, dy \tag{A.31}$$

Finally, the interpolation conditions, $f(x_{c_i}, y_{c_i}) = z_{c_i}$, and (A.30) yield a linear system to obtain the weights λ_i

$$\begin{vmatrix} \boldsymbol{K} & \boldsymbol{P} \\ \boldsymbol{P}^T & \boldsymbol{O} \end{vmatrix} \begin{vmatrix} \boldsymbol{\lambda} \\ \boldsymbol{a} \end{vmatrix} = \begin{vmatrix} \boldsymbol{z} \\ \boldsymbol{o} \end{vmatrix} \tag{A.32}$$

On the left side of the linear system (A.32) , $K_{ij} = \varphi(\|(x_{c_i}, y_{c_i}) - (x_{c_j}, y_{c_j})\|)$, the i^{th} row of $\boldsymbol{P}$ is $(1, x_{c_i}, y_{c_i})$ and $\boldsymbol{O}$ is some 3×3 matrix of zeros. $\boldsymbol{\lambda}$ is a column vector formed from λ_i and $\boldsymbol{a}$ is the column vector with elements a_0, a_x and a_y.

On the right side of (A.32), $\boldsymbol{z}$ is a column vector formed from z_{c_i} and $\boldsymbol{o}$ a 3×1 column vector of zeros.

The $(p+3) \times (p+3)$ matrix of this system is described by $\boldsymbol{L}$ which is non-singular matrix, as discussed in [367]. If the upper left $p \times p$ block of $\boldsymbol{L}^1$ is denoted by $\boldsymbol{L}_p^1$, it can be shown that

$$E_f \propto \boldsymbol{z}^T \boldsymbol{L}_p^{-1} \boldsymbol{z} = \boldsymbol{\lambda}^T \boldsymbol{K} \boldsymbol{\lambda} \tag{A.33}$$

As an alternative one may relax the exact interpolation requirement based on regularization. The objective cost function to minimize would, therefore, be

$$E = a_0 + a_x x + a_y y + \sum_{i=1}^{N} (z_{c_i} - (f(x_{c_i}, y_{c_i}))^2 + \rho \iint \Big[\left(\frac{\partial^2 f}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 f}{\partial xy} \right)^2 + \left(\frac{\partial^2 f}{\partial y^2} \right)^2 \Big] dx\, dy \tag{A.34}$$

ρ denotes a regularization constant, representing the trade-off between the smoothness of the interpolated function and its progression through or near the control points $\boldsymbol{c}$. As discussed in ([168]; [497]), one can solve (A.34) by replacing the matrix K by $K + \rho I$ in (A.32), where I is the $p \times p$ identity matrix.

In section 8.1 (part III) we choose $\rho = 0$, as we want the function to definitely move through all the control points $\boldsymbol{c}$, which represent certain acoustical states.

As we do not need many control points, in part III, complexity of finding appropriate weights λ_i can be reduced considerably and solutions such as *Lower Upper (LU)* decomposition ([368]; [365]) can be employed. However, since inverting $\boldsymbol{L}$ is an $O(p^3)$ operation, more optimized methods, as in [117], should be employed for larger sets of control points.

Appendix B

Visual Perception

B.1 The Human Visual System

Color Vision

Color vision starts, when light of various wavelengths, emitted or reflected by objects of the surrounding world, enters the human eye and stimulates photo receptors. Humans have lightness receptors, called "rods" and 3 types of color receptors, called color "cones", within the retina, which are sensitive to short(S), medium (M) an long (L) wavelength of light which can loosely be assigned to blue, green and red. This theory of trichromatic color vision was first postulated in 1802 by Young [526]. Nevertheless, through neuro physiological studies [103] the theory of opponent colors by Hering ([202]; [237]) could be affirmed to play a significant role in the further processing of incoming color stimuli from the retina, and could therefore be connected to the trichromatic Theory found by Young and Helmholtz ([174]; [526]). According to Hering, humans perceive certain colors like blue, green, red and yellow as significantly clean, which is why Hering calls them unique-colors. All other colors are perceived as mixtures, whereupon 2 of the 4 colors cannot be mixed, such as there is no green-red or blue-yellow. Hering calls these 2 pairs opponent-colors. Figure B.1 visualizes the principle of color processing in the human brain. The cones within the retina deliver impulses to neural color channels. On a neural level, these color stimuli are interconnected. The difference of red and green is transmitted to the red / green channel. The addition of red and green creates the perception of yellow. The difference between blue stimuli and this perception of yellow is send to the blue/yellow channel. Luminance can be assigned to a white/black channel that measures R,G,B, against each other.

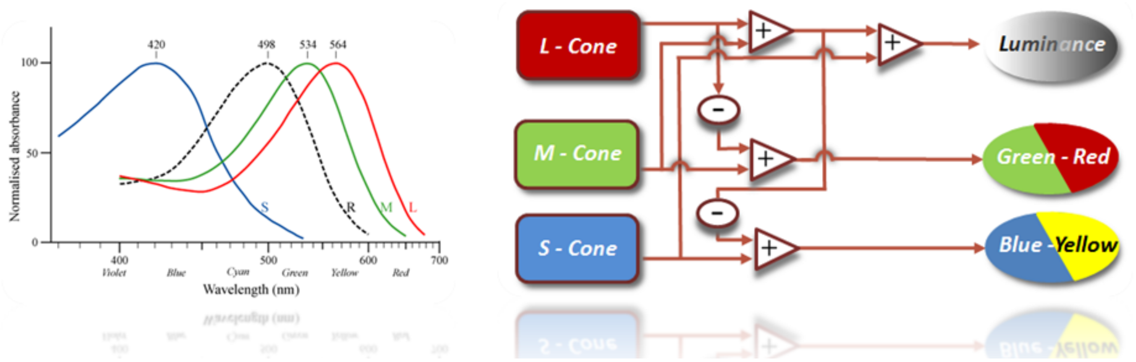


Figure B.1: Left: Normalized spectral absorption curves of color receptors (and rods in intersected black). (b) Color processing in the human brain from trichromatic to opponent dimensions. (Visualizations based on [174] and [237])

The Retina

The human eye is equipped with an "inverted-type" retina, which means that light must pass through all retinal layers before it reaches the photo-receptor cells that are aligned at the back of the tissue. As the vertebrate retina contains various structures that differ in size and refractive index, such differences should lead to significant scattering. Accordingly, Goldsmith [173] pointed out that the application of the inverted retina *is equivalent to placing a thin diffusing screen directly over the film in your camera.* Interestingly, in 2007, Franze et al. ([154]; [382]) discovered the remarkable properties of radial glial cells, also known as "Mueller cells", that span the entire retina from front to back. These cells act as optical fibers and guide light to the photo-receptor cells which otherwise would be scattered from the retinal surface.

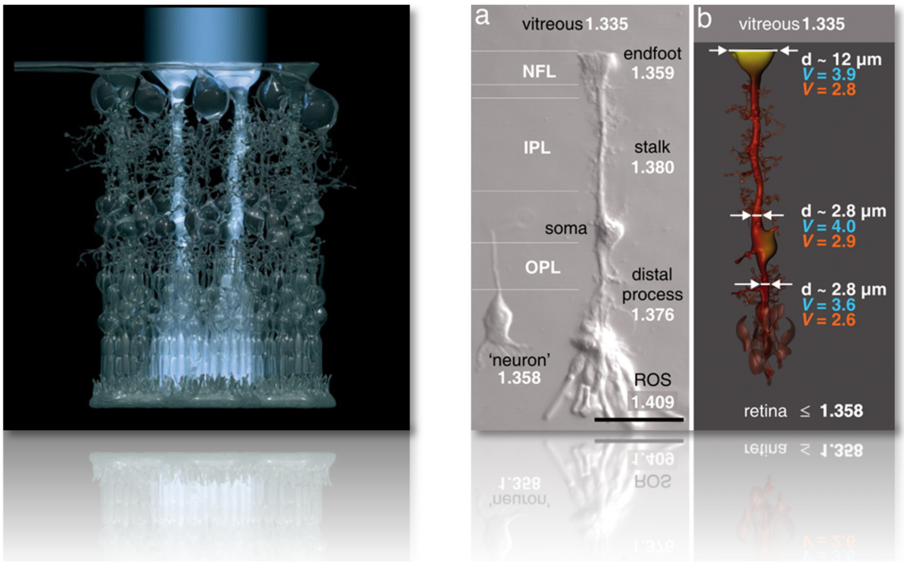


Figure B.2: Left: Mueller glial cells acting as living optical fibers, transporting light through the inverted retina. Due to their funnel-shaped endfeed, Mueller cells collect light at the retinal surface and guide it to photo-receptor cells on the opposite side. Images are thus transmitted through optically distorting tissue. Right: *Mueller cell shape, refractive properties, and light-guiding capability. (a) Nomarski differential interference contrast microscopy image of a dissociated (guinea pig) Mueller cell with several adherent photoreceptor cells, including their outer segments (ROS) and a dissociated retinal neuron (bipolar cell) to the left. (b) Schematic illustration of a Mueller cell in situ. The lighter the coloring of the Mueller cell, the lower the refractive index. Typical diameters and the calculated V parameters for 700 nm (red) and 500 nm (blue) are indicated at the endfoot, the inner process, and the outer process. Although diameters and refractive indices change along the cell, its light-guiding capability remains fairly constant. (Scale bar, 25 μm.).* Text and picture taken from [154]

The collective presence of these parallel optical fibers mediates the image transfer through the retina with minimal distortion. This recent findings explain a fundamental feature of the inverted retina as an optical system and ascribe new functions to glial cells. Light enters the Mueller cells, illustrated in figure B.2 at a shallow angle and is slowed down by the cells' high refractive index. When hitting the cells' boundaries, light is reflected back along the tube. Due to their funnel shape, Muller cells gather and transmit as much light as possible, and as they are narrow in the middle, they take up a very small amount of space and leave plenty of room for nerves and blood vessels that the retina needs. On average, every Muller cell serves several rod cells but only a single cone photo-receptor cell, which ensures that distortion-free and high contrast images that eventually hit the light sensors. Therefore, the way in which Mueller cells transport light is similar to that of optical fibers.

The Visual Pathway & The Visual Cortex

The purpose of this chapter is not to give an in depth introduction into the human visual system , but rather a short introduction to the visual cortex and its various visual areas and pick out specific features, taking place at various positions within the visual cortex, that resemble or inspired some of the low-level features we extracted in various sections of the work. For a more in depth introduction into the visual system and the visual cortex see ([237]; [179])

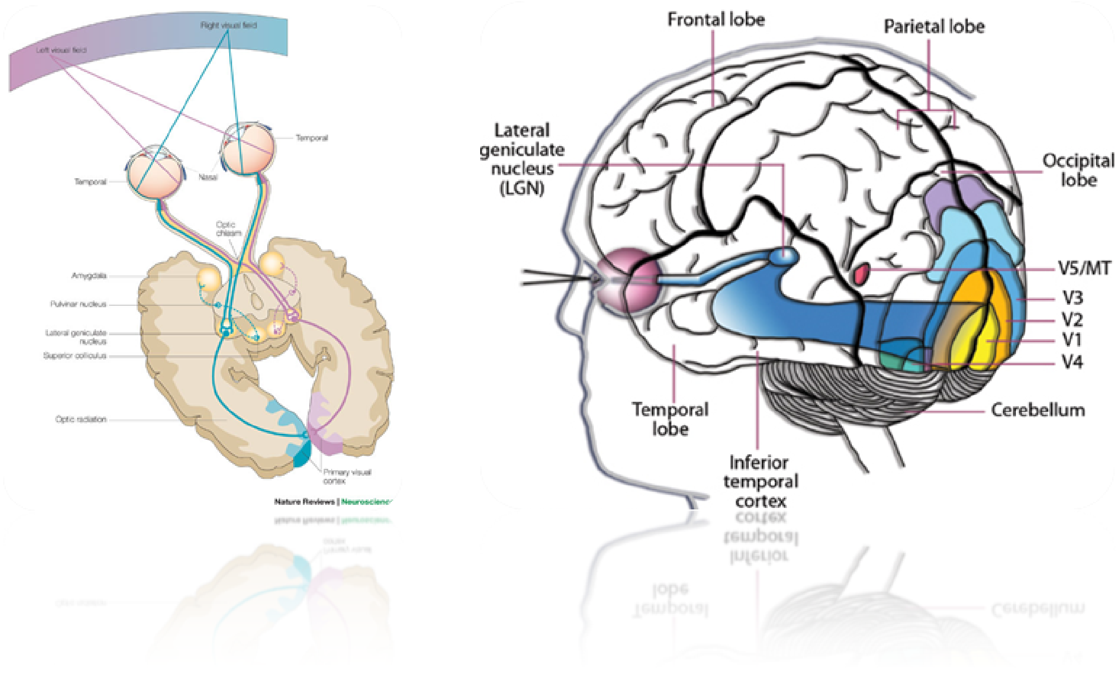


Figure B.3: Left: The visual pathway. Picture taken from [189]. Right: Overview over the *Visual Cortex* and its specific areas V_1, V_2, V_3 and V_4. Picture modified from [294]

At the start of the visual pathway, illustrated in figure B.3 (left), images from the retina at the back of each eye are channeled first to a pair of small structures deep in the brain called the **lateral geniculate nuclei (LGN)** ([294]; [237]). Individual neurons in the LGN can be activated by visual stimulation from either one eye or the other but not both. They respond to any change of brightness or color in a specific region within an area of view known as the receptive field, which varies among neurons. From the LGN, visual information moves to the primary visual cortex, known as V1, which is at the back of the head. Neurons in V1 behave differently than those in the LGN do. They can usually be activated by either eye, but they are also sensitive to specific attributes, such as the direction of motion of a stimulus placed within their receptive field. Visual information is transmitted from V1 to more than two dozen other distinct cortical regions.

Some information from V1 can be traced as it moves through areas known as V2 and V4 before winding up in regions known as the **inferior temporal cortex (ITC)** ([294]; [237]), which like all the other structures are bilateral. A large number of investigations, including neurological studies of people who have experienced brain damage, suggest that the ITC is important in perceiving form and recognizing objects. Neurons in V4 are known to respond selectively to aspects of visual stimuli critical to discerning shapes. In the ITC, some neurons behave like V4 cells, but others respond only when entire objects, such as faces, are placed within their very large receptive fields. Other signals from V1 pass through regions V2, V3 and an area known as MT/V5 before eventually reaching a part of the brain called the parietal lobe. Most neurons in MT/V5 respond strongly to items moving in a specific direction. Figure B.3 (right) illustrates the locations of the various visual areas.

Luminance and Brightness

Luminance is the measure of the per-area intensity of light travelling in a particular direction, measured in “candela per square metre” (cd/m2). **Illuminance** is a measure of the per-area incident of luminous flux (a measure of the perceived (adjusted based on human sensitivity to different wavelengths) power of light). There are several sources of illumination, and the eyes can cope with a vast range from starlight (10^{-4}) lux to direct sunlight ($32000 - 130000$) lux. The brain takes in illuminance, and from there computes brightness (stimuli and sensation). The perception of the apparent lightness of an object depends on the context in which this object is embedded in, as illustrated in figure B.4 ([11]; [250]; [398]; [5]; [164]; [165]).

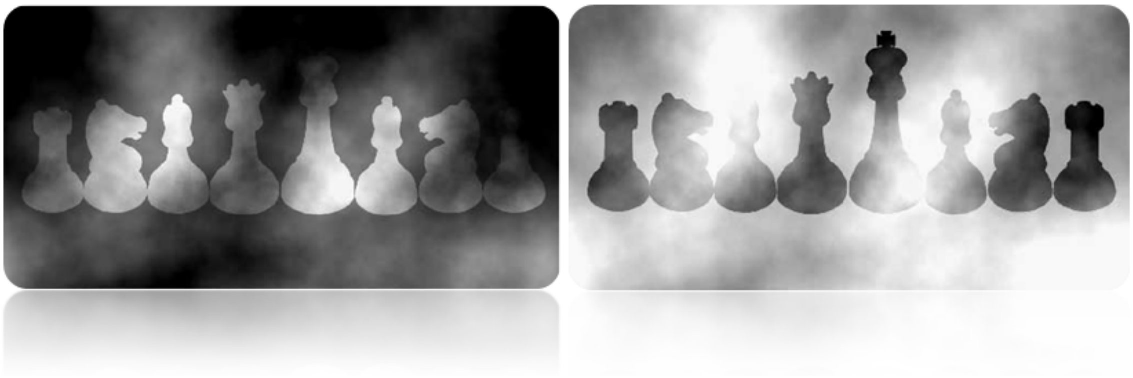


Figure B.4: Corresponding chess pieces on the two surrounds are physically identical. Figures on light surround appear as dark objects visible through light haze, whereas figures on dark surround appear as light objects visible through dark haze. Picture modified from [11]

The Kanizsa square, visualized in figure B.5 (left), shows that one can perceive the borders of an object even in regions of an image without direct visual evidence for them. This is an example of the phenomenon of illusory or subjective contours [239], which have a rich history in psychology ([443]; [346]), although they are still an active field in research.

