A MODULAR COMPUTER VISION SONIFICATION MODEL FOR THE VISUALLY IMPAIRED

Michael Banf & Volker Blanz

Universität Siegen,
Media Systems Group, Germany
{banf, blanz}@informatik.uni-siegen.de

ABSTRACT

This paper presents a *Modular Computer Vision Sonification Model* which is a general framework for acquisition, exploration and sonification of visual information to support visually impaired people. The model exploits techniques from Computer Vision and aims to convey as much information as possible about the image to the user, including color, edges and what we refer to as *Orientation maps* and *Micro-Textures*. We deliberatively focus on low level features to provide a very general image analysis tool. Our sonification approach relies on MIDI using "real-world" instead of synthetic instruments. The goal is to provide direct perceptual access to images or environments actively and in real time. Our system is already in use, at an experimental stage, at a local residential school, helping congenital blind children develop various cognitive abilities such as geometric understanding and spatial sense as well as offering an intuitive approach to colors and textures.

1. INTRODUCTION

Sonification is defined in [1] as "use of non-speech audio to convey information". It describes both a scientific and an artistic discipline. In recent times it is even used to audibly browse the RNA structure [2] or model an acoustical representation of the standard model of particle physics [3]. In this paper we address the sub-field of developing sonification methods to support visually impaired people. In the last years, we have seen a number of special purpose devices that help visually impaired people to solve everyday problems, such as finding their way [4] or reading text. These devices tend to be restricted to a specific problem by extracting very specific information from the input data or by relying on additional information, such as positioning systems. In contrast, our goal is to develop a general-purpose system that can be used in a wide range of situations. Our system analyzes visual data using general image descriptors from computer vision, and maps these to auditory signals. Unlike approaches that use, for example, face detectors, we deliberately focus on low-level descriptors such as colors and edges in order to obtain a device that can sonify any given image and help to solve any given question that persons with normal vision could solve: We want to enable users to recognize what kind of scene is shown, find objects in this scene, or explore the photos of a friend. Unlike high-level image analysis, our device gives direct feedback on what is where in the image. We are inspired by the way how visually impaired persons can explore reliefs of scenes with their fingers and get a direct haptic experience of the shapes and locations of objects – they can feel what the peak of a mountain or a what cloudy sky is. 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 part of the system.

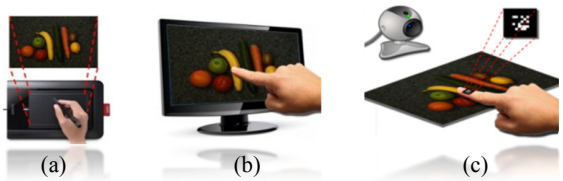


Figure 1: Several exploring interfaces: (a) Pen to tablet interface with absolute coordinates. (b) Touch screen working directly on the acquired image. (c) Finger position *(x,y)* is tracked using a marker system and estimated within the image

There has been done previous work on the sonification of low-level characteristics of images for visual impaired. [5] generates sounds depending on a pixel's lightness and its position within the image. [6] demonstrates a technique, using color patches within images as chromatic patterns that can be put together to form a melody. [7] uses color attributes to filter an underlying white noise using *Subtractive Synthesis*. [8] mixes 8 pre-recorded musical timbres depending on the quantity of 8 hue values within an image. All such synthesized sounds are not easily interpreted at first. Especially the last two approaches map all attributes of a certain color onto a single sound parameter, such as timbre, pitch or frequency spectrum. Such mappings are hardly reversible and might lead to identical sound synthesis from different input color combinations. Our contribution to previous work is to present a new and holistic approach to color and texture sonification, which is intuitive enough not only to be understood and applied by congenital blind, but also helps to convey e.g. the concept of colors and color mixing itself. We consider our approach to be holistic, as it maps each attribute of a particular color within a color-space – such as hue, saturation and lightness - to an intuitive, but separate, counterpart within the sound space at once. Additionally we map texture features to unique sound in the same way. The sonification of colors is important for two reasons, first, to offer a way to congenital blind people to understand colors and to be able to communicate with non-visually impaired about such fundamental quality of human vision. Second, in illuminated environments, colors are crucial in the process of detecting objects. If our sonification is intuitive enough, a person will quickly learn to understand the concept of colors and textures as well as the audification, recognize objects, interpret images and develop their own strategies. Designing such a system, we face the following challenges:

- Sensory input: We propose a system that analyzes images that users may find in a photo collection, on the internet, or capture with a still camera. Future devices could allow for depth information (stereo or time-of-flight devices), motion or other input.

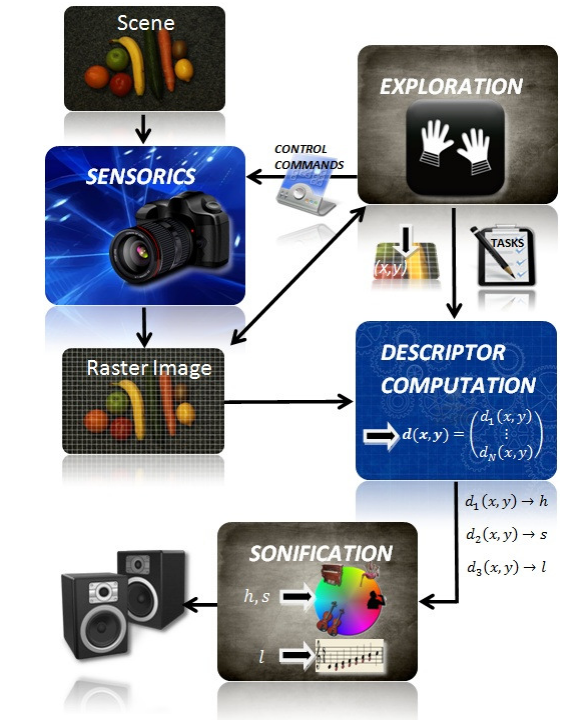


Figure 2: The *Modular Computer Vision Sonification Model.*

- Design the most appropriate visual descriptors to represent particular image values. They should be informative, general and stable under transformations such as illumination and pose.
- Define a way to sonify these image descriptors. The goal is to convey as much information as possible without interaction between the channels as well as to give an auditory perception that enables users to develop an "intuition" about the visual data.
- Develop an 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, constant auditory signal. Therefore, we decided that users should explore the image locally, for example on a touch screen.

