Color-Audio Encoding Interface for Visual Substitution: See ColOr Matlab-based Demo

Juan Diego Gomez
Computer Science Department,
University of Geneva.
Route de Drize 7, 1227 Carouge,
Tel +41 (22) 379 0152
Juan.Gomez@unige.ch

Guido Bologna
Computer Science Department,
University of Geneva.
Route de Drize 7, 1227 Carouge,
Tel +41 (22) 379 0151
Guido.Bologna@unige.ch

Thierry Pun
Computer Science Department,
University of Geneva.
Route de Drize 7, 1227 Carouge,
Tel +41 (22) 379 0223
Thierry.Pun@unige.ch

ABSTRACT

Providing the blind with substitute visual perception is a relentless challenge confronting researchers of diverse areas. The See ColOr (Seeing Color with an Orchestra) system translates aimed-at regions of a color scene into 2D-spatialized sound signals represented by musical instruments. Associating sounds with colors is achieved thanks to an efficient quantization of the regions' HSL representation (Hue, Saturation, Luminosity). See ColOr can be used both for exploring static images and as a mobility aid. In this report, we introduce its general framework for color-audio encoding of images to provide visually impaired individuals with a low-cost application for color perception substitution, using the sense of hearing instead of sight. The experimental tests using a demo implementation have shown the feasibility and usefulness of this approach.

Categories and Subject Descriptors

J.6 [**Computer Applications**]: Computer-Aided engineering.

General Terms

Algorithms, Measurement, Experimentation.

Keywords

Visual substitution, color, sonification, instruments, HSL, Matlab.

1. INTRODUCTION

Losing the sense of sight is a very disabling condition that largely arises because of neurological factors and that affects millions of people due to injuries, diseases or birth defects. Perhaps, one of the most important aspects of the reality that surrounds us, which is imperceptible to the blind, is color. The real world is populated by colored objects; not surprisingly, colors are extremely important in both locating objects in space and identifying them, since colors are essential to objects' appearance.

Hopefully, Meers and Ward among others authors, assert that in general terms, colors are not exclusively perceived through the eyes but also thanks to other aspects related to the conscious and subconscious such as, symbolism, experiences and mostly, personal ways of perceiving [3]. This hypothesis sets up a large amount of pathways to address the lacking-color-perception problem. See ColOr uses instrument sounds to label colors instead of spelling out their names; this increases information transfer rate between system and user as well as offers an infinite gamut of perceivable colors [1].

Nowadays, particular attention is being paid in the research community to create new aids for assisting the blind. Hence, significant progresses have been reached in fields such as electronic and software visual substitution. For instance, the authors in [3] describe a method for achieving 3D color perception via haptic simulation to navigate familiar environments. Moreover, access technology such as screen readers and refreshable Braille displays enable the blind to use mainstream computer applications. The availability of assistive technology is increasing, accompanied by concerted efforts to ensure the accessibility of information technology to the blind.

This paper aims at describing a systematic approach to design a low-cost potential solution for those lacking of color perception, which transmits limited visual information through the auditory channel. Therefore, blind people can be trained, using a demo implementation, to classify colors based on specific associated sounds, allowing them to interpret the main components of a particular image for environment navigation. The testing demo and its open source matlab-code are available on *http://cvml.unige.ch/doku.php/members/juandiego*.

2. See ColOr Framework

The See ColOr approach accomplishes three main tasks for achieving an efficient encoding of designated image regions into sound sources. At the outset, uploading/managing of digital audio files to be used as sound sources, must be carried out independently (§2.2 *Sound driver*). Once the audio signals have been made available in memory, a graphical interface displays a chosen image and encodes user region selections based on an efficient quantization of their HSL-represented pixels (§2.1 *Color-sound encoder*). Finally, a specific processing of the encoded audio signal leads to its sonification (§2.3 *Sound player*). This process must be accomplished within the color-sound encoder every time a codified region is achieved.

The next sub-sections provide further details on the components of this framework, completing the previous top-down explanation. These components are described in a different order compared to the above, with the aim of providing a more coherent description of the system itself.

ASSETS'10, October 25–27, 2010, Orlando, Florida, USA.
ACM 978-1-60558-881-0/10/10.