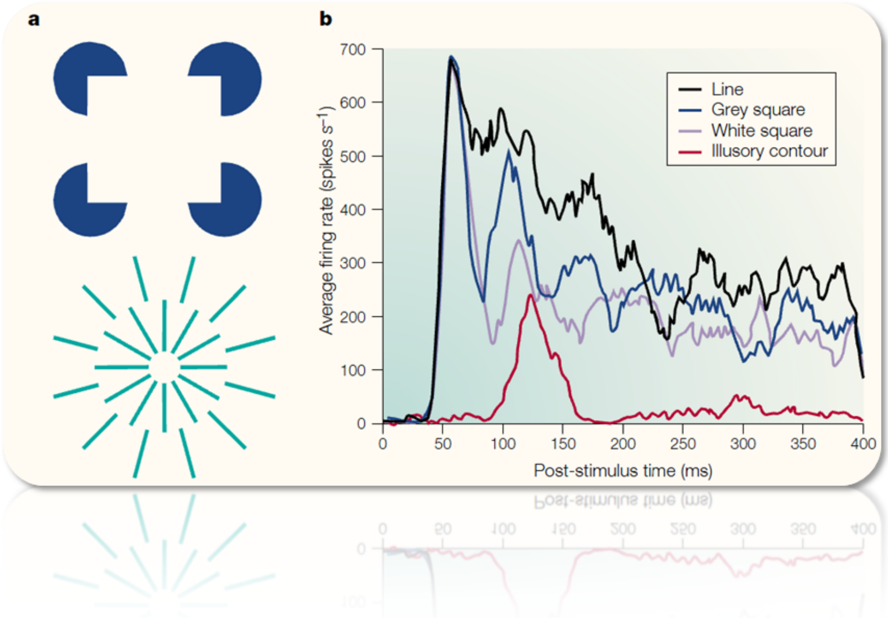


Figure B.5: Illusory contours and brightness enhancement. Left: Illusory contours (a) generated by the visual system from suggestions of an occluding figure (Kanisza square, top), or by fracture lines between two textures (circle, bottom). Note the illusory brightness enhancement accompanying the sides of the square as well as the central circle. Right: Illusory contour signals (b) in V1 are weaker and arrive later than signals in V2, indicating the perception of illusory contours to involve intercortical feedback interactions [283]. Picture modified from [127]

Simple & Complex Cells

Simple Cells, discovered by Hubel and Wiesel [215] in the primary visual cortex (V1), are retinal ganglion cells ([237]; [179]) that respond primarily to oriented edges. Thus, they can be as linear filters that compute a weighted sum of the intensities in a stimulus, with weights given by the receptive field [132]. Daugman discovered that simple cells within the visual cortex of mammalian brains can be modeled by Gabor functions ([99]; [282]), as illustrated in figure B.6. Hubel and Wiesel [215] defined simple cells as having distinct antagonistic regions, excitatory and inhibitory, in their receptive fields. They further suggested that knowing those regions, one could predict responses to any shape of a given stimulus, stationary or moving. Simple cells were complemented by the discovery of another kind of ganglion cells, which perform non-linear operations, suggesting that they sum the distorted output of sub-units that in turn have linear receptive fields ([132]; [487]).

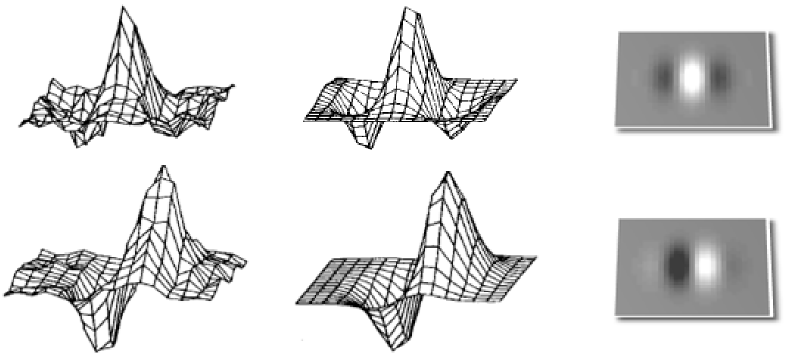


Figure B.6: Left: Illustrations of empirical 2D receptive field profiles in simple cells of the cat viual cortex . Middle and Right: Best fitting 2D Gabor wavelets representation. Picture modified from [230]

Complex cells, located in V1, V2 and the nearby Brodmann area, are insensitive to the specific position of a bar within the receptive field, and respond both to the onset and to the offset of the bar ([215]; [216]). They receive inputs from a number of simple cells and their receptive field would therefore be the summation and integration of the receptive fields of many input simple cells ([323]; [324]; [68]), shown in figure B.7.

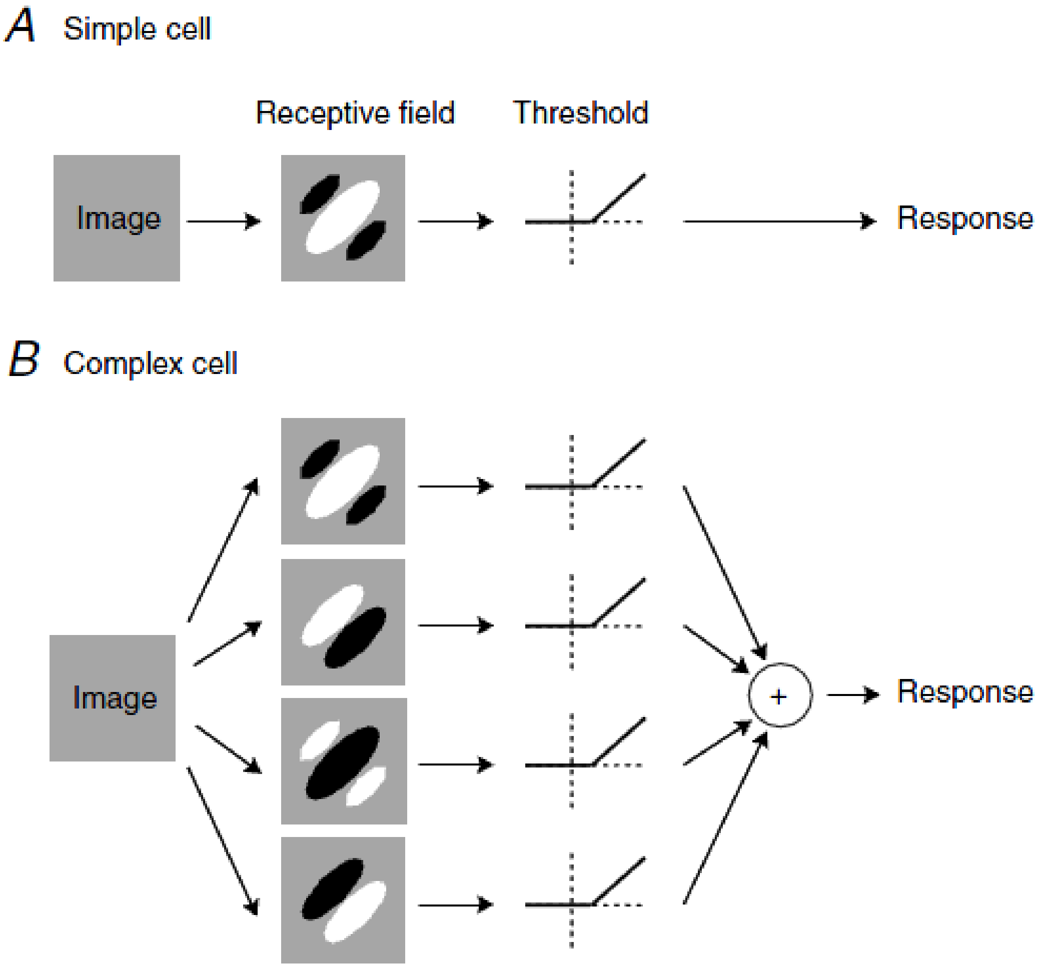


Figure B.7: *Left: The models of simple and complex cells proposed by Movshon, Thompson and Tolhurst ([323]; [324]). A: Linear model of simple cells. The first stage is linear filtering, i.e. a weighted sum of the image intensities, with weights given by the receptive field. The second stage is rectification: only the part of the responses that is larger than a threshold is seen in the firing rate response. B: Subunit model of complex cells. The first stage is linear filtering by a number of receptive fields such as those of simple cells (here only four of them with spatial phases offset by 90 degrees are shown). The subsequent stages involve rectification, and then summation.* Text and Picture taken from [68].

Grating Cells

Simple and complex cells respond to periodic stimuli and aperiodic stimuli such as sine- and square-wave grating, bars, and edges of a preferred orientation. Von der Heydt et al. ([492]; [119]) discovered a new type of cell in areas V1 and V2 that responded to periodic stimuli, which they called "grating cells", as they respond vigorously to periodic patterns and only weakly or not at all to aperiodic patterns such as bars or edges. Thus, these structures could presumably be responsible for texture processing in the visual system. Computational models inspired by these findings have been used as texture operator ([261]; [295]), illustrated in figure B.8.

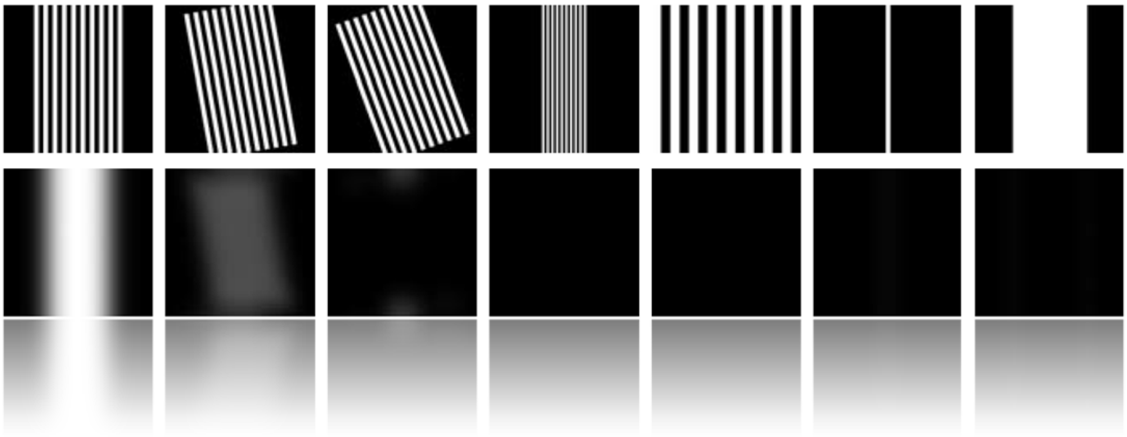


Figure B.8: Responses to square gratings of different orientations and spatial frequencies. Top rows show stimuli and bottom rows the responses of the modeled grating operator. Grating cells responded vigorously to grating patterns of preferred orientation and frequency. Responses decreased when stimuli differed from this pattern. Responses strongly decreased when the gratings were rotated by 10 degrees and completely vanished at a 20 degrees rotation. Halving or doubling the spatial frequency also abolished these responses. Picture modified from [295]

Surround Suppression

Relatively recent insights obtained in neurophysiology of the visual system reveal a neural mechanism called surround suppression which can observed in areas V1 and V2 ([339]; [229]; [69]; [439]). Essentially, it enables the response of an orientation selective neuron to a local oriented stimulus to be inhibited by the presence of other similar stimuli in the immediate surroundings. Recently, computational models based on Gabor wavelets have been proposed to simulate surround suppression ([344]; [354]), visualized in figure B.9.

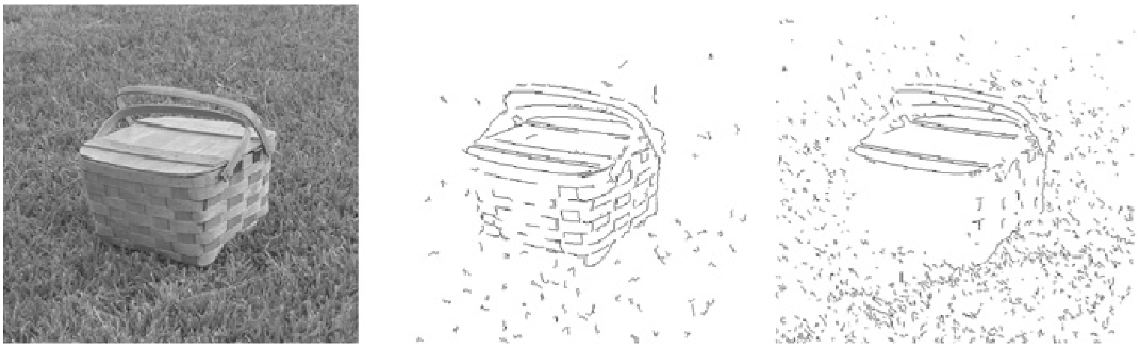


Figure B.9: Simulated surround suppression model by [344] (Middle) of an input image (left), compare to the plain results of the Canny edge detector (right). Picture modified from [344]

B.2 Color Spaces

Figure B.10: Entitled *Paris, Momatre - a painters paradise.* Picture by *Chris Willis* taken from *flickr*.

A “color space” is a concept to understand the color capabilities of a specific file or device. When reproducing color on another device, color spaces can be a reference to what extent details concerning shadowing or highlighting or color saturation can be retained or to what extent either will be compromised. As an artist mixes primary colors on a palette in order to visualize the range of colors and shades, as visualized in B.10, color spaces are sometimes represented as digital palettes, except these palettes are much more precisely quantified.

RGB

The RGB color space [286], standing for red, green and blue, is an additive color model, where these three primary colors of light are added to produce specific colors as combinations. It is based on the theory of trichromatic color vision proposed by Young and Helmholtz [526]. Thereby, the intensity of the light is determines the color perceived. No intensity leads to the perception of black, whereas full intensity leads to that of white. Differing intensities among primary colors are responsible for the hue of a color. The RGB model is the color space mostly utilized on, e.g., computer or TV monitors. Geometrically it is often depicted as a cube with red, green and blue occupying three vertices, as illustrated in figure B.11 (left).

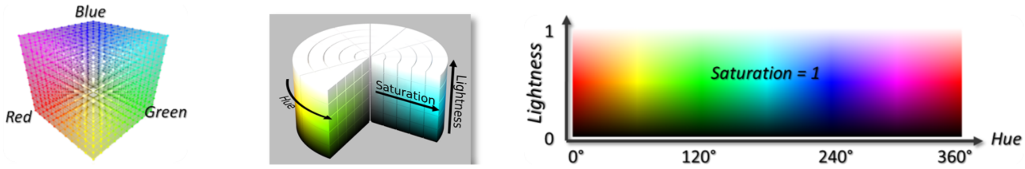


Figure B.11: Left: the RGB color model. Right: The HSL space in a cylindrical representation and unrolled

HSL

The HSL color space ([286]; [441]) stands for hue, saturation and lightness. Thereby, hue describes the color type, ranging from 0° to 360°. Saturation denotes the variation of the color depending on the lightness. Thus, the HSL space is rather intuitive. The hue channel is represented as a circle going through all the colors in the color wheel. Saturation indicates the intensity of color and lightness how bright a color would be (see B.11 (right)). Thus, the HSL model can be interpreted as a double-cone or a “stretched” cylinder, visualized in figure B.12 (right). Where the minimal saturation ($s = 0$) forms a straight line aling l within the double-cone model, it is more of a curve within the cylindrical model. This observation leads to the interpolation of the minimum saturation curve $s_{min}(l)$ in section 8.1.

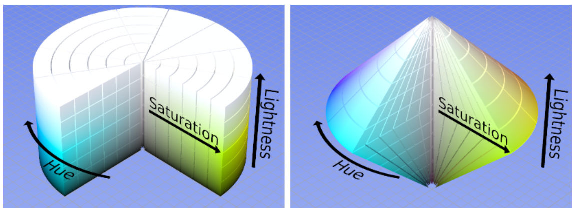


Figure B.12: The HSL color space as a cylinder (left) or double-cone model (right)

Irregularities in the HSL Color Space

As introduced in section, the HSL color space contains some “perceived irregularities”. Below 50% luminance around yellow ($h = 60°$) there is a certain region that would be visually perceived as “olive green” rather than dark yellow. Additionally, what should be a visual “deep blue” at $h = 240°$ increasing in luminance tends to fade into what is perceived visually as “violet”. These visual irregularities can be observed in comparison to the development of other colors in the HSL model (see figure B.13).