Our paper is on the interface between computer vision and sonification. Our contribution is the general concept of a *Modular Computer Vision Sonification Model* with the components *Sensorics – Exploration - Machine Vision – Sonification – Human Learning*, 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 have been formulized e.g. in [9]. The importance of interaction in sonification is argued in [10].

2. THE MODULAR SONIFICATION MODEL

Figure 2 gives an overview of the general concept of our *Modular Computer Vision Sonification Model.* Stage one is the

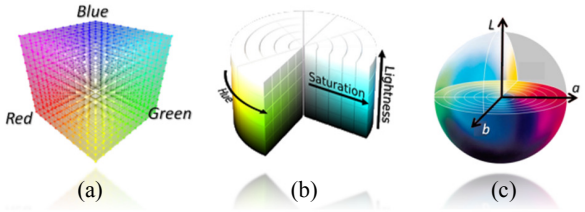


Figure 3: Color Models: (a) *RGB.* (b) *HSL.* (c) *CieLab*

acquisition of both a rasterized image ***I*** as well as a particular pixel position *(x,y)* where the image is to be explored. Further, several tasks about what features shall be calculated and sonified are determined and passed through during exploration. Next step in the process is the creation of a *pixel descriptor* ***d****(x,y)* – a vector gathering all the information to be sonified. The descriptor captures image information in the local neighborhood of *(x,y)* and is calculated using computer vision techniques. This information might be very fundamental such as color, texture, edges as well as more complex such as an estimation of depth or whether the current pixel belongs to a face. Every feature *i* makes an element $d_i(x,y)$ of the pixel descriptor ***d****(x,y)* and is transferred to the sonification unit. In the next sections, we discuss each module and describe our specific design and its implementation.

3. SENSORICS

The sensory module acquires the data to be sonified. In the system presented in this paper, we rely on still images that are available as files. It is easy to apply our system to still images from a camera operated by the user, or on images from web pages. Future extensions may include depth information (from stereo vision or depth sensors), infrared images, GPS positioning, and motion information in videos.

4. EXPLORATION

Our exploration module has two major tasks. First, during the image acquisition, it allows the user to transmit control commands to the sensory module such as activating certain sensors or controlling their viewing direction. Second, it enables the user to navigate within an image, passing on its position *(x,y)* to the descriptor computation unit, along with several tasks that determine which features of ***d****(x,y)* shall be sonified at all.

4.1. Navigation

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. Hence, we worked with several interfaces, as shown in Fig. 1, to see what suits best to a blind person. The pen – tablet interaction method 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 pads without a pen usually provide only relative positioning (similar to a mouse). For training and several user studies in section 7 we utilized a touch screen that allows the user to interact direct with the image plane (without seeing the image). Even though it does not make sense for a

(a) (b)

Figure 4: (a) Raster Images. (b) *Bilateral filtered* images.

blind person to buy a touch screen, the absolute positioning proved very successful. Therefore we implemented a more effective and far cheaper interaction method, applying a camera-based finger position tracking based on *ARToolKitPlus* pose tracking system [11]. The same camera as to acquire the visual data is utilized to detect a marker, attached to the user's fingernail, which is thereafter calculated back to estimate the fingers position within the image.

4.2. Control Gestures

The use of a pose tracking system turns out to be of great interest in controlling the whole system as well. As the utilized *ARToolKitPlus* is able to deal with several markers at once it allows us to recognize many finger and hand gestures to submit control commands to the sensory module as well as sonification tasks to the descriptor computation unit. In contrast, a keyboard or virtual buttons on the touch screen are difficult to operate for blind users.

5. PIXEL DESCRIPTOR COMPUTATION

The *pixel descriptor* ***d****(x,y)* holds all relevant features to be sonified at an image position *(x,y)*. We focus on fundamental characteristics such as colors and what we call *Orientation maps* and *Micro-Textures*. However, as intended, the module can be extended to extract and store more complex information.

5.1. Color Information

There are different color systems with several motivational backgrounds, as shown in Fig. 3. The *RGB* model uses additive mixtures of red, green and blue. It is motivated by the human eye receptors [12] and applied, e.g. in many display devices. However, providing a non-visual access to colors, as in our case, requires a more intuitive system, especially for congenital blind persons. This is why we prefer the *HSL* model [13], where each color value is described by hue *h*, saturation *s* and lightness *l*. 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 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* [14], as shown in Fig. 4, which filters noise while preserving edges within an image and will play a significant role in finding orientation maps. Subsequently, we use the smoothed color values as the first three elements of our pixel descriptor ***d****(x,y):*

$$d_1(x,y) = h_{smooth}(x,y), \quad d_2(x,y) = s_{smooth}(x,y), \quad d_3(x,y) = l_{smooth}(x,y)$$

5.2. Orientation Maps

The rationale behind *Orientation Maps* and *Micro-Textures* is

(a) (b)

Figure 5: (a) *Gaussian Pyramids* on red, green and blue channels. (b) *Gabor Transform* response images.

to create something like an acoustical relief that allows the user to hear what is under his fingers, instead of feeling it. While micro-textures - explained in the next section - express the overall roughness characteristics of a particular patch, orientation maps represent dominant structures within the image. 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 *Canny* [15] 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 Inhibition* in the human visual system that improves contour detection [12]. Moreover, regular repeating patterns of a certain size tend to be human made, unlike the more fractal patterns that are often found in nature [16]. Finding human made structures is important when using the system to orient within an environment to find windows, doors, ways, tables, shelves and so forth.