2.1 Color-sound encoder

After uploading of an image, a user selects a pixel by clicking on it. A 25 pixels row from the image centered at the selected pixel is extracted, representing the region to be sonified[1]. Each of the 25 pixels is represented in HSL format and then encoded by labeling it with the index of a corresponding sound based on the following empirical quantization of the hue variable *H*:

1. Oboe for red ($0 \leq H < 1/12$)
2. Viola for orange ($1/12 \leq H < 1/6$)
3. Pizzicato violin for yellow ($1/6 \leq H < 1/3$)
4. Flute for green ($1/3 \leq H < 1/2$)
5. Trumpet for cyan ($1/2 \leq H < 2/3$)
6. Piano for blue ($2/3 \leq H < 5/6$)
7. Saxophone for purple ($5/6 \leq H \leq 1$)

In addition, a linear mixture of two instruments whose gains depend on the proximity of the two closest ranges is used. For instance, if *H* lies between yellow and green, the resulting sound is a linear combination of pizzicato violin and flute.

Moreover, in order to represent the degree of purity of the color, the saturation variable *S* is rendered by a sound pitch, which is assigned depending on four saturation values by means of four different notes:

1. C for ($0 \leq S < 0.25$)
2. G for ($0.25 \leq S < 0.5$)
3. B flat for ($0.5 \leq S < 0.75$)
4. E for ($0.75 \leq S \leq 1$)

Finally, when the luminance *L* is rather low (i.e. $L<0.5$), the pixel is represented by mixing the resulting sound from the above *H-S* encoding with a double bass, using one of the four possible notes (C, G, B flat or E) depending on the luminance level. The double bass is replaced by a singing voice (using the same possible notes) if $L > 0.5$. However, if the luminance *L* of the pixel is close to zero (the perceived color is black) only the double bass is retained. Similarly, when the luminance of the pixel is almost one, the resulting sound is an unmixed singing voice.

2.2 Sound Driver

To better understand the functioning of See ColOr and this stage in particular, knowledge of the actual amount of sounds to use is necessary. The encoding process requires a set of 36 sounds made up of a singing voice, a double bass and seven more instruments, each with four different pitches. Moreover, the goal is to sonify a row of 25 pixels by allocating a sound to each one of them. Thus, sounds must be associated in spatial correspondence with pixels. For instance, pixels at the leftmost position in the row are associated with sounds spatialized to the left, and pixels completely to the right are associated with sounds originating from the right.

To perform the 2D spatialization of sounds [4] there is the need of having 25 HRTF (Head-related Transfer Function) filtered versions of each sound. Hence, a total of 900 (36x25) predefined digital sounds are stored in memory to support the application. These digital files are all stereophonic and have a sample rate of 44.1 kHz. The sound driver loads the files, using the Matlab 'aviread' command, in concordance with the labeling strategy carried out by the encoder, and sets the signals to have a length of 302 ms and to 44x302 samples. The result of running the sound driver is a 16 bytes vector of 36x25x44x302x2 components housing the whole set of stereo-audios (two channels) required for the application to work properly in any potential case of color-audio encoding.

2.3 Sound Player

The resulting labeled vector from the color-sound encoder contains the indexes of the sounds associated with pixels. Now, a final sound signal must be generated based on this vector. This can be obtained by using the indexes into the vector as pointers to the components of the array obtained with the sound driver. Gaining access to those components is not trivial, since this array contains 25 versions per each of the 36 possible sounds. Therefore, once we point the right sound, one of its 25 versions must be chosen depending on the pixel's position we want to associate the sound with. The final signal is sonified with a frequency of 44.1 KHz, using the command 'wavplay' in Matlab.

3. CONCLUSION

A general framework for transforming target regions of a color-image into spatialized sound signals represented by musical instruments has been presented, providing a methodology to design a low-cost potential solution for those lacking of color perception. A free-demo application to demonstrate the usefulness of this approach for training blind people is provided on the web. This application has been used with embossed images presenting edges of obstacles [2] as well as for a visual substitution mobility prototype with stereoscopic vision for depth estimation [1].

4. REFERENCES

[1] Bologna, G. Deville, B. Pun, T. 2009. On the use of the auditory pathway to represent image scenes in real-time. *Neurocomputing Vol. 72, Issues 4-6: 839-849.*

[2] Bologna, G. Deville, B. Pun, T. 2007. Identifying major components of pictures by audio encoding of colors. *Lecture Notes in Computer Science Vol. 4528/ IWINAC (2): 81-89.*

[3] Meers, S. and Ward, K.. A vision system for providing the blind with 3D color perception of the environment. *Proc. Asia-Pacific Workshop on Visual Information Processing-Hong Kong, December 11-13, 2005: pag. 102-108.*

[4] Ordonez, C. Navarun, G. Barreto, A. Sound spatialization as a navigational aid in virtual environments. *Proc. 6th CSI Conference –Orlando Fl, July 14-18, 2002: pag. 297-302.*

[1] It is also possible to use high sampling resolution around the center pixel and lower the resolution at the borders of the row of 25 pixels. This approach replicates a crude model of the human vision.