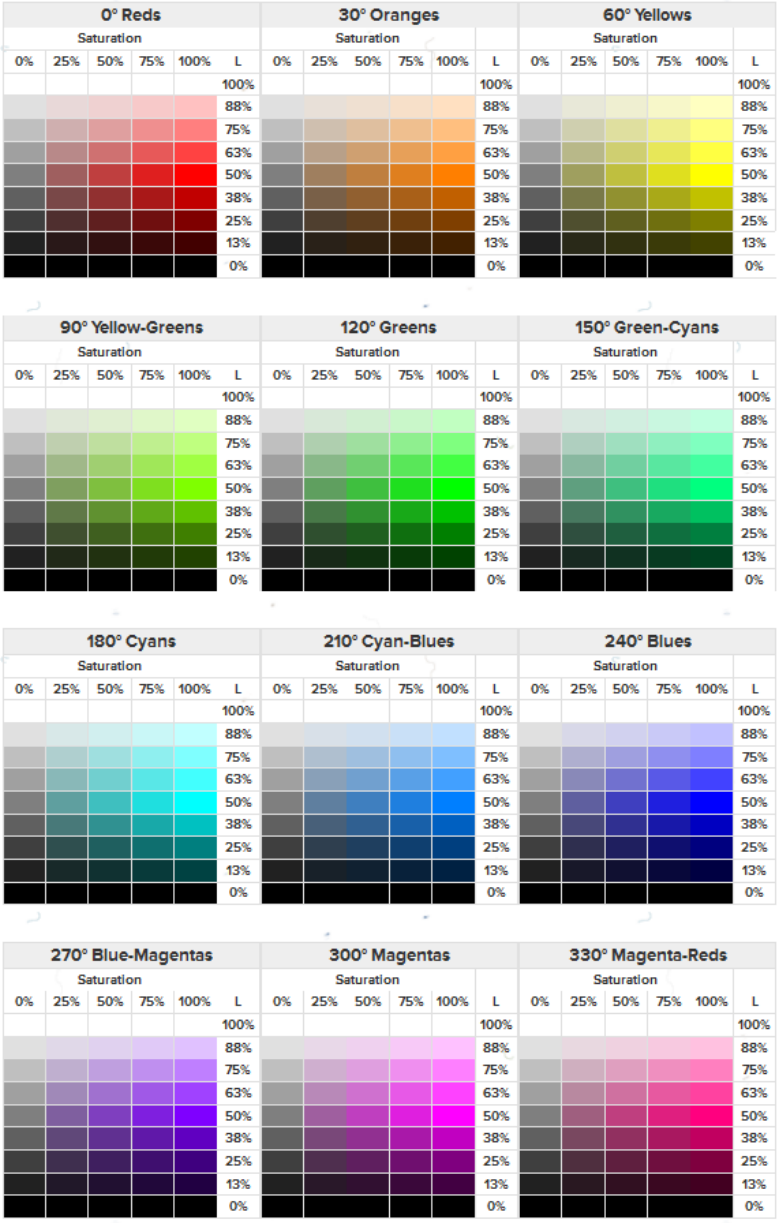


Figure B.13: Color Table of the HSL color space, taken form [441]

To deal with both irregularities, we match the acoustical perception to the visual. For the problem around yellow, we adjust the values of $\vartheta_{yellow}(h, s, l)$ along h as illustrated in figure B.14. Ranges of h for yellow below 50 % luminance were adjusted by our own perception, we, however, propose the application of region growing algorithms [397] based on representative "olive yellow-green" regions as seed points.

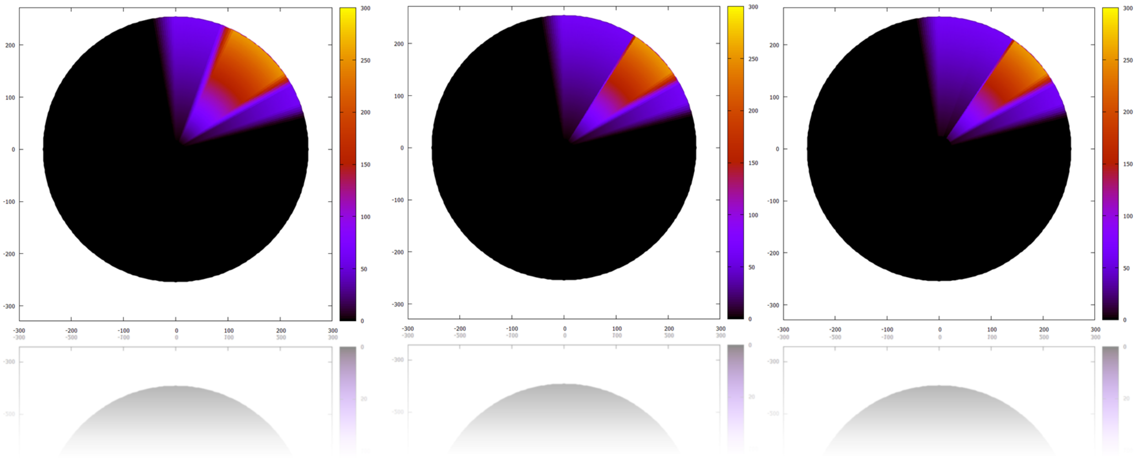


Figure B.14: Matching the acoustical perception around "olive" (green-yellow) to the visual. Exemplary values of $\vartheta_{yellow}(h, s, l)$ at 50 % (left), 35 % (middle) and 20 % (right). Note that the range of values for each $\vartheta(h, s)$ is from 0 (black) to max. 255 (orange-yellow)

To adjust the acoustical perception around "violet" $h = 240°$ to the visual we need to "add" some red. However, the region is rather unpredictable than the irregularity around yellow. Thus, due to our own observations of the color space, we set $\vartheta_{red}(h, s, l)$ at specific positions. These are $\vartheta_{red}(h_{265}, s_{50\%}, l_{70\%}) = 0.125$, $\vartheta_{red}(h_{265}, s_{100\%}, l_{70\%}) = 0.125$ at 70 % luminance. Further choices are $\vartheta_{red}(h_{265}, s_{50\%}, l_{50\%}) = 0.125$ at 50 % luminance and $\vartheta_{red}(h_{260}, s_{50\%}, l_{20\%}) = 0.125$ as well as $\vartheta_{red}(h_{265}, s_{100\%}, l_{20\%}) = 0.125$ at 20 % luminance. Subsequently, the Thin Plate Spline based diffusion is employed to "diffuse" these manipulated values $\vartheta_{red}(h, s, l)$ into their immediate surrounding area (at constant luminance level l for hue range from 260° to 265°). Results are shown isolated in figure B.15 and within the complete color circle in figures B.16 ($l = 20\%$), B.17 ($l = 50\%$) and B.18 ($l = 70\%$). Thereafter, interpolation of $\vartheta_{red}(h, s, l)$ along l is performed linearly.

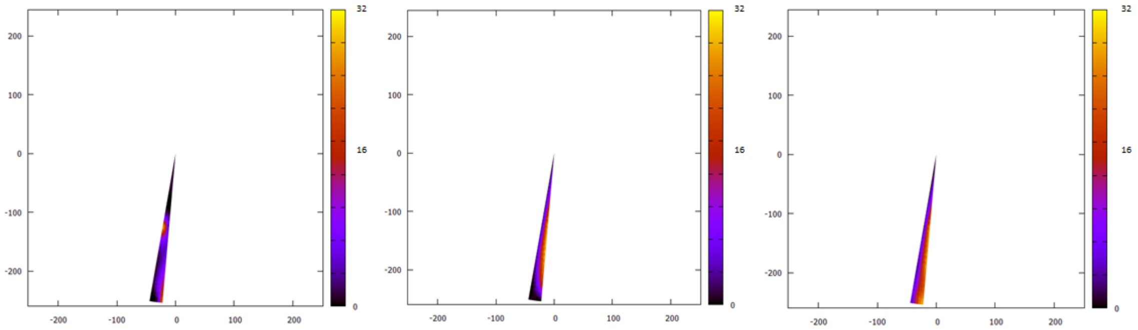


Figure B.15: Matching the acoustical perception around "violet" $h = 240°$ to the visual. Thin plate spline manipulated values $\vartheta_{blue}(h, s, l)$ and their diffusion in the immediate surrounding area at levels at 20 % (left), 50 % (middle) and 70 % (right). Note that the range of values for each $\vartheta(h, s)$ is from 0 (black) to max. $32 (= 0.125)$ (yellow)

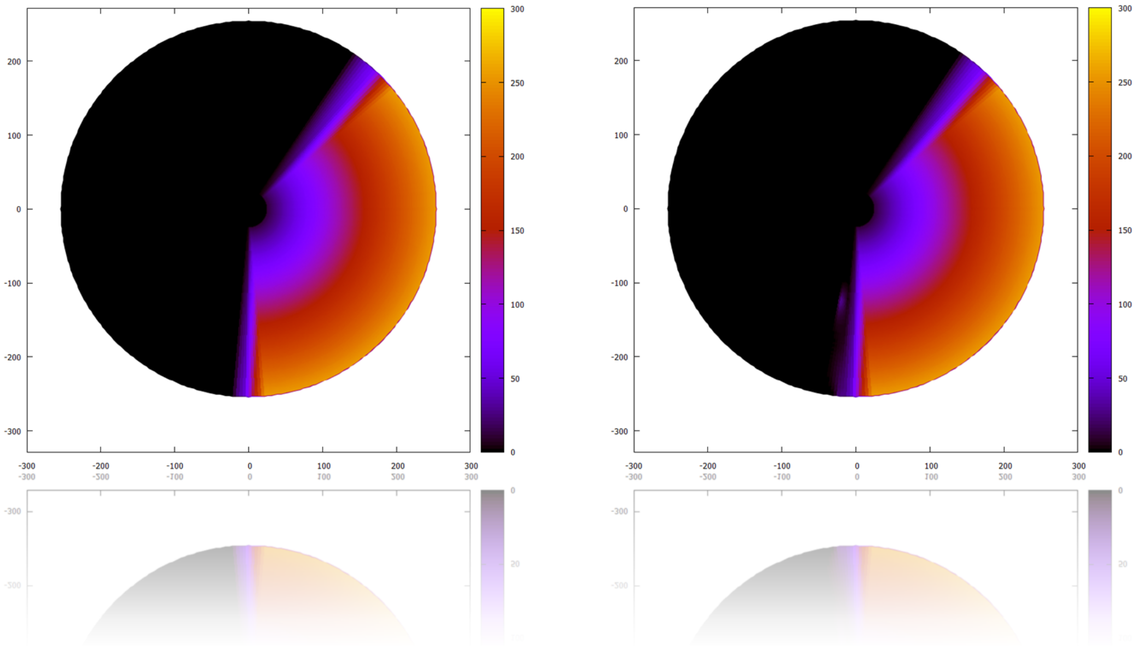


Figure B.16: Adjusted color circle at $l = 20\%$. Range of values is from 0 (black) to max. 255 (orange-yellow)

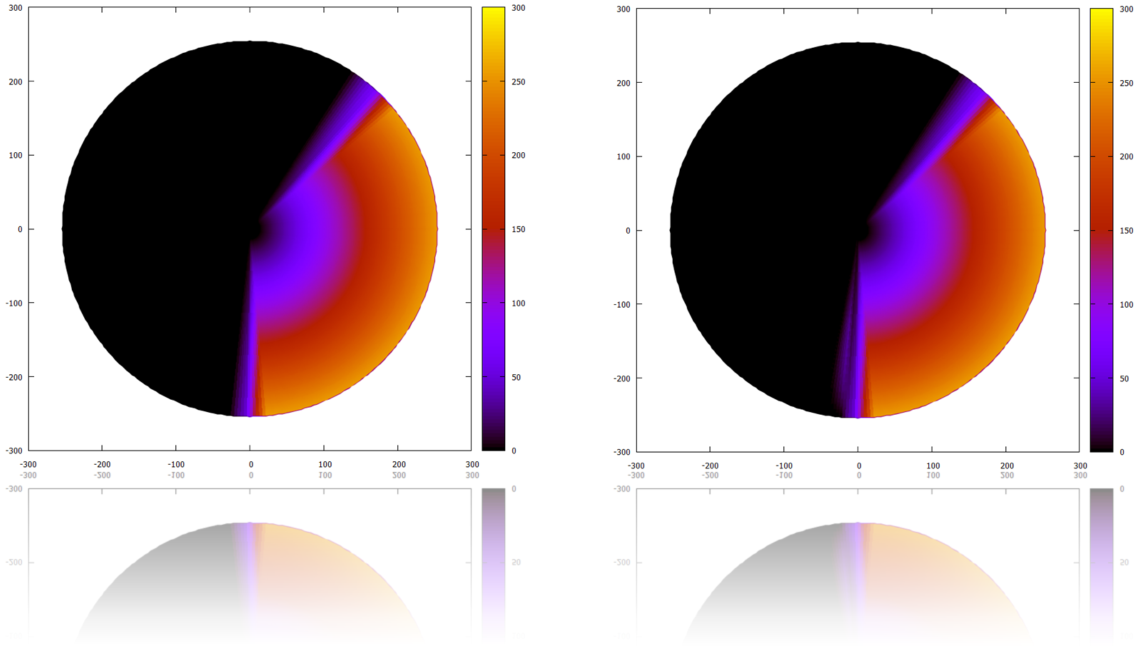


Figure B.17: Adjusted color circle at $l = 50\%$. Range of values is from 0 (black) to max. 255 (orange-yellow)

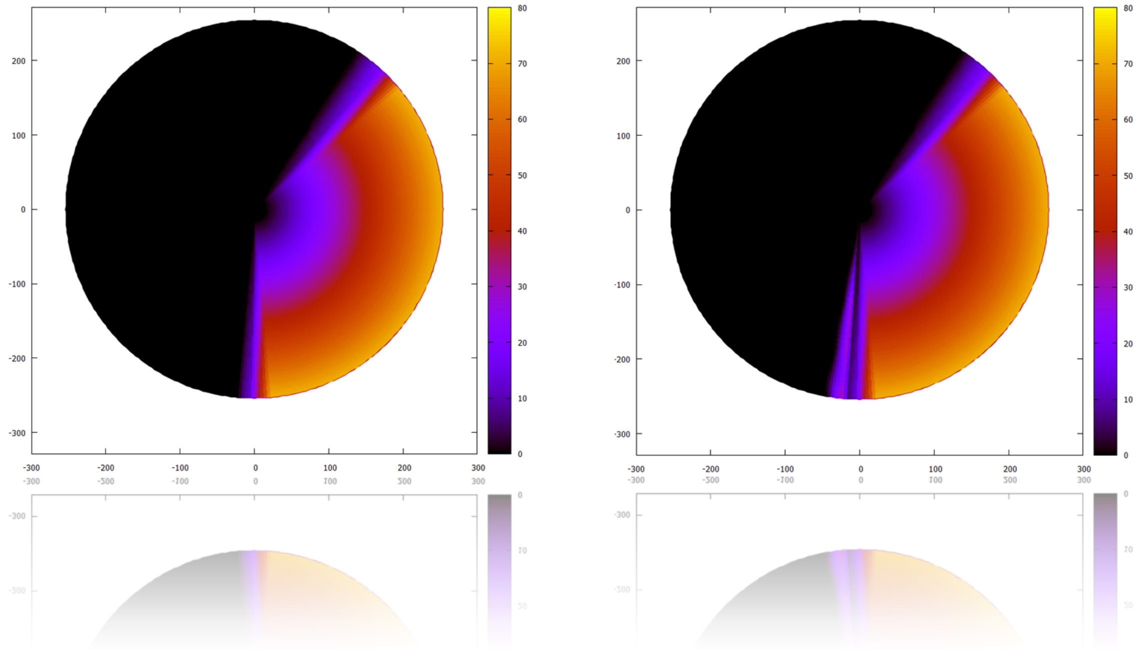


Figure B.18: Adjusted color circle at $l = 70\%$. Range of values is from 0 (black) to max. 80 (orange-yellow)

CIELab

The CIELab color model [286] was developed in 1976 as a system for representing perceptible colors in a device independent and especially uniform way, where the latter entails that equidistant distances within the color model equate to equally perceived color differences, evaluated by a human observer. As it shall be "device independent", the color space should overcome or be independent of limitations which are inherent in specific devices, such as displays or printers.

Because it has been developed to be free from such restrictions, the CIELab model can act as some sort of universal translator between the different color systems native to various devices. Figure B.19 (left) visualizes the CIELab space as a sphere. The L axis refers to the lightness or luminance value. a* represent the shares of the red-green component, i.e., a negative number would be more green, positive is more red. The b* axis refers to the blue-yellow component. Again, going in direction of the negative axis increases the blue share, in the positive axis more the yellow share. However, the real implementation of the CIELab color space does not fill the complete sphere along the L axis, as illustrated in figure B.19 (right).