5.2.1. Calculating Orientation Maps

In the first step of calculating orientation maps, we use cascades of *Median* [17] and *Bilateral Filtering* to suppress both noise and small corners, as shown in Fig. 4 (b). Then, we reduce the spatial resolution of the red, green and blue color channels to obtain a *Gaussian Image Pyramid* [18], as shown in Fig. 5 (a). On each level, the width and height is reduced by a factor of 2. This reduction process removes low-scale variations which may be irrelevant for the task of the user. By later combining information of different layers using the image pyramid we can select the most appropriate resolution for each visual feature. Next, we perform a *Gabor Transform* [19] on each channel separately and build a final response image by measuring all individual responses. The transform relies on *Gabor Wavelets* $\psi_{\varphi,\nu}(x,y)$ [20] of the form:

$$\psi_{\varphi,\nu}(x,y) = \frac{\|k_{\varphi,\nu}\|^2}{\sigma^2} e^{-\frac{\|k_{\varphi,\nu}\|^2 \|(x,y)\|^2}{2\sigma^2}} \left[e^{ik_{\varphi,\nu}(x,y)} - e^{-\frac{\sigma^2}{2}} \right] \quad (1)$$

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. $k_{\varphi,\nu}$ is the wave vector, combining orientations and the spatial frequency in the frequency domain. Gabor Wavelets are widely used in computer vision [20], because they provide an analysis of spatial frequency that is local, unlike the global analysis in a Fourier Transform [17]. To filter orientation maps, we choose $\nu = 1$ and later combine their particular response – inspired by the cascading of several *simple cells* to form *Complex Cells* [12] in the human visual cortex. We apply them in 32 orientations $\varphi = \{0°, 5.625°, 11.25°, 16{,}875, \ldots, 180°\}$. For the carrot and the orange in Fig. 4 (a), such response images are shown in Fig. 5 (b). The 32 orientations φ are visualized by gray scale values. We

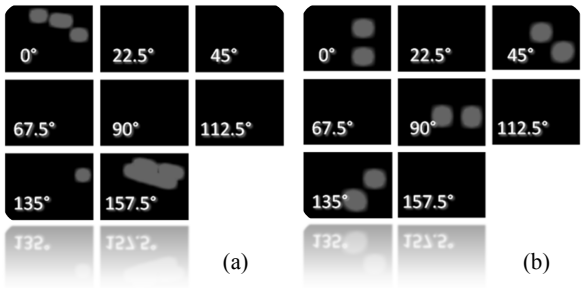


Figure 6: *8 Orientation Maps* for (a) carrot and (b) orange image

quantize the response image so that the each of the 32 orientations φ is mapped to the next of 8 orientations $\theta = \{0°, 22.5°, 45°, 67.5°, 90°, 112.5°, 135°, 157.5°\}$. Each θ is represented in an individual gray scale image $\boldsymbol{G}_\theta$, where $\boldsymbol{G}_\theta(x,y) = 255$, in case of an edge and $\boldsymbol{G}_\theta(x,y) = 0$ otherwise. One 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. Therefore, we calculate for each image position (x,y) the *variance* $\sigma^2(x,y)$ of values $\boldsymbol{G}_\theta$ on its local neighborhood on each of the 8 images $\boldsymbol{G}_\theta$, to obtain what we call *Orientation Maps* $\boldsymbol{O}_\theta$:

$$O_\theta(x,y) = \sigma^2(x,y) = \sum_{i=-w/2}^{w/2} \sum_{j=-w/2}^{w/2} (G_\theta(x+i, y+j) - \mu(x,y))^2$$

with

$$\mu(x,y) = \frac{1}{w^2} \sum_{i=-w/2}^{w/2} \sum_{j=-w/2}^{w/2} G_\theta(x+i, y+j) \qquad (2)$$

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 (Fig. 4 (a)) are shown in Fig. 6. Each coherent patch of gray scale pixels, on each orientation map is referred to as *Orientation Patch* $V_{\theta,i}$. As the diffusion approach might cause overlap in image positions (x,y) of different oriented orientation patches, as illustrated in Fig. 7 (a), we now compare orientation patches and emphasize the dominating ones while suppressing insignificant and therefore distracting others.

5.2.2. A Topological Representation of Orientation Patches

We consider orientation patches as dominant if they have a certain size – which is the number of their pixel positions. So far, we handle 4 cases. Case 1: If an image area is dominated by two very big orientation patches of almost equal sizes, both above a certain threshold t_{size} and such patches differ in orientation by more than 22.5°, they are retained as coexisting. This happens to be the case for 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 two orientation patches, both greater than t_{size} , having an orientation difference of only 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. After that, the smaller orientation patch is erased from its orientation map. Case 3: If the image area contains a large orientation patch, with a size greater than t_{size}, and further patches, whose sizes are below $t_{size,}$ such smaller patches are

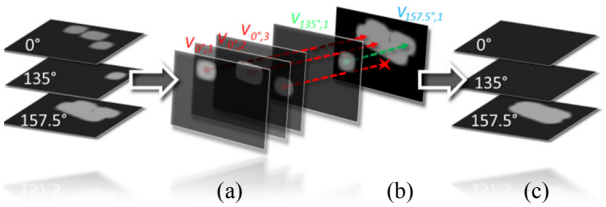


Figure 7: (a) Split initial Orientation Maps. (b) Arrows: centers of $V_{0°,1}$, $V_{0°,2}$ and $V_{3,1}$ lie within $V_{2,1}$. Cross: center of $V_{1,3}$ does not lie within $V_{2,1}$. (c) Final Orientation Maps.

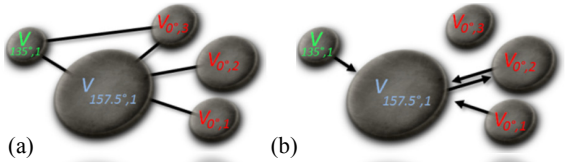


Figure 8: (a) G_o of Fig 6 (a). (b) G_c of Fig. 6 (a).

either (a) merged into the big one, as described in case 2, or (b) deleted from their orientation maps, depending on whether their particular centers lie within the large orientation patch or not. Case 4: If the image area contains several orientation patches, whose sizes are below t_{size}, they are merged as in case 2, in case both their center positions overlap and their orientation difference is equal to 22.5°. Otherwise they coexist. To implement the rules described in this section, we build a topological representation of all overlapping orientation patches in that area by the following procedure. At first, we apply a *contour finding algorithm* [21] 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, belong to an orientation patch $V_{\theta,i}$. We now represent each $V_{\theta,i}$ as a single image, as illustrated in Fig. 7 (a). So far, for each $V_{\theta,i}$ we have its size and can calculate its center of gravity. Starting with a particular orientation patch $V_{\theta,i}$, we now check pixel by pixel for overlaps with each different oriented orientation patch. In case of overlapping, we compute the number of mutual pixel positions and whether the center of one orientation patch lies within the other. The results can be processed by *graph theory* [22] in the following way. Overlapping orientation patches can be modeled as an *undirected Graph* $G_o = \{V,E\}$, where *knots* V represent all orientation patches and *edges* E represent the existence of an overlap between a pair of different oriented orientation patches. A second graph G_c is set up to represent only such overlaps, where at least the center c of one of the two orientation patches involved is found inside the related orientation patch, as visualized in Fig. 7 (b). In this case connections may be only in one direction, so G_c is a *directed graph*. Both graphs G_o and G_c, for all orientation patches of the carrot image (Fig. 6 (a)), are illustrated in Fig. 8. Such topological representations can now be used to find which of the 4 previously mentioned cases fit. For the carrot image we find, based on G_o and G_c (Fig. 8 (c)) that $V_{0°,1}$ and $V_{0°,2}$ merge into $V_{157.5°,1}$ applying case 3a and $V_{0°,3}$ is deleted, applying case 3b. Hence, the final orientation maps $\boldsymbol{O}_{0°}$, $\boldsymbol{O}_{135°}$ and $\boldsymbol{O}_{157.5°}$ are shown in Fig. 7 (c). In contrast, the Graph G_o based on the orientation maps of the orange image, Fig. 6 (b), would result in a ring shaped connected structure, where every knot would be about the same size. Hence, such orientation patches are left as they are, according to case 4. Eventually, we can assign the shares of all eight orientation maps at a particular pixel position (x,y) to the pixel descriptor

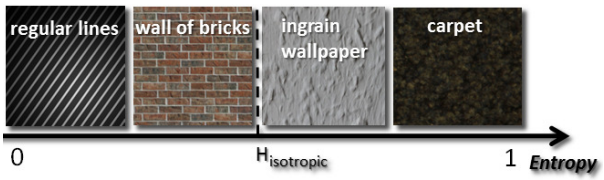


Figure 9: 4 types of textures with increasing *Entropy*.

$\boldsymbol{d}(x,y)$. We define $\Omega_\theta(x,y) \in \{0,1\}$ to be either 1, in case $\boldsymbol{O}_\theta(x,y) > 0$, or 0, if $\boldsymbol{O}_\theta(x,y) = 0$.

$d_4(x,y) = \Omega_{0°}(x,y)$, $d_5(x,y) = \Omega_{22.5°}(x,y)$ $d_6(x,y) = \Omega_{45°}(x,y)$,
$d_7(x,y) = \Omega_{67.5°}(x,y)$, $d_8(x,y) = \Omega_{90°}(x,y)$, $d_9(x,y) = \Omega_{112.5°}(x,y)$,
$d_{10}(x,y) = \Omega_{135°}(x,y)$, $d_{11}(x,y) = \Omega_{157.5°}(x,y)$

Note that as borders of orientation patches fade out, caused by our *variance* σ^2 *diffusion* approach, we are also able to have $\Omega_\theta(x,y)$ run from 0 to 1 in several steps: $\Omega_\theta(x,y) \in \{0,\ldots,1\}$.

5.3. Micro Textures

What we call *Micro-Textures* in our system captures the roughness and local structure at a point in an image. Examples of textures that we want to distinguish are shown in Fig. 9: regular lines, brick walls, ingrain wallpaper or carpet. In the Computer vision literature, there are many approaches to describe texture, even though it is difficult to give a general definition. [13] describes three different groups of texture measures. First, there are *First Order Statistical Texture Measures* [17] such as the *mean* μ and *variance* σ^2 of color values in a local neighborhood. Second, there are *Second Order Statistical Texture Measures*. These texture measures do not analyze single intensities, but correlations between pairs of pixel values. An example of this approach are *Haralick texture measures* based on *Co-Occurrence-Matrices* [23]. Unfortunately, a major drawback are high calculation costs. There have been efforts to speed up the processing using GPU [24]. Finally there is *Spectral Image Analysis* such as e.g. the *Fast Fourier Transform* [17] or *Gabor Transform*. To build micro-textures we use the *Gabor Transform* and compute the *Entropy* as a texture measure.