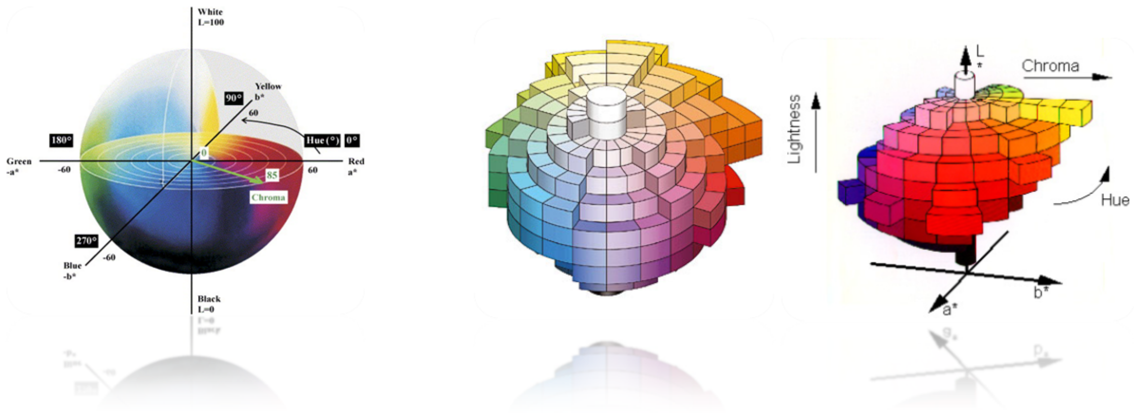


Figure B.19: The CIELab color space. Left: Sphere representation. Picture taken from [105]. Right: Visualization of "equally perceived" regions. Picture from [462]

Bibliography

[1] *The American Heritage Dictionary of the English Language.* Houghton Mifflin Company, 2000. Fourth Edition.

[2] K. Aallouche, H. Albeiriss, R. Zarghoune, J. Arrasvuori, A. Eronen, and J. Holm. Implementation and evaluation of a background music reactive game. In *Proceedings of the 4th Australasian conference on Interactive entertainment*, IE '07, pages 1–6. RMIT University, 2007.

[3] H. Abdi and L. J. Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2:433?459, 2010.

[4] D. Adams, L. Morales, and S. Kurniawan. A qualitative study to support a blind photography mobile application. *ACM 6th International Conference on Pervasive Technologies Related to Assistive Environments (Petra)*, 2013.

[5] E. Adelson. Perceptual organization and the judgement of brightness. *Science*, 262:2042–2044, 2004.

[6] A. Aït Younes, I. Truck, and H. Akdag. Color image profiling using fuzzy sets. *Turkish Journal of Electric Engineering & Computer Sciences*, 13(3):343.

[7] A. Aizerman, E. M. Braverman, and L. I. Rozoner. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837, 1964.

[8] ambiera. irrklang - a cross platform sound library. http://www.ambiera.com/irrklang.

[9] A. Amedi, G. Jacobson, T. Hendler, R. Malach, and E. Zohary. Convergence of visual and tactile shape processing in the human lateral occipital complex. *Cereb. Cortex*, 12(11):1202–1212, 2002.

[10] A. Amedi, R. Malach, T. Hendler, S. Peled, and E. Zohary. Visuo-haptic object-related activation in the ventral visual pathway. *Nature Neuroscience*, 4(3):324–330, 2001.

[11] B. Anderson and J. Winawer. Image segmentation and lightness perception. *Nature*, 434(7029):79–83, 2005.

[12] C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden. Pyramid methods in image processing, 1984.

[13] A. Axen and I. Choi. Investigating geometric data with sound. In *Proceedings of ICAD 4th Meeting of the International Conference on Auditory Display*, June 1996.

[14] T. Bailey and A. Gatrell. *Interactive spatial data analysis.* Longman, Harlow, 1995.

[15] P. Bak, C. Tang, and K. Wiesenfeld. Self-organized criticality - An explanation of 1/f noise. *Physical Review Letters*, 59:381–384, July 1987.

[16] M. Banf. Lernbasierte Bewegungssimulation Hochauflösender Augenmodelle. Master's thesis, Media Systems Group, University of Siegen, 2008.

[17] M. Banf and V. Blanz. Example-based rendering of eye movements. *Computer Graphics Forum*, 28(2):659–666, 2009.

[18] M. Banf and V. Blanz. A modular computer vision sonification model for the visually impaired. In *Proceedings of ICAD 2012 Eighteenth Meeting of the International Conference on Auditory Display*, June 2012.

[19] M. Banf and V. Blanz. Man made structure detection and verification of object recognition in images for the visually impaired. *roceedings of Mirage 2013, 6th International Conference on Computer Vision / Computer Graphics Collaboration Techniques and Applications, in Cooperation with Eurographics Association*, 2013.

[20] M. Banf and V. Blanz. Sonification of images for the visually impaired using a multi-level approach. *Augmented Human Conference '13 in cooperation with ACM SIGCHI*, 2013.

[21] D. Barash. A fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(6):844–847, 2002.

[22] D. Barina. Gabor wavelets in image processing. Technical report, Faculty of Information Technology, Brno University of Technology, 2009.

[23] S. Barrass. *Auditory Information Design.* PhD thesis, Australian National University, 1997.

[24] E. Bates and D. Fitzpatrick. Spoken mathematics using prosody, earcons and spearcons. In K. Miesenberger, J. Klaus, W. Zagler, and A. Karshmer, editors, *Computers Helping People with Special Needs*, volume 6180 of *Lecture Notes in Computer Science*, pages 407–414. Springer Berlin Heidelberg, 2010.

[25] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features. *Comput. Vis. Image Underst.*, 110(3):346–359, 2008.

[26] U. Beer. *Was Farben uns verraten.* Stuttgart Kreuz Verlag, 1992.

[27] A. Bell. Hearing: Travelling wave or resonance? *PLoS Biol*, 2(10):e337, 10 2004.

[28] R. Bencina. Real-time audio programming 101: time waits for nothing. http://www.rossbencina.com/code/real-time-audio-programming-101-time-waits-for-nothing.

[29] J. P. Bigham, C. Jayant, H. Ji, G. Little, A. Miller, R. C. Miller, R. Miller, A. Tatarowicz, B. White, S. White, and T. Yeh. Vizwiz: nearly real-time answers to visual questions. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, UIST '10, pages 333–342, 2010.

[30] C. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1996.

[31] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics).* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[32] M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger. Robust anisotropic diffusion. *IEEE Trans. Image Processing*, 7:421–432, 1998.

[33] J. Blanchette and M. Summerfield. *C++ GUI Programming with Qt 4*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2006.

[34] M. M. Blattner, D. A. Sumikawa, and R. Greenberg. Earcons and icons: Their structure and common design principles. In *Human Computer Interaction*, volume 4(1).

[35] J. Blauert. *Spatial Hearing.* MIT Press, MA, 1983.

[36] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[37] S. Bly. *Sound and Computer Information Presentation.* PhD thesis, University of California, 1982.

[38] S. Bochkanov. Alglib, 2010. http://mloss.org/software/view/231/.

[39] G. Bologna, B. Deville, J. Diego Gomez, and T. Pun. Toward local and global perception modules for vision substitution. *Neurocomput.*, 74(8):1182–1190, Mar. 2011.

[40] G. Bologna, B. Deville, and T. Pun. On the use of the auditory pathway to represent image scenes in real-time. *Neurocomput.*, 72(4-6):839–849, Jan. 2009.

[41] F. L. Bookstein. Principal warps: thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585, 1989.

[42] J. Borenstein. The navbelt - a computerized multi-sensor travel aid for active guidance of the blind. In *Proceedings of the Fifth Annual CSUN Conference on Technology and Persons With Disabilities*, 1990.

[43] I. Borg and P. J. F. Groenen. *Modern multidimensional scaling : theory and applications.* Springer, 2005.

[44] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, COLT '92, pages 144–152, 1992.

[45] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(9):1124–1137, 2004.

[46] Y. Boykov and O. Veksler. Graph Cuts in Vision and Graphics: Theories and Applications. In N. Paragios, Y. Chen, and O. Faugeras, editors, *Handbook of Mathematical Models in Computer Vision*, chapter 5, pages 79–96. 2006.

[47] Y. Boykov, O. Veksler, and R. Zabih. Efficient approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2001.

[48] R. Bracewell. *The Fourier Transform and Its Applications.* New York: McGraw-Hill, 1999. Third Edition.

[49] G. R. Bradski. *Learning OpenCV: Computer Vision in C++ with the OpenCV Library.* O'Reilly Media, 2012. second edition.

[50] A. S. Bregman. *Auditory Scene Analysis: The Perceptual Organization of Sound.* Cambridge, MA: MIT Press, 1990.

[51] S. Brewster. Using non-speech sound to overcome information overload. volume 17, pages 179–189, 1997.

[52] S. A. Brewster. *Providing a structured method for integrating non-speech audio into human-computer interfaces.* PhD thesis, 1994.

[53] R. Bridson. Fast poisson disk sampling in arbitrary dimensions. In *ACM SIGGRAPH 2007 sketches*, SIGGRAPH '07. ACM, 2007.

[54] J. Briet and P. Harremoes. Properties of classical and quantum Jensen-Shannon divergence. *Physical Review A: Atomic, Molecular and Optical Physics*, 79(5), May 2009.

[55] L. Brown and S. Brewster. Design guidelines for audio presentation of graphs and tables. In *Design guidelines for audio presentation of graphs and tables*, 2003.

[56] M. L. Brown, S. L. Newsome, and E. P. Glinert. An experiment into the use of auditory cues to reduce visual workload. pages 339–346, 1989.

[57] R. Brown. A brief account of microscopical observations made in the months of june, july and august, 1827, on the particles contained in the pollen of plants; and on the general existence of active molecules in organic and inorganic bodies. *Phil. Mag.*, 4:161–173, 1828.

[58] R. Brown. *Introduction to Random Signal Analysis and Kalman Filtering.* John Wiley and Sons, 1983.

[59] J. Brownlee. *Clever Algorithms: Nature-Inspired Programming Recipes.* Lulu Enterprises, 2011.

[60] M. D. Buhmann and M. D. Buhmann. *Radial Basis Functions.* Cambridge University Press, New York, NY, USA, 2003.

[61] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.*, 2(2):121–167, 1998.

[62] J. B. Burns, A. R. Hanson, and E. M. Riseman. Extracting straight lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:425–455, 1986.

[63] P. J. Burt and E. H. Adelson. A multiresolution spline with application to image mosaics. *ACM Trans. Graph.*, 2(4):217–236, Oct. 1983.

[64] P. J. Burt, Edward, and E. H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31:532–540, 1983.

[65] W. Buxton, S. A. Bly, S. P. Frysinger, D. Lunney, D. L. Mansur, and J. J. Mezrich. Communicating with sound. pages 115–119, 1985.

[66] J. Caivano. *Color and Sound: Physical and Psychophysical Relations.* J. Wiley, 1994.

[67] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–698, 1986.

[68] M. Carandini. What simple and complex cells compute. *The Journal of Physiology*, 577(2):463–466, 2006.

[69] J. R. Cavanaugh, W. Bair, J. A. Movshon, J. R, W. Bair, and J. A. Movshon. Nature and interaction of signals from the receptive field center and surround in macaque v1 neurons. *J Neurophysiol*, pages 2530–2546, 2002.

[70] R. Chandra, L. Dagum, D. Kohr, D. Maydan, J. McDonald, and R. Menon. *Parallel programming in OpenMP*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001.

[71] C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):1–27, 2011.

[72] B. Chapman, G. Jost, and R. v. d. Pas. *Using OpenMP: Portable Shared Memory Parallel Programming (Scientific and Engineering Computation)*. The MIT Press, 2007.

[73] G. Chartrand. *Introductory Graph Theory*. Dover, 1985.

[74] B. B. Chaudhuri and N. Sarkar. Texture segmentation using fractal dimension. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(1):72–77, Jan. 1995.

[75] Q. Chen, X. Chen, and Y. Wu. The combining kernel principal component analysis with support vector machines for time series prediction model. In *Intelligent Information Technology Application, 2008. IITA '08. Second International Symposium on*, volume 2, pages 90 –94, 2008.

[76] Y. Q. Chen and G. Bi. On texture classification using fractal dimension. *IJPRAI*, 13:929–943, 1999.

[77] L. Chittka and A. Brockmann. Perception space the final frontier. *PLoS Biol*, 3(4):e137, 04 2005.

[78] J. Chowning. The synthesis of complex audio spectra by means of frequency modulation. volume 21, pages 526–534. Journal of the Audio Engineering Society, 1973.

[79] R. Clausius. *On the Motive Power of Heat, and on the Laws which can be deduced from it for the Theory of Heat*. Annalen der Physik (Band 79), Dover, 1960, 1850.

[80] D. Conway. An experimental comparison of three natural language colour naming models. In *Proc. East-West International Conference on Human-Computer Interactions*, pages 328–339, 1992.

[81] P. R. Cook. Sound synthesis for auditory display. In T. Hermann, A. Hunt, and J. G. Neuhoff, editors, *The Sonification Handbook*, chapter 9, pages 197–235. Logos Publishing House, Berlin, Germany, 2011.

[82] P. R. Cook and G. P. Scavone. The synthesis toolkit in c++ stk. https://ccrma.stanford.edu/software/stk/.

[83] P. R. Cook and G. P. Scavone. The synthesis toolkit (stk), 1999.

[84] W. Cook, W. Cunningham, W. Pulleyblank, and A. Schrijver. *Combinatorial Optimization*. John Wiley & Sons, 1998.

[85] C. E. Cormen, T. H. nand Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press and McGraw-Hill, 2001. Second Edition.

[86] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.

[87] T. M. Cover and J. A. Thomas. *Elements of information theory.* Wiley-Interscience, New York, NY, USA, 1991.

[88] T. Cox and M. Cox. *Multidimensional Scaling.* Chapman & Hall, 1994.

[89] C. D. Creelman. Human discrimination of auditory duration. *Journal of the Acoustical Society of America*, 34(5):582–593, 1962.

[90] J. Cronly-Dillon et al. Blind subjects analyse photo images of urban scenes encoded in musical form. In *Investigative Ophthalmology & Visual Science*, 2000.

[91] J. Cronly-Dillon, K. C. Persaud, and R. Blore. Blind subjects construct conscious mental images of visual scenes encoded in musical form. In *Investigative Ophthalmology & Visual Science*, 2000.

[92] J. R. Cronly-Dillon, K. C. Persaud, and R. Gregory. The perception of visual images encoded in musical form: a study in cross-modality information transfer. In *Proceedings Of The Royal Society Of London Series B-Biological Sciences*, pages 2427–2433, 1999.

[93] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.

[94] C. Cullen and E. Coyle. Rhythmic parsing of sonified dna and rna sequences. In *Irish Signals Systems Conference*, 2003.

[95] R. E. Cytowic. *Synesthesia: A Union of the Senses.* Cambridge, Massachusetts: MIT Press, second edition edition, 2002.

[96] R. E. Cytowic and D. M. Eagleman. *Wednesday is Indigo Blue: Discovering the Brain of Synesthesia.* Cambridge, Massachusetts: MIT Press, 2009.

[97] G. D. Acoustical quanta and the theory of hearing. *Nature*, 159:591–594, 1947.

[98] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *In CVPR*, pages 886–893, 2005.

[99] J. G. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *JOSA A,*, 2:1160–1169, 1985.

[100] J. G. Daugman. Complete discrete 2-d gabor transforms by neural networks for image analysis and compression. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36:1169?1179, 1988.

[101] E. R. Davies. Introduction to texture analysis. 2009.

[102] M. Davis, A. Khotanzad, D. Flamig, and S. Harms. A physics-based coordinate transformation for 3-d image matching. *Medical Imaging, IEEE Transactions on*, 16(3):317 –328, 1997.

[103] R. L. De Valois and G. H. Jacobs. Primate color vision. *Science*, 162(3853):533–540, 1968.

[104] S. Dehaene, F. Pegado, L. W. Braga, P. Ventura, G. N. Filho, A. Jobert, G. Dehaene-Lambertz, R. Kolinsky, J. Morais, and L. Cohen. How learning to read changes the cortical networks for vision and language. *Science*, 330(6009):1359–1364, 2010.

[105] A. V. der Salm, M. Martnez, G. Flik, and S. W. Bonga. Effects of husbandry conditions on the skin colour and stress response of red porgy, pagrus pagrus. *Aquaculture*, 241:371 – 386, 2004.

[106] F. Devernay, F. Devernay, P. Robotique, and P. Robotvis. A non-maxima suppression method for edge detection with sub-pixel accuracy. Technical report, INRIA Research Rep. 2724, SophiaAntipolis, 1995.

[107] F. Diebold. *Elements of Forecasting.* Thomson/South-Western, 3. ed edition, 2003.

[108] P. Diggle. *Statistical Analysis of Spatial Point Patterns.* Academic Press, New York, 2003. Second Edition.

[109] M. B. Dillencourt, H. Samet, and M. Tamminen. A general approach to connected-component labeling for arbitrary image representations. *J. ACM*, 39(2):253–280, 1992.

[110] A. Disley, D. Howard, and A. Hunt. Practical synthesis control by timbral adjectives. *Proceedings of the Institute of Acoustics*, 30(2), 2008.