5.3.1. Entropy as Texture Measure

The *Gabor Transform* is applied to the original image and not to the bilateral filtered image, as we now want 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 [25]. In information theory, it is formulated as:

$$H = -\sum_{i=1}^{N} p_i \log p_i \quad \text{with} \quad p_i = \frac{N_i}{N} \qquad (3)$$

We calculate the *Entropy* $H(x,y)$ for each pixel position (x,y), based on the *Gabor Transform* response within a local neighborhood. 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. Based on $H(x,y)$ we can now measure the roughness of an image region and assign it to the last element of $\boldsymbol{d}(x,y)$:

$$d_{12}(x,y) = \begin{cases} 0 \ (\textit{smooth surface}), & \text{if } N < N_{minimum} \\ 1 \ (\textit{anisotropic roughness}), & \text{if } 0 \le H(x,y) < H_{isotropic} \\ 2 \ (\textit{isotropic roughness}), & \text{if } H_{isotropic} \le H(x,y) \end{cases}$$

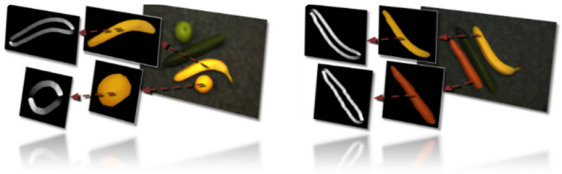


Figure 10: Some results of color segmentation.

5.4. Grabbing Objects

We utilize state of the art segmentation algorithms to separate the image into regions that are likely to show different real life objects. The goal is to help users find and scrutinize objects and other entities in the image, based on orientation maps and micro-textures calculated for such parts only. Image Segmentation or more precisely a foreground / background segmentation - is performed using *Gaussian Mixture Models* calculated using *Expectation-Maximization* and *Graph Cuts* [26], [27], [28]. First, the user moves over an area of interest and initiates the segmentation procedure by pressing a button or by gesture. Based on the users current position (x,y) we apply a *flood-fill* algorithm [29] 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. The color distances are calculated as the Euclidean distance $\|\ldots\|$ in *CieLab* [13] Color Space, as illustrated in Fig. 4 (c). All these selected pixels are marked as "definite foreground", all others as "probably background", and both groups serve as first segmentation estimation and as input to the *Expectation Maximization* and *Graph Cut* algorithms, which then calculate the final segmentation. Fig. 10 shows some exemplary results of the segmentation, as well as the responses of the *Gabor Transform* applied to such segmentations. The whole segmentation process takes approximately *4.5* seconds and can therefore be considered for interactive usage.

6. SONIFICATION CONCEPT

The previous sections dealt with extracting features to form the pixel descriptor $\boldsymbol{d}(x,y)$. We now discuss the question of how to sonify those features. A great challenge is to avoid conflicting signals and information overload, as well as the transformation of quasi-static 2D image data into a dynamic audio stream. Though humans can distinguish many attributes such as *pitch, volume, ADSR-Curve, timbre, roughness or vibrato*, it is still impossible to transport all potential descriptors of visual information simultaneously. Unlike approaches that sonify a whole image sequentially e.g. by scanning its pixels row by row [30] we want the user, as already described, to fully interact with the visual data in real- time and to be able to hear what is currently under his finger. Second, we want a method to simultaneously sonify features such as color, orientation maps and micro-textures and even more, instead of focusing on a single feature such as the progression of edges [31], [32]. Third the sonification model should meet aesthetical demands that are important for comfortable and extensive usage.

6.1. Sonification of Color Information

Sonification systems may use different techniques of sound synthesis, such *as Subtractive Synthesis*, *Additive Synthesis, Granular Synthesis, Physical Modeling* or *FM Synthesis* [33]. The methods presented in this paper rely on sounds from common instruments based on the General MIDI (GM) Standard (based on *Wavetable Synthesis* [33]). Visual impaired people

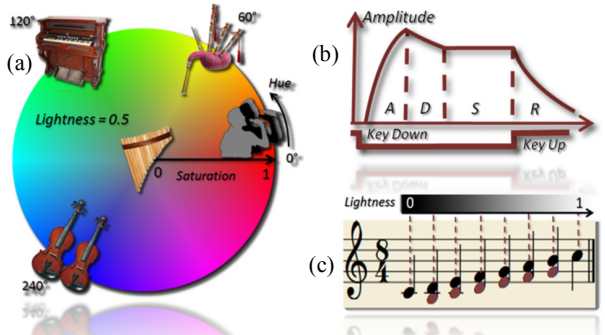
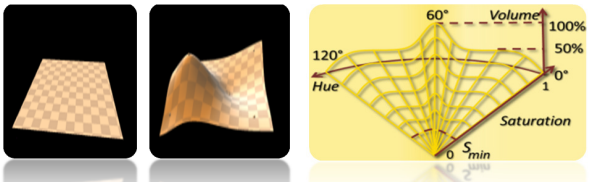


Figure 11: (a) *Complementary Instruments* represent pairs of *opponent colors*. (b) *ADSR* Envelope for the hit of a piano key. (c) Lightness $l \in \{0,\ldots,1\}$ and musical scale each l is assigned to. Brown notes are the added thirds.

may find this a comfortable and soon a familiar way to get perceptual access to colors and textures. Instead of strictly learning particular associations between instruments and colors (which they do not see and therefore have to memorize) we want to help them to build connections between sonification signals and objects of their daily life. In fact, in our experiments, we often heard participants say that an image region "sounds like a tomato" or any other object rather than announcing the correct mixture of colors, such as red – as in case of the tomato.