[111] A. C. Disley and D. M. Howard. Spectral correlates of timbral semantics relating to the pipe organ. 46, 2004.

[112] A. C. Disley, D. M. Howard, and A. D. Hunt. Timbral description of musical instruments. 2006.

[113] A. C. Disley, D. M. Howard, and A. D. Hunt. Timbral adjectives for the control of a music synthesizer. 2007.

[114] C. Dodge and T. A. Jerse. *Computer Music.* New York: Schirmer Books, 1997.

[115] M. M. Dodson and S. Kristensen. Fractal geometry and applications: a jubilee of benoit mandelbrot. *Proceedings of Symposia in Pure Mathematics*, pages 305–347, 2004.

[116] F. Dombois. Using audification in planetary seismology. pages 227–230, 2001.

[117] G. Donato and S. Belongie. Approximate thin plate spline mappings. In *Proceedings of the 7th European Conference on Computer Vision-Part III*, ECCV '02, pages 21–31, London, UK, UK, 2002. Springer-Verlag.

[118] A. Downey. *Think Complexity.* O'Reilly Media, 2012.

[119] J. M. H. du Buf. Improved grating and bar cell models in cortical area v1 and texture coding. *Image and Vision Computing*, 25(6):873–882, 2007.

[120] R. C. Dubes and A. K. Jain. Random field models in image analysis. *Journal of Applied Statistics*, 16(2):131–164, 1989.

[121] J. Duchon. Splines minimizing rotation-invariant semi-norms in Sobolev spaces Constructive Theory of Functions of Several Variables. volume 571 of *Lecture Notes in Mathematics*, pages 85–100. Springer Berlin / Heidelberg, 1977.

[122] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* John Wiley and Sons, 2001. second edition.

[123] D. Dunn and W. Higgins. Optimal gabor filters for texture segmentation. *IEEE Transactions on Image Processing*, 4(7):947–964, 1995.

[124] D. Dunn, W. Higgins, and J. Wakeley. Texture segmentation using 2-d gabor elementary functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:130–149, 1994.

[125] F. Durand and J. Dorsey. Fast bilateral filtering for the display of high-dynamic-range images. In *SIGGRAPH*, pages 257–266, 2002.

[126] P. Duvall, J. Keesling, and A. Vince. The hausdorff dimension of the boundary of a self-similar tile. *Journal of the London Mathematical Society*, 61:748–760, 5 2000.

[127] D. M. Eagleman. Visual illusions and neurobiology. *Nature Reviews Neuroscience*, 2(12):920–926, Dec. 2001.

[128] A. Edwards. Soundtrack: An auditory interface for blind users. In *Human-Computer Interaction*, volume 4(1), pages 45–66, 1989.

[129] A. Edwards and R. Stevens. Mathematical representations: Graphs, curves and formulas. In *Non- Visual Human-Computer Interactions: Prospects for the visually handicapped*, pages 181–194, 1993.

[130] J. Edworthy. Does sound help us to work better with machines? a commentary on rautenberg?s paper 'about the importance of auditory alarms during the operation of a plant simulator'. In *Interacting with Computers*, volume 10, pages 401 – 409, 1998.

[131] J. Elonen. Interactive thin plate spline (tps) demo/editor for opengl+glut. http://www.http://elonen.iki.fi/archives.html.

[132] C. Enroth-Cugell and J. G. Robson. The contrast sensitivity of retinal ganglion cells of the cat. *The Journal of physiology*, 187(3):517–552, Dec. 1966.

[133] EpicGames. Unreal engine 3. http://www.unrealengine.com/.

[134] R. Ethington and B. Punch. SeaWave: A system for musical timbre description. *Computer Music Journal*, 18, 1994.

[135] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results. http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html.

[136] A. Ezust and P. Ezust. *An Introduction to Design Patterns in C++ with Qt 4 (Bruce Perens Open Source)*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2006.

[137] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 99, 2012.

[138] G. Farin. *Curves and Surface for Computer Aided Geometric Design.* Boston: Academic Press, 1996. Fourth Edition.

[139] G. Farin and D. Hansford. *Lineare Algebra: ein geometrischer Zugang.* Springer, 2003.

[140] H. Federer. *Geometric Measure Theory.* New York: Springer-Verlag, 1969.

[141] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. *CVPR*, pages 524–531, 2005.

[142] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

[143] W. Feng, Y. Zhao, and J. Deng. Application of svm based on principal component analysis to credit risk assessment in commercial banks. In *Intelligent Systems, 2009. GCIS '09. WRI Global Congress on*, volume 4, pages 49 –52, 2009.

[144] I. Fine, A. R. Wade, A. A. Brewer, M. G. May, D. F. Goodman, G. M. Boynton, B. A. Wandell, and D. I. A. MacLeod. Long-term deprivation affects visual perception and cortex. *Nat Neurosci*, 6(9):915–6, 2003.

[145] G. D. Finlayson, M. S. Drew, and C. Lu. Entropy minimization for shadow removal. *Int. J. Comput. Vision*, 85(1):35–57, 2009.

[146] G. D. Finlayson, S. D. Hordley, C. Lu, and M. S. Drew. Removing shadows from images. In *In ECCV 2002: European Conference on Computer Vision*, pages 823–836, 2002.

[147] W. T. Fitch and G. Kramer. *Sonifying the body electric: Superiority of an auditory over a visual display in a complex, multivariate system.* SFI studies in the sciences of complexity. Addison Wesley Longman, 1992. In Kramer, G (ed) 1994. Auditory display: Sonification, Audification, and Auditory Interfaces. Santa Fe Institute Studies in the Sciences of Complexity, Proceedings, Volume XVIII. Reading, MA: Addison Wesley Publishing Company.

[148] R. Fletcher. *Practical methods of optimization; (2nd ed.).* Wiley-Interscience, New York, NY, USA, 1987.

[149] J. H. Flowers, D. C. Buhman, and K. D. Turnage. Cross-modal equivalence of visual and auditory scatterplots for exploring bivariate data samples. volume 39(3), pages 341–351, 1997.

[150] J. H. Flowers and T. A. Hauer. The ear's versus the eye's potential to assess characteristics of numeric data: Are we too visuocentric? volume 24(2), pages 258–264, 1992.

[151] T. W. Forbes. Auditory signals for instrument flying. In *J. Aeronautical Soc.*, pages 255–258, 1946.

[152] L. Ford and D. Fulkerson. *Flows in Networks.* Princeton University Press, 1962.

[153] E. E. Fournier d'Albe. On a type-reading optophone. In *Proceedings of the Royal Society of London*, 1914.

[154] K. Franze, J. Grosche, S. N. Skatchkov, S. Schinkinger, C. Foja, D. Schild, O. Uckermann, K. Travis, A. Reichenbach, and J. Guck. Müller cells are living optical fibers in the vertebrate retina. *Proceedings of the National Academy of Sciences*, 104(20):8287–8292, 2007.

[155] J. Frasnelli, O. Collignon, P. Voss, and F. Lepore. Crossmodal plasticity in sensory loss. *Prog Brain Res*, 191, 2011.

[156] D. Gabor. Theory of communication. *Journal of the Institution of Electrical Engineers ? Part III: Radio and Communication Engineering*, 93:429?457, 1946.

[157] C. Galamhos, J. Matas, and J. Kittler. Progressive probabilistic hough transform for line detection. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 1, 1999.

[158] B. Galerne. *Stochastic image models and texture synthesis.* PhD thesis, Ecole Normale Suprieure de Cachan, 2010.

[159] W. R. Garner and R. L. Gottwald. The perception and learning of temporal patterns. volume 20(2), 1968.

[160] W. W. Gaver. Auditory icons: using sound in computer interfaces. *Human Computer Interaction*, 2(2):167–177, June 1986.

[161] H. Gekeler. *DuMont's Handbuch der Farben.* DuMont, Koeln, 1988.

[162] K. Giannakis and M. Smith. Auditory-visual associations for music compositional processes: A survey. 2000.

[163] K. Giannakis and M. Smith. Imaging soundscapes: Identifying cognitive associations between audatory and visual dimensions. In *Musical Imagery. Swets & Zeitlinger (2001) 161?179*, 2001.

[164] A. L. Gilchrist. When does perceived lightness depend on perceived spatial arrangement? *Perception & Psychophysics*, 28(6):527–538, 1980.

[165] A. L. Gilchrist. Lightness contrast and failures of constancy: A common explanation. *Perception & Psychophysics*, pages 415–424, 1988.

[166] D. Gillan, P. Barraza, A. Karshmer, and S. Pazuchanics. Cognitive analysis of equation reading: Application to the development of the math genie. In K. Miesenberger, J. Klaus, W. Zagler, and D. Burger, editors, *Computers Helping People with Special Needs*, volume 3118 of *Lecture Notes in Computer Science*, pages 628–628. Springer Berlin Heidelberg, 2004.

[167] M. Gipp, G. Marcus, N. Harder, A. Suratanee, K. Rohr, R. Knig, and R. Mnner. Haralick's texture features computed by GPUs for biological applications, 2009.

[168] F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 7:219–269, 1995.

[169] R. B. Girshick, P. F. Felzenszwalb, and D. McAllester. Discriminatively trained deformable part models, release 3. http://people.cs.uchicago.edu/ rbg/latent/.

[170] F. Glover and M. Laguna. *Tabu Search.* Kluwer Academic Publishers, Norwell, MA, USA, 1997.

[171] J. Goethe. Farbenkreis, aquarellierte federzeichnung von goethe, 1809. Freies Deutsches Hochstift - Goethe-Museum im Goethe-Haus, Frankfurt.

[172] A. V. Goldberg and R. E. Tarjan. A new approach to the maximum-flow problem. *J. ACM*, 35(4):921–940, Oct. 1988.

[173] T. H. Goldsmith. Optimization, constraint, and history in the evolution of eyes. *Q Rev Biol*, 65(3):281–322, Sept. 1990.

[174] E. Goldstein. *Sensation and Perception.* C. L. Emea, 2009.

[175] J. L. Gonzlez-Mora et al. Seeing the world by hearing: Virtual acoustic space (vas) a new space perception system for blind people. In *Touch Blindness and Neuroscience*, pages 371–383, 2004.

[176] J. Gossmann. Towards an auditory representation of complexity. pages 264–268, 2005.

[177] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1–8, 2009.

[178] A. Gounaropoulos and C. Johnson. Synthesising timbres and timbre-changes from adjectives/adverbs. In *Proceedings of the 2006 international conference on Applications of Evolutionary Computing*, EuroGP'06, pages 664–675. Springer-Verlag, 2006.

[179] P. Gray. *Psychology.* Worth Publishers, 2001.

[180] R. Gregory. Seeing after blindness. *Nature Neuroscience*, 6:909–910, 2003.

[181] D. Greig, B. Porteous, and A. Seheult. Exact Maximum A Posteriori Estimation for Binary Images. *Royal Journal on Statistical Society*, 51(2):271–279, 1989.

[182] J. M. Grey. An exploration of musical timbre. Master's thesis, Stanford University, Stanford, California, 1975.

[183] R. Grompone von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. LSD: a Line Segment Detector. *Image Processing On Line*, 2012, 2012.

[184] F. Grond and J. Berger. Parameter mapping sonification. In T. Hermann, A. Hunt, and J. G. Neuhoff, editors, *The Sonification Handbook*, chapter 15, pages 363–397. Logos Publishing House, Berlin, Germany, 2011.

[185] F. Grond, T. Drossard, and T. Hermann. Sonicfunction: Experiments with a function browser for the visually impaired. In *Proceedings of the 16th International Conference on Auditory Display*, Washington D.C., 2010. ICAD.

[186] F. Grond, S. Janssen, S. Schirmer, and T. Hermann. Browsing rna structures by interactive sonification. In R. Bresin, editor, *Proceedings of the 3rd Interactive Sonification Workshop (ISon 2010)*, Stockholm, 04 2010. ISon, ISon.

[187] M. Haindl. Texture synthesis. *CWI Quart*, 4:305–331, 1991.

[188] R. W. Hall. *The sound of numbers.* Department of Mathematics and Computer Science, Saint Joseph's University, 2000.

[189] D. E. Hannula, D. J. Simons, and N. J. Cohen. Imaging implicit perception: promise and pitfalls. *Nature reviews. Neuroscience*, 6(3):247–55, 2005.

[190] R. M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804, 1979.

[191] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural Features for Image Classification. *Systems, Man and Cybernetics, IEEE Transactions on*, (6):610–621, 1973.

[192] C. Harris and M. Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988.

[193] J. W. Harris and H. Stocker. *Handbook of Mathematics and Computational Science.* New York: Springer-Verlag, 1998.

[194] K. Hayashi and N. Munakata. Gene music; tonal assignments of bases and amino acids. In *Visualizing Biological Information*, pages 72–83. World Scientific.

[195] K. Hayashi and N. Munakata. Basically musical. In *Nature*, volume 310(96), 1984.

[196] C. Hayward. Listening to the earth sing. In edited by G. Kramer, editor, *In Auditory Display. Sonification, Audification, and Auditory Interfaces*, pages 369–404. Reading, Massachusetts, USA: Addison-Wesley, June 1994.

[197] X. He and R. S. Zemel. Learning hybrid models for image annotation with partially labeled data. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, pages 625–632. MIT Press, 2008.

[198] X. He, R. S. Zemel, and M. A. Carreira-Perpiñán. Multiscale conditional random fields for image labeling. In *Proceedings of the 2004 IEEE computer society conference on Computer vision and pattern recognition*, CVPR'04, pages 695–703, 2004.

[199] heartsoffireproductions.com. Sound engineers survival guide. http://heartsoffireproductions.com/.

[200] H. L. F. Helmholtz. *On the Sensations of Tone as a Physiological Basis for the Theory of Music.* New York: Dover Publications (1954), 1877. 4th Edition.

[201] J. Hereford and W. Winn. Non-speech sound in human-computer interaction: A review and design guidelines. volume 11, pages 211–233, 1994.

[202] E. Hering. *Outlines of a Theory of the Light Sense.* Harvard University Press, Cambridge, Mass., 1964.

[203] T. Hermann. Thomas hermann's research on sonification, data mining and ambient intelligence. http://sonification.de/.

[204] T. Hermann. *Sonification for Exploratory Data Analysis.* PhD thesis, Bielefeld University, Bielefeld, Germany, 02 2002.

[205] T. Hermann. Taxonomy and definitions for sonification and auditory display. Paris, France, 2008. inproceedings.

[206] T. Hermann, A. Hunt, and J. G. Neuhoff, editors. *The Sonification Handbook.* Logos Publishing House, Berlin, Germany, 2011.

[207] T. Hermann, A. V. Nehls, F. Eitel, T. Barri, and M. Gammel. Tweetscapes - real-time sonification of twitter data streams for radio broadcasting. In *Proceedings of the 18th International Conference on Auditory Display (ICAD)*, 2012.

[208] T. Hermann and H. Ritter. Listen to your data: Model-based sonification for data analysis. In *Advances in intelligent computing and multimedia systems*, pages 189–194, 1999.

[209] P. Herrera, G. Peeters, and S. Dubnov. Automatic classification of musical instrument sounds. *Journal of New Music Research*, 32, 2003.

[210] T. Hofmann. Probabalistic Latent Semantic Analysis. 1999.

[211] D. Hofstadter. *Gödel, Escher, Bach: An Eternal Golden Braid.* Basic Books, 1979.

[212] D. Hong, S. Kimmel, R. Boehling, N. Camoriano, W. Cardwell, G. Jannaman, A. Purcell, D. Ross, and E. Russel. Development of a semi-autonomous vehicle operable by the visually-impaired. In *Multisensor Fusion and Integration for Intelligent Systems, 2008. MFI 2008. IEEE International Conference on*, pages 539–544, Aug. 2008.

[213] D. Howard, A. Disley, and A. Hunt. Towards a music synthesizer controlled by timbral adjectives. 2007.

[214] T. Hsu and K.-J. Hu. Multi-resolution texture segmentation using fractal dimension. In *Proceedings of the 2008 International Conference on Computer Science and Software Engineering - Volume 06*, CSSE '08, pages 201–204, 2008.

[215] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160:106–154, 1962.

[216] D. H. Hubel and T. N. Wiesel. Receptive fields and functional architecture in two nonstriate visual areas (18 and 19) of the cat. *Journal of Neurophysiology*, 28:229–289, 1965.

[217] A. Hunt and T. Hermann. The importance of interaction in sonification. In *ICAD*. International Community for Auditory Display, 2004.

[218] A. Hunt and T. Hermann. Interactive sonification. In T. Hermann, A. Hunt, and J. G. Neuhoff, editors, *The Sonification Handbook*, chapter 11, pages 273–298. Logos Publishing House, Berlin, Germany, 2011.

[219] J. Illingworth and J. Kittler. A survey of the hough transform. In *Computer Vision, Graphics and Image Processing*, volume 44, pages 87–116, 1988.

[220] A. N. S. Institute, editor. *American national psychoacoustical terminology.* American Standards Association, 1973.

[221] E. Jacobson, W. C. Granville, and C. Foss. *Color Harmony Manual.* Container Corporation of America, Chicago, 1948.

[222] B. Jähne. *Digital Image Processing: Concepts, Algorithms and Scientific Applications.* Springer, 2002. 5th rev. and extended ed.

[223] A. K. Jain and F. Farrokhnia. Unsupervised texture segmentation using gabor filters. *Pattern Recognition*, 4:1167–1186, 1991.

[224] Z. Jaunmahomed and M. Chait. The timing of change detection and change perception in complex acoustic scenes. *Front Psychol*, 3, 2012.

[225] C. Jayant, H. Ji, S. White, and J. P. Bigham. Supporting blind photography. In *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility*, ASSETS '11, pages 203–210, 2011.

[226] A. J. Jerri. Correction to "The Shannon sampling theorem&#8212;Its various extensions and applications: A tutorial review". *Proceedings of the IEEE*, 67(4), 1979.

[227] C. G. Johnson and A. Gounaropoulos. Timbre interfaces using adjectives and adverbs. In *Proceedings of the 2006 conference on New interfaces for musical expression*, NIME '06, pages 101–102, 2006.

[228] I. Jolliffe. *Principal Component Analysis.* Springer, Heidelberg, 2002. Second Edition.

[229] H. E. Jones, K. L. Grieve, W. Wang, A. M. Sillito, J. Neurophysiol, S. P. Macevoy, T. D. Hanks, M. A. Paradiso, G. A. Orban, P. Rev, M. Gur, D. M. Snodderly, Z. m. Shen, W. f. Xu, C. y. Li, S. c. Yen, J. Baker, C. M. Gray, H. E. Jones, K. L. Grieve, W. Wang, and A. M. Sillito. Surround suppression in primate v1. *J. Neurophysiol*, 86:2011–2028, 2001.

[230] J. Jones and L. Palmer. An evaluation of two-dimensional gabor filter model of simple receptive fields in cat strait cortex. *Journal of Neurophysiology*, 58:1233?1258, 1987.

[231] M. Jones et al. *Music Perception, Springer Handbook of Auditory Research.* Springer Science Business Media, LLC, 2010.

[232] M. R. Jones and M. Boltz. Dynamic attending and responses to time. *Psychological Review*, 96:459–491, 1989.

[233] W. Jones. *Beginning DirectX 10 Game Programming.* Cengage Learning Emea, 2007.

[234] B. D. Josephson. A trans-human source of music? *New Directions in Cognitive Science, Finnish Artificial Intelligence Society*, page 280?285, 1995.

[235] E. Kabisch, F. Kuester, and S. Penny. Sonic panoramas: experiments with interactive landscape image sonification. In *Proceedings of the 2005 international conference on Augmented tele-existence*, ICAT '05, pages 156–163, 2005.

[236] H. M. Kamel, P. Roth, and R. R. SInha. Graphics and user's exploration via simple sonics (guess): Providing interrelational representation of objects in a non-visual environment. In *Proceedings of 7th International Conference on Auditory Display*, pages 261–266, 2001.

[237] S. J. Kandel ER and J. TM. *Principles of Neural Science.* McGraw Hill, New York, 2000. 4th edition.

[238] V. Kandinsky. *The Effect of Color.* 1911.

[239] G. Kanizsa. Subjective contours. *Sci Am*, 234(4):48–52, 1976.

[240] I. Kant. *Critique of Pure Reason.* The Cambridge Edition of the Works of Immanuel Kant. Cambridge University Press, 1998. Translated by Paul Guyer and Allen W. Wood.

[241] D. Karatzas and A. Antonacopoulos. Colour text segmentation in web images based on human perception. *Image Vision Comput.*, 25(5):564–577, May 2007.

[242] Y. A. Karayiannis and T. Stouraitis. Texture classification using the fractal dimension as computed in a wavelet decomposed image. In *IEEE Work. Nonlin. Sign. Image Proc*, pages 186–189, 1995.

[243] K. Karplus and A.Strong. Digital synthesis of plucked-string and drum timbres. *Computer Music Journal*, 7(2):43–55, 1983.

[244] E. Kennard. Zur Quantenmechanik einfacher Bewegungstypen. *Zeitschrift für Physik*, 44:326–352, 1927.

[245] J. Kepler. *Mysterium Cosmographicum.* Fritz Krafft, Marixverlag, 2005. Nachdruck erhältlich unter: Johannes Kepler - Was die Welt im Innersten zusammenhält. Antworten aus Schriften von Johannes Kepler.

[246] J. Kepler. *Harmonice Mundi.* Oldenbourg Verlag, 2006. Unveränderter Nachdruck der Ausgabe von 1939. Übersetzt und eingeleitet von Max Caspar.

[247] W. F. Kern and J. R. Bland. *Solid Mensuration with Proofs.* New York: Wiley, 1948.

[248] R. Kindermann and J. L. Snell. *Markov Random Fields and Their Applications.* AMS, 1980.

[249] D. Knuth. *The Art of Computer Programming, Volume 1: Fundamental Algorithms.* Adsison-Wesley, 1997. Third Edition.

[250] K. Koffka. *Principles of Gestalt Psychology.* Harcourt, Brace and World, New York, 1935.

[251] P. Kohli and P. H. S. Torr. Effciently solving dynamic markov random fields using graph cuts. In *Proceedings of the Tenth IEEE International Conference on Computer Vision - Volume 2*, ICCV '05, pages 922–929, 2005.

[252] P. Kohli and P. H. S. Torr. Dynamic graph cuts for efficient inference in markov random fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(12):2079–2088, Dec. 2007.

[253] P. Koirala, M. Hauta-Kasari, and J. Parkkinen. Highlight removal from single image. In J. Blanc-Talon, W. Philips, D. C. Popescu, and P. Scheunders, editors, *ACIVS*, volume 5807 of *Lecture Notes in Computer Science*, pages 176–187. Springer, 2009.

[254] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? In *Proceedings of the 7th European Conference on Computer Vision-Part III*, ECCV '02, pages 65–81, London, UK, UK, 2002. Springer-Verlag.

[255] F. Korc and W. Foerstner. Approximate parameter learning in conditional random fields: An empirical investigation. In *Proceedings of DAGM symp.on PR*, pages 11–20, 2008.

[256] G. Kramer. *An introduction to auditory display.* SFI studies in the sciences of complexity. Addison Wesley Longman, 1992. In Auditory display: Sonification, Audification, and Auditory Interfaces. Santa Fe Institute Studies in the Sciences of Complexity, Proceedings, Volume XVIII. Reading, MA: Addison Wesley Publishing Company.

[257] G. Kramer, B. Walker, T. Bonebright, P. Cook, J. Flower, N. Miner, and J. Neuhoff. Sonification report: Status of the field and research agenda. Technical report, International Community for Auditory Display, 1999.

[258] M. S. Krimphoff, J. and S. Winsberg. Caractrisation du timbre des sons complexes. ii: Analyses acoustiques et quantification psychophysique. In *Journal de Physique*, pages 625–628, 1994.

[259] S. Krishnamachari and R. Chellappa. Delineating buildings by grouping lines with mrfs. *Image Processing, IEEE Transactions on*, 5(1):164 –168, jan 1996.

[260] V. Krueger. *Gabor Wavelet Networks for Object Representation.* PhD thesis, Christian-Albrechts University, Kiel, Germany, 2001.

[261] P. Kruizinga and N. Petkov. Grating cell operator features for oriented texture segmentation. In *14th International Conference on Pattern Recognition*, volume 2, 1998.

[262] C. L. Krumhansl. Why is Musical Timbre so hard to understand? In S. Nielzén and O. Olsson, editors, *Structure and Perception of Electroacoustic Sound and Music, Proceedings of the Marcus Wallenberg symposium 1998*, pages 43–53. Excerpta Medica, 1989.

[263] H. Kueppers. *Harmonielehre der Farben.* DuMont, Koeln, 2000.

[264] M. P. Kumar and D. Koller. Efficiently selecting regions for scene understanding. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, pages 3217–3224. IEEE, 2010.

[265] S. Kumar. *Models for Learning Spatial Interactions in Natural Images for Context-Based Classification.* PhD thesis, The Robotics Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, 2005.

[266] S. Kumar. Discriminative graphical models for context-based classification. volume 285 of *Studies in Comp. Intell.*, pages 109–134. Springer, 2010.

[267] S. Kumar, J. August, and M. Hebert. M.: Exploiting inference for approximate parameter learning in discriminative fields: An empirical study. In *In: 5th International Workshop, EMMCVPR 2005, St. Augustine, Florida, Springer-Verlag (2005) 153 ? 168*, 2005.

[268] S. Kumar and M. Hebert. Discriminative random fields: A discriminative framework for contextual interaction in classification. In *In ICCV*, pages 1150–1157, 2003.

[269] S. Kumar and M. Hebert. Man-made structure detection in natural images using a causal multiscale random field. In *In Proc. IEEE Int. Conf. on Comp. Vision and Pattern Recog*, pages 119–126, 2003.

[270] V. K. A. Kutiyanawala. Eyes-free barcode localization and decoding for visually impaired mobile phone users. In *Proceedings of the 2010 International Conference on Image Processing, Computer Vision, Pattern Recognition (IPCV 2010), July 12-15*, 2010.

[271] J. Lafferty, X. Zhu, and Y. Liu. Kernel conditional random fields: representation and clique selection. In *Proceedings of the twenty-first international conference on Machine learning*, ICML '04, 2004.

[272] J. D. Lafferty et al. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*, 2001.

[273] R. Laganire. *OpenCV 2 Computer Vision Application Programming Cookbook.* Packt Publishing, 2011.

[274] S. Lakatos. A common perceptual space for harmonic and percussive timbres. *Perception & Psychoacoustics*, (62):1426–1439, 2000.

[275] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.

[276] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Efficient subwindow search: A branch and bound framework for object localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2129–2142, 2009.

[277] K. Laws. *Textured Image Segmentation.* PhD thesis, University of Southern California, 1980.

[278] T. H. Le and L. T. Bui. Face recognition based on svm and 2dpca. *CoRR*, 2011.

[279] C. Lee et al. Segmenting brain tumor with conditional random fields and support vector machines. In *Work. on Comp. Vision for Biomed. I. Appl. at ICCV*, 2005.

[280] C. Lee, R. Greiner, and M. Schmidt. Support vector random fields for spatial classification. In *Proc. of PDMKD*, pages 121–132, 2005.

[281] J. Lee. *A First Course in Combinatorial Optimization.* Cambridge University Press, 2004.

[282] T. S. Lee. Image representation using 2d gabor wavelets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:959–971, 1996.

[283] T. S. Lee and M. Nguyen. Dynamics of subjective contour formation in the early visual cortex. *Proceedings of the National Academy of Sciences of the United States of America*, 98(4):1907–1911, Feb. 2001.

[284] H. Lei and V. Govindaraju. Speeding up multi-class svm evaluation by pca and feature selection, 2005.

[285] V. Lempitsky, A. Vedaldi, and A. Zisserman. A pylon model for semantic segmentation. In *Advances in Neural Information Processing Systems*, 2011.

[286] M. Lew. *Principles of Visual Information Retrieval.* Springer, 2001.

[287] M. S. Lewicki. Efficient coding of natural sounds. nature neuroscience. *Nature Neuroscience*, 5:356–363, 10 2002.

[288] H. L. Lewis and C. H. Papadimitriou. *Elements of the Theory of Computation.* London: Prentice-Hall, 1981.

[289] J. Li, Q. Du, and C. Sun. An improved box-counting method for image fractal dimension estimation. *Pattern Recogn.*, 42(11):2460–2469, Nov. 2009.

[290] S. Z. Li. *Markov random field modeling in image analysis.* Springer-Verlag New York, Inc., 2001.

[291] H.-T. Lin, C.-J. Lin, and R. C. Weng. A note on platt's probabilistic outputs for support vector machines. *Mach. Learn.*, 68(3):267–276, Oct. 2007.

[292] B. Liu, S. Gould, and D. Koller. Single image depth estimation from predicted semantic labels. In *CVPR*, pages 1253–1260, 2010.

[293] C. Liu. Gabor-based kernel pca with fractional power polynomial models for face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:572–581, 2004.

[294] N. K. Logothetis. Vision: a window on consciousness. *Sci Am*, 281(5):69–75, 1999.

[295] T. Lourens, E. I. Barakova, H. G. Okuno, and H. Tsujino. A computational model of monkey cortical grating cells. *Biological Cybernetics*, 92(1):61–70, 2005.

[296] D. Lowe. Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1150 –1157, 1999.

[297] M. Luescher. *Luescher Test.* Rowohlt Verlag GmbH, 1984.

[298] H. Luke. The origins of the sampling theorem. *Communications Magazine, IEEE*, 37(4):106 –108, apr 1999.

[299] D. L. Macadam. Visual sensitivities to color differences in daylight. *J. Opt. Soc. Am.*, 32(5):247–273, May 1942.

[300] F. H. Mahnke. *Color, Environment and the Human Response.* Wiley, 1996.

[301] S. Mallat. *A Wavelet Tour of Signal Processing : The Sparse Way.* Academic Press, 2009. 3rd edition.

[302] B. Mandelbrot. How long is the coast of britain? statistical self-similarity and fractional dimension. *Science*, 156(3775):636–638, Jun 1967.

[303] B. B. Mandelbrot. *The fractal geometry of nature.* San Francisco, California: Freeman, 1983. 5th rev. and extended ed.

[304] M. B. Mansur, D.L. and K. Joy. Sound-graphs: A numerical data analysis method for the blind. In *Journal of Medical Systems*, pages 163–174, 1985.

[305] Maple. Maple virtual cable midi driver. http://www.maplemidi.com/.

[306] D. Margounakis and et al. Converting images to music using their colour properties, 2006.

[307] J. Matas, C. Galambos, and J. Kittler. Robust detection of lines using the progressive probabilistic hough transform. *Computer Vision and Image Understanding*, 78(1):119 – 137, 2000.

[308] G. Mayer-Kress, R. Bargar, and C. I.˙ *Musical structures in data from chaotic attractors.* SFI studies in the sciences of complexity. Addison Wesley Longman, 1992. Auditory display: Sonification, Audification, and Auditory Interfaces. Santa Fe Institute Studies in the Sciences of Complexity, Proceedings, Volume XVIII. Reading, MA: Addison Wesley Publishing Company.

[309] S. Mcadams. Perspectives on the contribution of timbre to musical structure. *Comput. Music J.*, 23(3):85–102, Sept. 1999.

[310] S. McAdams and E. Bigand. *Thinking in Sound: The Cognitive Psychology of Human Audition.* 1993.

[311] K. McCabe and A. Rangwalla. Auditory display of computational fluid dynamics data. In edited by G. Kramer, editor, *In Auditory Display. Sonification, Audification, and Auditory Interfaces*, pages 369–404. Reading, Massachusetts, USA: Addison-Wesley, June 1994.

[312] P. McCullagh and J. A. Nelder. *Generalized linear models.* Chapman and Hall, 1983.

[313] P. B. Meijer. An experimental system for auditory image representations. *IEEE transactions on bio-medical engineering*, 39(2):112–121, 1992.

[314] L. B. Merabet and A. Pascual-Leone. Neural reorganization following sensory loss: the opportunity of change. *Nature Reviews Neuroscience*, 11(1):44–52, 2009.

[315] T. J. Misa and P. L. Frana. An interview with edsger w. dijkstra. *Commun. ACM*, 53(8):41–47, 2010.

[316] A. Misra and P. R. Cook. Toward synthesized environments: A survey of analysis and synthesis methods for sound designers and composers. In *International Computer Music Conference*, 2009.

[317] B. C. J. Moore. *An Introduction to the Psychology of Hearing.* San Diego, Calif.: Academic Press, 1993. 4th edition.

[318] F. R. Moore. *Elements of Computer Music.* Prentice Hall, Englewood Cliffs, 1990. Second Edition.

[319] P. Morse. Vibration and sound. *American Institute of Physics, for the Acoustical Society of America*, 1986.

[320] R. L. Mott. *Sound Effects: Radio, Tv, and Film.* Focal Press, 1990.

[321] R. L. Mott. *Radio Sound Effects: Who Did It, and How, in the Era of Live Broadcasting.* Mcfarland & Co Inc, 2005.

[322] J. Moussouris. Gibbs and markov random systems with constraints. *Journal of Statistical Physics*, 10:11–33, 1974.

[323] J. Movshon, I. Thompson, and D. Tolhurst. Spatial summation in the receptive fields of simple cells in the cat's striate cortex. *J Physiol*, 283, 1978.

[324] J. A. Movshon, I. D. Thompson, and D. J. Tolhurst. Receptive field organization of complex cells in the cat's striate cortex. *The Journal of Physiology*, 283(1):79–99, Oct. 1978.