6.1.1. Complementary Instruments

As we use common instruments to sonify visual information, we propose a concept that represents each color value in the *HSL* model as a mixture of instruments, inspired by Hering's *theory of opponent colors* [12]. In principle, we use what we call *Complementary Instruments* to represent the *opponent color pairs* red-green and blue-yellow, as shown in Fig. 11 (a), and later combine adjacent instruments to represent color mixtures. As no mixture of a pair of opponent colors exists [12], there will be no mixture of a pair of complementary instruments in the sonification model either. Further we apply a musical scale to represent the luminance scale from black to white. Complementary instruments therefore must guarantee certain characteristics. First, they must possess a relatively stable frequency spectrum over time. That means that in terms of *Attack-Decay-Sustain-Release - Amplitude envelope (ADSR)*, as shown in Fig. 11 (b), they should have a short *Attack-* and *Decay-*, an infinite *Sustain-* and a short *Release-Phase*. To avoid mutual masking of instruments, their frequency spectra should have narrow bandwidths (i.e. little noise components). In addition to appropriate *ADSR*-characteristics, there are further criteria that a set of 4 complementary instruments has to fulfill: *Separability* 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. Second we need *Uniqueness:* Even complementary instruments need to be unique enough to be associated with its particular color. Finally, we want to make sure that mixtures of instruments do not sound like other, new instruments. Fig. 11 (a) shows our final selection of instruments: Choir (red), bagpipe (yellow), organ (green), strings (blue) and flute (white, black, gray). The software allows users to assign own selection of preferred instruments. The specific role of gray-scale, black and white with only one instrument will be explained in the next section.



(a) (b) (c)

Figure 12: (a) 2D spline. (b) Same 2D spline deformed given 6 control values c. (c)*Volume Shape* $\vartheta(h,s)$ scheme around yellow.

6.1.2. The Concept of HSL-Color Sonification

As explained in section 5 the *HSL* color model describes a certain color using hue h, as an angle from 0° to 360°, lightness l and saturation s. This color information of a pixel is stored in the first three element of $\mathbf{d}(x,y)$: $h = d_1(x,y)$, $s = d_2(x,y)$, $l = d_3(x,y)$. 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. The fade of saturation s, moving inward to the center of Fig. 11 (a), is considered as a general absolute decrease in volumes of any two color instruments playing simultaneously, while their relative volume ration is maintained. However, below a certain threshold s_{min} we regard the color as gray and sonify it using a single instrument, the flute. 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. The lightness l of gray or any other (combination of) colors is sonified as the pitch of the tone. Gray scale images, therefore will be sonified as a flute playing at varying pitch. Based on a musical scale, as shown in Fig. 11 (c), 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 11 semitones. For harmonic reasons we only utilize the whole tones of the octave and map each lightness value l between 0 and 1 to one of the eight tones. Further, we add thirds to all six intermediate tones. This creates a more comforting and aesthetical 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. 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 *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.

6.1.3. Calculation of Volume Shapes

Calculating the volumes of instruments in a mixture of sounds for all intermediate colors is an interpolation problem. Simple linear (barycentric) interpolation would be too restricted because once the overall volume of each instrument is set, there would be no way to counteract the dominance of some instruments in some specific mixtures. Therefore, we use *thin plate spline interpolation* [34] based on a set of control points. The fundamental idea behind the method is the physical model of a flat thin medal plate as in Fig. 12 (a) that is deformed by a few punctual strains, which we will call control values c. The plate is than forced into a new form that minimizes the

Figure 13: (a) Test set. (b) Possible orientations of objects.

deformation energy (Fig. 12 (b). We calculate a *Volume Shape* $\vartheta(h,s)$ that maps a volume ϑ to each color (h,s). For the bagpipe (color: yellow), this is visualized in Fig. 12 (c). The volume should be 100% at hue $h = 60°$ and full saturation $s = 1$, 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 in new positions (h_c,s_c). The calculation of such volume shapes $\vartheta(h,s)$ involves linear combinations of *radial basis functions* $f(h,s)$ [34]:

$$\vartheta(h,s) \approx \sum_{i=1}^{N} \lambda_i f_i(h,s) \quad (4)$$

where λ_i represents weighting of each $f_i(h,s)$ involved. These weights are found by minimizing a cost function that involves the sum of squared distances to all control values c_i, as well as the integral of the squares of the second partial derivatives of $\vartheta(h,s)$, serving as a smoothness term:

$$E = \sum_{i=1}^{N} \|c_i - \vartheta(h,s)\|^2 + \iint \left[\left(\frac{\partial^2 \vartheta}{\partial h^2} \right)^2 + \left(\frac{\partial^2 \vartheta}{\partial h s} \right)^2 + \left(\frac{\partial^2 \vartheta}{\partial s^2} \right)^2 \right] dh ds \quad (5)$$

6.2. Sonification of Texture Information

Sonifying both texture and color information is challenging in many ways. On the one hand we have to make sure that simultaneously played information is distinguishable, on the other hand we want to maintain a pleasing sound.

6.2.1. Acoustical Reliefs - Hearing Orientation Maps

We decided to utilize four more instruments, playing an octave below our keynote, to represent orientation maps of $\theta = \{0°, 45°, 90°, 135°\}$. The instruments we chose: Didgeridoo (0°), wood percussion (45°), uilleann pipes (90°), metal percussion (135°), create a hum alike sound at 0° and 90° and a percussion sound at 45° and 135°, to quickly distinguish horizontal and vertical from diagonal structures. Again, the framework allows exchanging instruments according to personal taste. To avoid *auditory masking* [35] we should guarantee that the Volume V_θ of all orientation map instruments are always lower than $\vartheta(h,s)$ for all h *and* s. The four orientation maps in between $\theta = \{22.5°, 67.5°, 112.5°, 157.5°\}$ are expressed using combinations of two neighbored *Orientation map* instruments, both playing at 50 % V_θ.