[325] H. Mueller, S. Marchand-Maillet, and T. Pun. The truth about corel – evaluation in image retrieval. In *Proceedings of the Challenge of Image and Video Retrieval (CIVR2002)*, pages 38–49, 2002.

[326] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Application VISSAPP'09)*, pages 331–340. INSTICC Press, 2009.

[327] N. Munakata. Musical representation of gene sequences. In *Arts Medicine*, pages 73–82, 1997.

[328] D. Munoz, J. A. Bagnell, and M. Hebert. Stacked hierarchical labeling. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Computer Vision - ECCV 2010 - 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part VI*, volume 6316 of *Lecture Notes in Computer Science*, pages 57–70. Springer, 2010.

[329] A. Munsell. *A Color Notation.* Munsell Color Company, Baltimore, MD, 1946.

[330] E. Murphy, E. Bates, and D. Fitzpatrick. Designing auditory cues to enhance spoken mathematics for visually impaired users. In *Proceedings of the 12th international ACM SIGACCESS conference on Computers and accessibility*, ASSETS '10, pages 75–82. ACM, 2010.

[331] K. Murphy, A. Torralba, and W. T. Freeman. Using the Forest to See the Trees: A Graphical Model Relating Features, Objects, and Scenes, 2003.

[332] NativeInstruments. Kontakt 5 player. http://www.native-instruments.com/en/products/komplete/synths-samplers/kontakt-5-player/.

[333] M. A. Nees and B. N. Walker. *Auditory interfaces and sonification.* New York: CRC Press, 2009. The Universal Access Handbook.

[334] P. Nelson. *Biological Physics.* Palgrave MacMillan, 2007.

[335] A. Neubeck and L. Van Gool. Efficient non-maximum suppression. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 3, pages 850 –855, 2006.

[336] I. Newton. *Opticks.* Dover, New York, 1952. book III.

[337] M. Nieto, C. Cuevas, L. Salgado, and N. Garca. Line segment detection using weighted mean shift procedures on a 2d slice sampling strategy. *Pattern Analysis and Applications*, 14:149–163, 2011.

[338] M. Nixon and A. Aguado. *Feature Extraction & Image Processing.* Academic Press, 2007.

[339] Y. Ohtani, S. Okamura, Y. Yoshida, K. Toyama, and Y. Ejima. Surround suppression in the human visual cortex: an analysis using magnetoencephalography. *Vision Res*, 42(15):1825–35, 2002.

[340] Y. Ostrovsky, A. Andalman, and P. Sinha. Vision Following Extended Congenital Blindness. *Psychological Science*, 17(12):1009–1014, 2006.

[341] Y. Ostrovsky, E. Meyers, S. Ganesh, U. Mathur, and P. Sinha. Visual Parsing After Recovery From Blindness. *Psychological Science*, 20(12):1484–1491, 2009.

[342] C. Padgham. The scaling of the timbre of the pipe organ. *Acta Acustica united with Acustica*, 60(3):189–204, 1986.

[343] C. Pantofaru, R. Unnikrishnan, and M. Hebert. Toward generating labeled maps from color and range data for robot navigation. In *IROS*, pages 1314–1321, 2003.

[344] G. Papari and N. Petkov. An improved model for surround suppression by steerable filters and multilevel inhibition with application to contour detection. *Pattern Recognition*, pages 1999–2007, 2011.

[345] D. Parikh and D. Batra. CRFs for image classification, 2006. Class Project: Probabilistic Graphical Models.

[346] T. E. Parks. Rock's cognitive theory of illusory figures: a commentary. *Perception*, 30(5):627–31, 2001.

[347] A. Pascual-Leone and R. Hamilton. The metamodal organization of the brain. *Progress in brain research*, 134:427–445, 2001.

[348] D. Payling, S. Mills, and T. Howle. Hue music - creating timbral soundscapes from coloured pictures. pages 91–97. Schulich School of Music, McGill University, 2007.

[349] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(11):559–572, 1901.

[350] J. Pelletier. Perceptually motivated sonification of moving images. *Proceedings of the 2009 International Computer Music Conference*, pages 207–210, 2009.

[351] A. Pentland. Fractal-based description of natural scenes. In *IEEE Trans. Pattern Analysis and Machine Intelligence*, volume 6, pages 661–674, 1984.

[352] S. C. Peres, V. Best, D. Brock, B. Shinn-Cunningham, C. Frauenberger, T. Hermann, et al. Auditory interfaces. pages 147–196. Burlington, MA: Morgan Kaufmann, 2008.

[353] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:629–639, 1990.

[354] N. Petkov and E. Subramanian. Motion detection, noise reduction, texture suppression and contour enhancement by spatiotemporal gabor filters with surround inhibition. *Biological Cybernetics*, 97(5-6):423–439, 2007.

[355] J. Pevsner. Leonardo da Vinci's contributions to neuroscience. *Trends Neurosci*, 25(4):217–20, 2002.

[356] D. P. Phillips and S. E. Hall. Response timing constraints on the cortical representation of sound time structure. *J Acoust Soc Am*, 88(3):1403–11, 1990.

[357] M. Pietikainen. *Texture Analysis in Machine Vision.* World Scientific Pub Co, 2000.

[358] M. Pietikainen. *Computer Vision Using Local Binary Patterns.* Springer, 2011.

[359] P. Pietrini, M. L. Furey, E. Ricciardi, M. I. Gobbini, W. H. Wu, L. Cohen, M. Guazzelli, and J. V. Haxby. Beyond sensory images: Object-based representation in the human ventral pathway. *Proc Natl Acad Sci U S A*, 101(15):5658–5663, 2004.

[360] C. J. Plack. *The Sense Of Hearing.* Lawrence Erlbaum Assoc Inc, 2005.

[361] M. Planck. *The Theory of Heat Radiation: Authorized Translation by Morton Masius.* Dover Publications, 1959.

[362] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, pages 61–74. MIT Press, 1999.

[363] R. Plomp. The rate of decay of auditory sensation. *Journal of the Acoustical Society of America*, 36:277–282, 1964.

[364] I. Pollack and L. Ficks. Information of elementary multidimensional auditory display. In *J. Acous. Soc. Amer.*, volume 26, page 155?158, 1954.

[365] D. Poole. *Linear Algebra: A Modern Introduction.* Canada: Thomson Brooks/Cole, 2006. second edition.

[366] A. Pope. *The Language of Drawing and Painting.* Harvard University Press, Cambridge, 1949.

[367] J. Powell. *A Thin Plate Spline Method for Mapping Curves Into Curves in Two Dimensions.* Cambridge DAMTP. Department of Applied Mathematics and Theoretical Physics, University of Cambridge, 1995.

[368] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C.* Cambridge University Press, 1992.

[369] R. Prim. Shortest connection networks and some generalizations. 36:1389?1401, 1957.

[370] P. Puranik, P. Bajaj, A. Abraham, P. Palsodkar, and A. Deshmukh. Human perception-based color image segmentation using comprehensive learning particle swarm optimization. In *Emerging Trends in Engineering and Technology (ICETET), 2009 2nd International Conference on*, pages 630–635, 2009.

[371] F. D. e. a. Purves D, Augustine GJ. *Neuroscience.* Sunderland (MA): Sinauer Associates, 2001. 2nd edition.

[372] A. Quattoni, M. Collins, and T. Darrell. Conditional random fields for object recognition. In *In NIPS*, pages 1097–1104. MIT Press, 2004.

[373] A. Quattoni, S. Wang, L.-P. Morency, M. Collins, and T. Darrell. Hidden conditional random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1848–1852, 2007.

[374] M. Quinn. Research set to music: The climate symphony and other sonifications of ice core, radar, dna, seismic, and solar wind data. In *Proceedings of the 7th Meeting of the International Conference on Auditory Display*, 2001.

[375] M. Quinn. For those who died: A 9/11 tribute. In *Proceedings of the 9th Meeting of the International Conference on Auditory Display*, 2003.

[376] M. Quinn. Walk on the sun: an interactive image sonification exhibit. *AI Soc.*, 27(2):303–305, May 2012.

[377] P. R. Timbre as a multi dimnensional attribute of complex tones. Technical report, 1970.

[378] R. Ramloll, W. Yu, S. Brewster, B. Riedel, M. Burton, and G. Dimigen. Constructing sonified haptic line graphs for the blind student: first steps. In *Proceedings of the fourth international ACM conference on Assistive technologies*, pages 17–25, New York, NY, USA, 2000. ACM.

[379] T. Rees. Detection of man-made structures in natural images, 2004.

[380] L. Reich, S. Maidenbaum, and A. Amedi. The brain as a flexible task machine: implications for visual rehabilitation using noninvasive vs. invasive approaches. *Curr Opin Neurol*, 25(1):86–95, 2012.

[381] L. Reich, M. Szwed, L. Cohen, and A. Amedi. A ventral visual stream reading center independent of visual experience. *Curr Biol*, 21(5):363–8, 2011.

[382] A. Reichenbach, K. Franze, S. Agte, S. Junek, A. Wurm, J. Grosche, A. Savvinov, J. Guck, and S. N. Skatchkov. Live cells as optical fibers in the vertebrate retina. *Selected Topics on Optical Fiber Technology*, 2012.

[383] X. Ren, L. Bo, and D. Fox. Rgb-(d) scene labeling: Features and algorithms. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2759–2766, 2012.

[384] W. Research. *Mathematica Edition: Version 8.0.* Wolfram Research, Inc., Champaign, Illinois, 2010.

[385] L. F. Richardson. The problem of contiguity: An appendix to statistic of deadly quarrels. *General systems: Yearbook of the Society for the Advancement of General Systems Theory*, 6:139–187, 1961.

[386] T. H. Riedenklau, E. and H. Ritter. Tangible active objects and interactive sonification as a scatter plot alternative for the visually impaired. In *Proceedings of the The 16th International Conference on Auditory Display*, pages 1–7, 2010.

[387] R. Rilke. *Primal Sound and Other Prose Pieces.* Cummington Press, 1943.

[388] C. Roads. *The Computer Music Tutorial.* MIT Press, MA, 2007.

[389] C. Roads and J. Strawn. *Foundations of Computer Music.* MIT Press, MA, 1985.

[390] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods (Springer Texts in Statistics).* Springer-Verlag New York, Inc., 2005.

[391] R. T. Rockafellar. Lagrange multipliers and optimality. *SIAM Rev.*, 35(2):183–238, June 1993.

[392] J. Roederer. *Introduction to the Physics and Psychophysics of Music.* The English Universities Press, London, 1973.

[393] G. L. Rogers. *Psychology of Music*, pages 198–207, title = Four Cases of Pitch–Specific Chromesthesia in Trained Musicians with Absolute Pitch, volume = 15, year = 1987.

[394] K. Rohr, M. Fornefett, and H. S. Stiehl. Approximating thin-plate splines for elastic registration: Integration of landmark errors and orientation attributes. In *In Proc. of IPMI'99, volume 1613 of LNCS*, pages 252–265. Springer, 1999.

[395] G. Rong, X. Song-yun, C. Xi-na, and Z. Hai-tao. Combined svm and pca to recognize the brain function from fmri images. In *Bioinformatics and Biomedical Engineering , 2009. ICBBE 2009. 3rd International Conference on*, pages 1 –3, 2009.

[396] R. Roscher, B. Waske, and W. Förstner. Kernel discriminative random fields for land cover classification. In *Pattern Recognition in Remote Sensing (PRRS), 2010 IAPR Workshop on*, pages 1–5, 2010.

[397] A. Rosenfeld. *Digital Picture Processing.* Acad.Press, Orlando, 1984.

[398] A. F. Rossi, C. D. Rittenhouse, and M. A. Paradiso. The representation of brightness in primary visual cortex. *Science*, 273:1104–1107, 1996.

[399] P. Roth, L. Petrucci, A. Assimacopoulos, and T. Pun. Ab-web: Active audio browser for visually impaired and blind users. In *Proceedings of the International Conference on Auditory Display*, 1998.

[400] P. Roth, L. S. Petrucci, and T. Pun. From dots to shapes: an auditory haptic game platform for teaching geometry to blind pupils. pages 603–610, 2000.

[401] C. Rother, V. Kolmogorov, and A. Blake. "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004.

[402] M. Rozak. Talk to your computer and have it answer back with the microsoft speech api. White paper, Microsoft Systems Journal, 1996.

[403] M. Russ. *Sound Synthesis and Sampling.* Focal Press, 2008.

[404] C. Russell, P. H. S. Torr, and P. Kohli. Associative hierarchical crfs for object class image segmentation. In *in Proc. ICCV*, 2009.

[405] R. Sahak, W. Mansor, Y. K. Lee, A. Yassin, and A. Zabidi. Performance of combined support vector machine and principal component analysis in recognizing infant cry with asphyxia. In *Engineering in Medicine and Biology Society (EMBC)*, 2010.

[406] N. Salamati, A. Germain, and S. Suesstrunk. Removing Shadows from Images Using Color and Near-infrared. In *Proc. IEEE International Conference on Image Processing (ICIP)*, IEEE International Conference on Image Processing ICIP, 2011.

[407] P. M. Sanderson. The multimodal world of medical monitoring displays. applied ergonomics. In *Applied Ergonomics*, volume 37, pages 501–512, 2006.

[408] M. Sarkar, C. Lan, J. Diaz, and B. Vercoe. The effect of musical experience on describing sounds with everyday words. *J Acoust Soc Am*, 125(4):18–22, 2009.

[409] M. Sarkar, B. Vercoe, and Y. Yang. Words that describe timbre: a study of auditory perception through language. In *Language and Music as Cognitive Systems Conference (LMCS-2007)*, 2007.

[410] N. Sarkar and B. B. Chaudhuri. An efficient approach to estimate fractal dimension of textural images. In *Pattern Recognition*, volume 25(9), pages 1035–1041, 1992.

[411] A. Saxena, J. Schulte, and A. Y. Ng. Depth estimation using monocular and stereo cues. In *Proceedings of the 20th international joint conference on Artifical intelligence*, IJCAI'07, pages 2197–2203, 2007.

[412] A. Saxena, M. Sun, and A. Y. Ng. Make3d: depth perception from a single still image. In *Proceedings of the 23rd national conference on Artificial intelligence - Volume 3*, AAAI'08, pages 1571–1576, 2008.

[413] C. Scaletti. *Sound synthesis algorithms for auditory data representations.* SFI studies in the sciences of complexity. Addison Wesley Longman, 1992. In Kramer, G (ed) 1994.Auditory display: Sonification, Audification, and Auditory Interfaces. Santa Fe Institute Studies in the Sciences of Complexity, Proceedings, Volume XVIII. Reading, MA: Addison Wesley Publishing Company.

[414] N. Schaffert, K. Mattes, and A. O. Effenberg. A sound design for acoustic feedback in elite sports. pages 143–165. Springer-Verlag Berlin, 2009.

[415] T. Schmele and I. Gomez. Exploring 3d audio for brain sonification. In *Proceedings of ICAD 2012 Eighteenth Meeting of the International Conference on Auditory Display*, June 2012.

[416] P. Schnitzspan. *Conditional Random Fields for Detection of Visual Object Classes.* PhD thesis, TU Darmstadt, September 2010.

[417] B. Schoelkopf and A. Smola. *Learning with Kernels.* MIT Press, 2002.

[418] M. Schoenwiesner, R. Ruebsamen, and D. Y. von Cramon. Spectral and temporal processing in the human auditory cortex - revisited. *Ann N Y Acad Sci*, 1060:89–92, 2005.

[419] P. A. Scholes. *The Oxford Companion to Music.* Oxford University Press, 1970.

[420] F. E. Schubert. *Light-Emitting Diodes.* Cambridge University Press, 2006.

[421] C. Seashore. *Studies in the psychology of music: Volume 1: the vibrato.* University of Iowa City, 1938.

[422] R. Sedgewick. *Algorithmen in C++.* Addison-Wesley, 1992.

[423] E. Sengpiel. Calculations of harmonics from fundamental frequency. Technical report, University of Arts, Berlin, 2012.

[424] W. A. Sethares. Rhythm and transforms, perception and mathematics. Klouche, Timour (ed.) et al., Mathematics and computation in music. First international conference, MCM 2007, Berlin, Germany, May 18–20, 2007. Revised Selected Papers. Berlin: Springer. Communications in Computer and Information Science 37, 1-10 (2009)., 2009.

[425] W. A. Sethares and D. Bañuelos. *Rhythm and Transforms.* Springer, 2007.

[426] L. Shamir. Human perception-based color segmentation using fuzzy logic. In *Proceedings of the 2006 International Conference on Image Processing, Computer Vision, & Pattern Recognition, Las Vegas, Nevada, USA, June 26-29, 2006, Volume 2*, pages 496–502, 2006.

[427] C. E. Shannon. A mathematical theory of communication. In *Bell System Technical Journal*, volume 27, pages 379–423, 1948.