$V_{0°}(x,y) = d_4(x,y)$, $V_{22.5°}(x,y) = d_5(x,y)$, $V_{45°}(x,y) = d_6(x,y)$,
$V_{67.5°}(x,y) = d_7(x,y)$, $V_{90°}(x,y) = d_8(x,y)$, $V_{112.5°}(x,y) = d_9(x,y)$,
$V_{135°}(x,y) = d_{10}(x,y)$, $V_{157.5°}(x,y) = d_{11}(x,y)$

6.2.2. Audible Roughness - Sonification of Micro-Textures

Micro-Textures are sonified using one more instrument that has a vibrant temper. As [36] 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", which is an intuitive way to represent roughness acoustically.

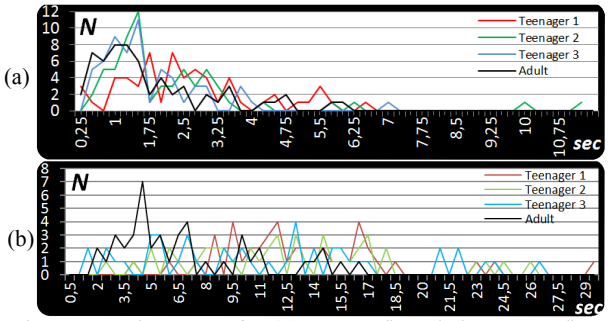


Figure 14: Histograms for (a) Exp. 1a/b and (b) Exp. 2a/b. N elements (y – *axis*) recognized in how many sec. (x – *axis*) each.

As anisotropic rough structures are visually salient and rarely occur in most environments, they are sonified more vibrant and at a Volume V_{Micro} being louder:

$$V_{Micro}(x,y) = \begin{cases} 0\%, & \text{if } d_{12}(x,y) = 0 \quad (\textit{smooth surface}) \\ 50\%, & \text{if } d_{12}(x,y) = 2 \quad (\textit{isotropic roughness}) \\ 100\%, & \text{if } d_{12}(x,y) = 1 \quad (\textit{anisotropic roughness}) \end{cases}$$

7. USER STUDIES

We did user studies on two groups of participants, following different motivations. First, as a proof of concept of our framework, we asked a congenital blind, 54 year old adult academic, who had acquired a geometric understanding and spatial sense throughout his life to solve several tests after 4 hours of training with our system. The participant was to solve three naming tasks at increasing difficulty:

- *Experiment 1a* was about identifying one out of four elements (orange, tomato, apple and lemon – as in Fig. 13 (a)) only by color while sonification of orientation maps and micro-textures was 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 was selected at random and displayed at an arbitrary position on the touch screen. This was achieved by selecting one out of 40 images (10 per object, with the object in different positions) at random. The task of the participant was to find and name the object. In the evaluation, 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 (Table 1 and Fig. 14). The average time to simply find an object's position on the screen was about 1.7 seconds. Chance level (pure guessing) is 25% in this experiment.
- *Experiment 2a* involved orientation maps and color. This time, the participant had to recognize one out 7 objects (orange, tomato, apple, banana, cucumber, carrot, lemon), as shown in Fig. 13 (a), so both color and shape are important for correctly naming the object. Again, each element was presented individually (chance level: 14%) at arbitrary positions and also in one of eight orientations, as illustrated in Fig. 13 (b). The database consisted of 56 images (8 for each element, varying position and orientation). Again, times were measured between finding and naming the object verbally, as shown in Fig 14 and Table 1.
- *Experiment 3* was about recognizing an object within a set of other objects. Therefore, we presented images like the

one shown in Fig. 13 (a) on the touch-screen. In our database of 7 images, we made sure that two objects of equal color (e.g. banana and lemon) would not be positioned next to each other. In each trial, an image was presented and the participant was told, which object he had to find, based on a random generator. This time, we measured the overall time until an element was named.

In Experiment 1b and 2b, we tested the system on a group of congenital blind 14 year old teenagers. Unlike the adult participant, they had little geometric understanding and sense of space. We 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. We performed an experimental evaluation of our system to measure their progress and compare it with the results of the adult participant. Three teenagers were trained about 5 hours with the system. This also included fundamental lecturing about basic geometry. Then, we asked them to perform Experiment 1b and 2b, which had the same setup as 1a and 1b described above. Surprisingly, the teenagers were able to perform the tests with similar hit rates and times as our adult participant (Table 1 and Fig. 14).

RESULTS	**P**	**Hit rate**		$\tilde{X}$	μ	σ
Experiments	**-**	**%**	**elem.**	**sec**	**sec**	**sec**
Exp. 1a	**A**	**100.0**	**60/60**	**1.3**	**1.8**	**1.4**
	T_1	91.6	55/60	2.2	2.6	1.5
Exp. 1b	T_2	93.3	56/60	1.5	2.3	1.9
	T_3	100.0	60/60	1.3	1.7	1.1
Exp. 2a	**A**	**93.3**	**42/45**	**5.6**	**7.0**	**3.9**
	T_1	88.8	40/45	12.1	13.3	4.7
Exp. 2b	T_2	93.3	42/45	11.9	12.5	5.5
	T_3	88.8	40/45	10.1	11.4	6.6
Exp. 3	**A**	**100.0**	**45/45**	**5.6**	**10.6**	**12.0**

Table 1. Hit rates and times (median $\tilde{X}$, mean μ, and standard deviation σ), for each trial and participant *P*.

8. CONCLUSIONS

We have presented a general framework and a sample implementation of a device that can support blind and visually impaired persons in exploring images or scenes. Many details of our implementation, such as the choice of local image descriptors, may be modified or improved further. However, we tried our best to design descriptors that are most promising from the theoretical and most informative from the practical point of view. The same is true for our sonification concepts: they are only one way how this can be achieved, yet we argue that it is an appropriate and powerful way to do it. The experimental results indicate that the system enables users to solve simple recognition tasks fast and reliably. In future experiments, we are planning to consider more and more difficult tasks with cluttered scenes and a wider variety of objects. Both the design of image descriptors and sonification concepts went through many experimental steps, and we discarded many alternative designs before we ended up with the solution that we present here. Feedback from the users was that they found the setup that we presented in this paper both intuitive and helpful. Still, it is our goal to start a fruitful discussion about 1: which features of an image are most informative in this framework, and 2: how can sonification convey as much relevant information to the user as possible. As we mentioned in the paper, there are also many possible extensions in terms of sensorics (cameras, tracking systems) and exploration paradigms. In future work, we are planning to continue to improve and extend our system along these lines. Our vision is to provide visually impaired persons with software for web browsers, image "viewers", and on portable systems such as smart phones.