[428] C. E. Shannon. Communication in the presence of noise. In *Proceedings of the Institute of Radio Engineers (IRE)*, volume 37, pages 10–21, 1949.

[429] C. E. Shannon. Prediction and entropy. In *Bell System Technical Journal*, volume 30, pages 50–64, 1951.

[430] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis.* Cambridge Univ. Press, 2004.

[431] H.-L. Shen and Q.-Y. Cai. Simple and efficient method for specularity removal in an image. *Appl. Opt.*, 48(14):2711–2719, 2009.

[432] J. R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Technical report, School of Computer Science Carnegie Mellon University Pittsburgh, PA 15213, 1994.

[433] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *Int. J. Comput. Vision*, 81(1):2–23, Jan. 2009.

[434] J. B. Shoval, S. and Y. Koren. Auditory guidance with the navbelt - a computerized travel aid for the blind. In *IEEE Transactions on Systems, Man, and Cybernetics*, 1998.

[435] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering object categories in image collections. In *Proceedings of the International Conference on Computer Vision*, 2005.

[436] J. O. Smith. Acoustic modeling using digitalwaveguides. *Musical Signal Processing*, pages 221–263, 1997.

[437] J. O. Smith. Discrete-time modeling of acoustic systems, 1997. CCRMA Stanford University, Online Course.

[438] J. O. Smith. Principles of digital waveguide models of musical instruments. *Applications of Digital Signal Processing to Audio and Acoustics*, pages 417–466, 1998.

[439] M. A. Smith. Surround Suppression in the Early Visual System. *The Journal of Neuroscience*, 26(14):3624–3625, 2006.

[440] L. L. Southworth. *Music and Sound.* Ayer Publishing, 1970.

[441] E. Sowards. Colors values and css. Technical report, http://www.erinsowards.com, 2011.

[442] spectrum. actually-thats-not-the-sound-of-the-higgs. http://spectrum.ieee.org/tech-talk/aerospace/astrophysics/actually-thats-not-the-sound-of-the-higgs.

[443] L. Spillmann and B. Dresp. Phenomena of illusory form: can we bridge the gap between levels of explanation? *Perception*, 24(11):1333–64, 1995.

[444] R. Sprengel, K. Rohr, and H. S. Stiehl. Thin-plate spline approximation for image registration. volume 3, pages 1190–1191, 1996.

[445] G. N. Srinivasan and S. G. Statistical texture analysis. *Proceedings of World Academy of Science, Engineering and Technology*, 36, 2008.

[446] J. Stoer. *Numerische Mathematik I.* Berlin: Springer, 1999. 8th edition.

[447] E. Striem-Amit, O. Dakwar, L. Reich, and A. Amedi. The large-Scale Organization of Visual Streams Emerges Without Visual Experience. *Cerebral Cortex*, 22(7):1698–1709, 2012.

[448] E. Striem-Amit, M. Guendelman, and A. Amedi. 'visual' acuity of the congenitally blind using visual-to-auditory sensory substitution. *PLoS ONE*, 7(3):e33136, 03 2012.

[449] B. Stroustrup. *The C++ Programming Language.* Adsison-Wesley, 1997.

[450] J. Su, A. Rosenzweig, A. Goel, E. de Lara, and K. N. Truong. Timbremap - enabling the visually-impaired to use maps on touch-enabled devices. In *In Proceedings of MOBILECHI*, 2010.

[451] S. Suzuki and K. Be. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1):32–46, 1985.

[452] Y. Suzuki and H. Takeshima. Equal-loudness-level contours for pure tones. *The Journal of the Acoustical Society of America*, 116(2):918–933, 2004.

[453] Synaptics. Synaptics sdk v1.0. http://www.synaptics.com/developers/manuals.

[454] R. Takahashi and J. Miller. Conversion of amino acid sequences in proteins to classical music: Search for auditory patterns. In *Genome Biology*, 2007.

[455] H. Takeshima, . a. S. Suzuki, Y., and T. Sone. Growth of the loudness of a tone burst with a duration up to 10 seconds. *The Journal of the Acoustical Society of Japan*, 9:295–300, 1988.

[456] H. Tamura, S. Mori, and T. Yamawaki. Textural features corresponding to visual perception. *Ieee Transactions On Systems Man And Cybernetics*, 8(6):460–473, 1978.

[457] R. T. Tan and K. Ikeuchi. Separating reflection components of textured surfaces using a single image. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(2):178–193, 2005.

[458] M. Teaque. Image analysis via the general theory of moments. *Journal of the Optical Society of America*, 70(8):920?930, 1980.

[459] J. Theiler. Estimating fractal dimension. *J. Opt. Soc. Am. A*, 7(6):1055–1073, Jun 1990.

[460] W. F. Thompson and R. Parncutt. Perceptual judgments of triads and dyads: Assessment of a psychoacoustic model. *Music Perception*, 14:263?280, 1997.

[461] J. Tighe and S. Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. In *Proceedings of the 11th European conference on Computer vision: Part V*, ECCV'10, pages 352–365. Springer-Verlag, 2010.

[462] M. Tkalcic and J. F. Tasic. Colour spaces - project for the digital signal processing course. Technical report, Faculty of electrical engineering, University of Ljubljana, Slovenia, type = Technical report, year = 2002.

[463] S. Todorovic and M. Nechyba. Detection of artificial structures in natural-scene images using dynamic trees. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 1, pages 35–39 Vol.1, 2004.

[464] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proceedings of the Sixth International Conference on Computer Vision (ICCV 98)*, page 839. IEEE Computer Society, 1998.

[465] A. Torralba, K. P. Murphy, and W. T. Freeman. Contextual models for object detection using boosted random fields. June 2004.

[466] A. Torralba, K. P. Murphy, and W. T. Freeman. Using the forest to see the trees: exploiting context for visual object detection and localization. *Commun. ACM*, 53(3):107–114, Mar. 2010.

[467] A. Torralba and A. Oliva. Depth estimation from image structure. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(9):1226–1238, Sept. 2002.

[468] G. Toussaint. The geometry of musical rhythm. In *In Proc. Japan Conference on Discrete and Computational Geometry, LNCS 3742*, pages 198–212. Springer-Verlag, 2004.

[469] G. Toussaint. Computational geometric aspects of rhythm, melody, and voice-leading. *Comput. Geom. Theory Appl.*, 43(1):2–22, Jan. 2010.

[470] G. T. Toussaint. A comparison of rhythmic dissimilarity measures. *FORMA*, 21(2):129–149, November 2006.

[471] R. T. Trevor Hastie and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, 2011.

[472] B. Trivedi. Sensory hijack: rewiring brains to see with sound. In *New Scientist*, 2010.

[473] R. J. Trudeau. *Introductory Graph Theory.* Dover, 1994. Second Edition.

[474] M. Tuceryan and A. K. Jain. Texture analysis. *Handbook of Pattern Recognition and Computer Vision*, pages 235–276, 1993.

[475] R. J. Ulrick. *Principles of Underwater Sound.* New York: McGraw-Hill, 1967.

[476] M. Valente, H. Hosford-Dunn, and R. J. Roeser. *Audiology.* Thieme, 2008.

[477] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Color descriptors for object category recognition. In *Proceedings of the IS&T European Conference on Colour in Graphics, Imaging, and Vision*, 2008.

[478] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluation of color descriptors for object and scene recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008.

[479] J. van de Weijer, C. Schmid, J. Verbeek, and D. Larlus. Learning color names for real-world applications. *Trans. Img. Proc.*, 18(7):1512–1523, July 2009.

[480] K. van den Doel. Soundview: Sensing color images by kinesthetic audio. pages 303–306, 2003.

[481] K. van den Doel. Geometric shape detection with soundview. pages 1–8, 2004.

[482] V. Vapnik. *Statistical Learning Theory.* John Wiley and Sons, New York, 1998.

[483] V. N. Vapnik. *The nature of statistical learning theory.* Springer-Verlag New York, Inc., 1995.

[484] M. Varma and R. Garg. Locally invariant fractal features for statistical texture classification. In *Proceedings of the IEEE International Conference on Computer Vision, Rio de Janeiro, Brazil*, October 2007.

[485] F. Vatablus. *Aristotelis de sensu et sensibili.* Paganus, 1559.

[486] H. B. Vickers, P. Sonification abstraite/sonification concrete: An aesthetic perspective space for classifying auditory displays in the ars musica domain. In *Proceedings of the 12th Meeting of the International Conference on Auditory Display*, 2006.

[487] B. Vintch, A. Zaharia, J. A. Movshon, and E. P. Simoncelli. Fitting receptive fields in V1 and V2 as linear combinations of nonlinear subunits. In *Computational and Systems Neuroscience (CoSyNe)*, February 2012.

[488] K. Vogt and R. Höldrich. A metaphoric sonification method - towards the acoustic standard model of particle physics. Washington, D.C., USA, June 9-15 2010. International Community for Auditory Display.

[489] K. Vogt, D. Pirro, I. Kobenz, R. Hldrich, and G. Eckel. Physiosonic - evaluated movement sonification as auditory feedback in physiotherapy. pages 103–120. Springer-Verlag Berlin, 2009.

[490] G. von Bekesy. *Experiments in hearing.* New York: McGraw-Hill, 1960.

[491] G. von Bismarck. Timbre of steady tones: A factorial investigation of its verbal attributes. *Acustica,*, 30:146–159, 1974.

[492] R. von der Heydt, E. Peterhans, and M. R. Duersteler. Periodic-pattern-selective cells in monkey visual cortex. *Journal of Neuroscience*, 12(4):1416–34, 1992.

[493] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. Lsd: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:722–732, 2010.

[494] J. von Goethe. *Zur Farbenlehre.* Number Bd. 2 in Zur Farbenlehre. J.G. Cotta, 1810.

[495] H. von Helmholtz and J. P. C. Southall. *Helmholtz's treatise on physiological optics.* Helmholtz's Treatise on Physiological Optics. The Optical Society of America, 1924.

[496] D. Wagner and D. Schmalstieg. ARToolKitPlus for Pose Tracking on Mobile Devices. Technical report, Institute for Computer Graphics and Vision, Graz University of Technology, 2007.

[497] G. Wahba. *Spline models for observational data*, volume 59 of *CBMS-NSF Regional Conference Series in Applied Mathematics.* Society for Industrial and Applied Mathematics (SIAM), 1990.

[498] B. N. Walker. Magnitude estimation of conceptual data dimensions for use in sonification. volume 8, pages 211–221, 2002.

[499] B. N. Walker and G. Kramer. Ecological psychoacoustics and auditory displays: Hearing, grouping, and meaning making. pages 150–175. New York: Academic Press., 2004.

[500] B. N. Walker and M. A. Nees. Theory of sonification. In T. Hermann, A. Hunt, and J. G. Neuhoff, editors, *The Sonification Handbook*, chapter 2, pages 9–39. Logos Publishing House, Berlin, Germany, 2011.

[501] R. S. Wallace. *Principles of Lighting.* London, Constable, 1951.

[502] H. Wechsler. Texture analysis - a survey. *Sig Process*, 2:271–282, 1980.

[503] P. Weiss. *Music in the Western World: A History in Documents.* Schirmer, 1984.

[504] D. L. Wessel. Timbre Space as a Musical Control Structure. *Computer Music Journal*, 3(2):45–52, 1979.

[505] P. White. The secrets of warmth & air. *Sound on Sound Magazine*, 2001.

[506] C. D. Wickens, S. E. Gordon, and Y. Liu. *An Introduction to Human Factors Engineering.* Prentice Hall, 1998.

[507] C. D. Wickens and Y. Liu. Codes and modalities in multiple resources: A success and a qualification. volume 30(5), pages 599–616, 1988.

[508] T. Wiegand and K. A. Moloney. *Handbook of Spatial Point Pattern Analysis in Ecology.* Crc Pr Inc, 2013.

[509] S. M. Williams. *Perceptual principles in sound grouping.* SFI studies in the sciences of complexity. Addison Wesley Longman, 1992. Auditory display: Sonification, Audification, and Auditory Interfaces. Santa Fe Institute Studies in the Sciences of Complexity, Proceedings, Volume XVIII. Reading, MA: Addison Wesley Publishing Company.

[510] L. Wiskott, J. M. Fellous, N. Krüger, and C. Malsburg. Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:775–779, 1997.

[511] T. Wu, M. Bae, M. Zhang, R. Pan, and A. Badea. A prior feature svm-mrf based method for mouse brain segmentation. *Neuroimage*, 59(3):2298–306, 2012.

[512] T.-F. Wu, C.-J. Lin, and R. C. Weng. Probability estimates for multi-class classification by pairwise coupling. *J. Mach. Learn. Res.*, 5:975–1005, Dec. 2004.

[513] X. Wu and Z.-N. Li. A study of image-based music composition. In *Multimedia and Expo, 2008 IEEE International Conference on*, pages 1345–1348, 23 2008-April 26.

[514] G. Wyszecki and W. S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formula.* Wiley-Interscience, 2000. Second Edition.

[515] X-Rite. A guide to understanding color tolerancing. Technical report, X-Rite Incorporated, Grandville, Michigan 49418, USA, 1997.

[516] S.-y. Xie, R. Guo, N.-f. Li, G. Wang, and H.-t. Zhao. Brain fmri processing and classification based on combination of pca and svm. In *Proceedings of the 2009 international joint conference on Neural Networks*, IJCNN'09, pages 3510–3515, 2009.

[517] J. Xu, Z. G. Fang, D. H. Dong, and F. Zhou. An outdoor navigation aid system for the visually impaired. In *Industrial Engineering and Engineering Management (IEEM), 2010 IEEE International Conference on*, pages 2435 –2439, dec. 2010.

[518] L. Xu, F. Qi, and R. Jiang. Shadow Removal from a Single Image. *Intelligent Systems Design and Applications, International Conference on*, 2:1049–1054, 2006.

[519] C. G. Yang and X. B. Duan. Credit risk assessment in commercial banks based on svm using pca. In *Machine Learning and Cybernetics, 2008 International Conference on*, volume 2, pages 1207 –1211, 2008.

[520] M. Y. Yang and W. Frstner. A hierarchical conditional random field model for labeling and classifying images of man-made scenes. In *ICCV Workshops*, pages 196–203. IEEE, 2011.

[521] S. Yanhui, D. Enqing, L. Zhenzhi, L. Chenglin, C. Bo, and L. Zhenguo. Research on the segmentation of tiny multi-target in brain tissues based on support vector machines. In *Complex Medical Engineering (CME), 2011 IEEE/ICME International Conference on*, pages 478 –482, 2011.

[522] W. S. Yeo and J. Berger. Application of image sonification methods to music. 2005.

[523] W. S. Yeo and J. Berger. A framework for designing image methods. 2005.

[524] W. S. Yeo and J. Berger. Application of raster scanning method to image sonification, sound visualization, sound analysis and synthesis. In *Proc. of the Int. Conf. on Digital Audio Effects (DAFx-06)*, pages 309–314, Montreal, Quebec, Canada, Sept. 18–20, 2006.

[525] T. Yoshida, K. Kitani, S. Belongie, K. Schlei, and H. Koike. Edgesonic: Image feature sonification for the visually impaired. In *International Conference on the Augmented Human*, Tokyo, 2011.

[526] T. Young. The bakerian lecture: On the theory of light and colours. *Phil Trans R Soc*, 92(5):12–48, 1802.

[527] A. Zacharakis, K. Pastiadis, G. Papadelis, and J. D. Reiss. An investigation of musical timbre: Uncovering salient semantic descriptors and perceptual dimensions. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*, pages 807–812, 2011.

[528] A. Zacharakis, K. Pastiadis, G. Papadelis, and J. D. Reiss. Analysis of musical timbre semantics through metric and non-metric data reduction techniques. In *Proceedings of the 12th International Conference on Music Perception and Cognition*, 2012.

[529] R. J. Zatorre and P. Belin. Spectral and Temporal Processing in Human Auditory Cortex. *Cerebral Cortex*, 11(10):946–953, 2001.

[530] S. M. Zeki. Uniformity and diversity of structure and function in rhesus monkey prestriate visual cortex. *J Physiol*, 277(1):273–290, 1978.

[531] P. Zelanski and M. P. Fisher. *Color*. Pearson Prentice Hall, 2003. Fourth Edition.

[532] H. Zhao, C. Plaisant, B. Shneiderman, and J. Lazar. Data sonification for users with visual impairment: A case study with georeferenced data. *ACM Trans. Comput.-Hum. Interact.*, 15(1):4:1–4:28, May 2008.

[533] H. Zhou and D. Suter. Fast sparse gaussian processes learning for man-made structure classification. In *Online Learning for Classification Workshop 2007*, 2007.

[534] M. Zhou. *Gabor-Boosting Face Recognition*. PhD thesis, School of Systems Engineering, University of Reading, 2008.

[535] M. Zhou and H. Wei. Face verification using gaborwavelets and adaboost. In *18th International Conference on Pattern Recognition (ICPR 2006), 20-24 August 2006, Hong Kong, China*, pages 404–407, 2006.