9. ACKNOWLEDGEMENTS

We thank the Rheinischen Blindenfürsorgeverein, Düren, especially Marina, Larissa, Florian, Sascha and Mrs. Gut for their interest and participation in the project. We also thank Tobi and Rainer for their highly appreciated advisory support.

10. REFERENCES

[1] G. Kramer, B. Walker et. al.,"Sonification Report." in *ICAD,* 1999.
[2] F.Grond, S.Janssen et.al.,"Browsing RNA structures by interactive sonification," in *Proc. of ISon, 3rd ISW,* Stockholm, 2010.
[3] K. Vogt et al., "A Metaphoric Sonification Method – Towards the Acoustic Std. Model of Part. Physics," in *16th ICAD.,*USA,2010.
[4] J. Xu et al., "Sonification based Electronic Aid System for the Visually Impaired," in *JCIT,* vol. 6, no. 5, 2011.
[5] P. Meijer, "An experimental System for Auditory Image Representation," in *IEEE TBME.,* vol. 39, no.2, pp. 112-121, 1992.
[6] D. Margounakis et. al., "Converting Images to Music using their Color Properties," in *12th Int. Conf. on A.D., London* , 2006.
[7] K. Van den Doel, "Sound View: Sensing Color Images by Kinesthetic Audio," in *9th Int. Conf. on A.D.,* Canada , 2003.
[8] D. Payling et al., "Hue Music"in *13th Int. Conf. on A.D.,*CA, 2007.
[9] T. Hermann, "Taxonomy and Definitions for Sonification and Auditory Displays," in *Pro. 14th Int. Conf. on A.D.,* France, 2008.
[10] A. Hunt, T. Hermann, "The Importance of Interaction in Sonification," in *Pro. 10th Int. Conf. on A.D.,* Australia, 2004.
[11] D. Wagner et al., "ARToolKitPlus for Pose Tracking on Mobile Devices," in *Proc. of 12thCVWW,* Austria, 2007.
[12] E. Goldstein, *Sensation and Perception*, C. L. Emea, 2009.
[13] M. Lew, *Princ. of Visual Information Retrieval*, Springer, 2001.
[14] C. Tomasi, R. Manduchi, "Bilateral Filtering for Gray and Color Images," in *Proc. of Int. Conf. on CV,* India, 1998.
[15] J. Canny, "A Computational Approach to Edge Detection," in *IEEE TPAMI,* vol. 8, no. 6, pp. 679-698, 1986.
[16] B. Mandelbrot, *The Fractal Geometry of Nature*, H. Holt, 2000.
[17] B. Jähne, *Digital Image Processing,* Berlin: Springer, 2005.
[18] E. H. Adelson, P. J. Burt, "The Laplacian Pyramid as a Compact Image Code," in *IEEE TC,* pp. 284-299, 1983.
[19] D. Gabor., "Theory of comm.,"in *J. IEEE,*vol.93,pp.429-459,1946.
[20] M. Zhou, H. Wei, "Face Verification using Gabor Wavelets and Ada Boost," in *Int. Conf. on P.R.,* pp. 404-407, 2006.
[21] S. Suzuki et al.,"Top. structural analysis of digitized binary images by border following," in *CVGIP,* vol. 32, no. 1, pp. 32-46, 1985.
[22] R. Sedgewick, *Algorithms,* Addison Wesley, 1992.
[23] R. Haralick, "Statistical and structural approaches to texture," in *Proc. IEEE,* vol.67,no. 5, pp. 786 – 804, 1979
[24] M. Gippet, et al., "Haralick's texture features computed by GPU's for biological applications," in *IJCS*, 2009.
[25] P. Nelson, *Biological Physics*, Palgrave MacMillan, 2007.
[26] Y. Boykow, et al., "Efficient Approximate Energy Minimization via GraphCuts,"in *IEEE TPAMI,*vol.20,no.12,pp.1222-1239, 2001.
[27] Y. Kolmogorov, et al., "What Energy Functions can be minimized via Graph Cuts," in *IEEE TPAMI,*vol.26,no.2,pp.147-159, 2004.
[28] Y. Boykow, Y. Kolmogorow, "An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision," in *IEEE TPAMI,* vol. 26, no. 9, pp. 1124-1137, 2004.
[29] A. Rosenfeld, *Dig. Picture Processing*, Acad.Press, Orlando, 1982.
[30] S. Yeo, J. Berger, "Raster Scanning", in *Proc of ICMC,* 2006.
[31] T. Yoshida et. al., "EdgeSonic", in *Augm. Human Int. Conf.,* 2011.
[32] R. Ramlollet. al., "Constructing sonified haptiv line graphs for the blind student: first steps," in *4th Int. ACM CAT,* 2000.
[33] M. Russ, *Sound Synthesis and Sampling,* Focal Press, 2008.
[34] G. Donato, S. Belongie, "Approximation Methods for Thin Plate Spline Mappings and Principal Warps,"in *Proc of LNCS*,DK,2003.
[35] P. Gray, *Psychology,* Worth Publishers Inc., 2002.
[36] C. Seashore, Studies in the psychology of music Vol. 1: The vibrato, University of Iowa City, 1938.