Seeing the Movement through Sound: Giving Trajectory Information to Visually Impaired People

Nestor Z. Salamon, Julio C. S. Jacques Junior, Soraia R. Musse
Pontifícia Universidade Católica do Rio Grande do Sul

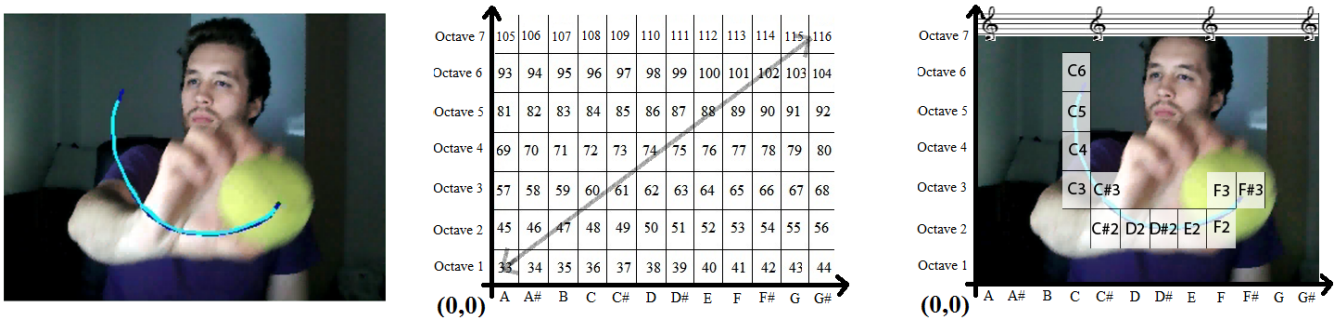


Figure 1: *From the tracked trajectory (left), we generate sound in real-time using a mapping function (middle). Illustration of the generated notes, based on the trajectory in the left image (right).*

Abstract

This paper presents a sonification model to convert object tracking information into sound in real time. The goal is to generate a sound that describes the information given by a trajectory - such as position, direction, velocity and shape - to help visually impaired people to "see" the world: how can we describe to them a square like we do by drawing it in a sheet of paper? The usage can be extensive: from audio games to computer-human interfaces. Experimental results are analyzed in practical tests using only audio with sighted people and the applicability of the model is discussed.

Keywords:: accessibility, games in education, virtual reality, computer human interaction

Author's Contact:

nzsalamon@gmail.com
juliojj@gmail.com
soraia.musse@pucrs.br

1 Introduction

Several approaches have been proposed for object detection and tracking in the last years [Yilmaz et al. 2006], with different levels of analysis and applications. Computer vision based methods can be very useful in many applications. For example, motion analysis is useful for video surveillance, pedestrian monitoring in traffic applications, measurement of athletic performance and virtual reality systems, among others.

The process of using non-speech audio to convey information is known as sonification [Kramer et al. 2010]. In this work we propose a sonification model which uses a transference function that receives a 2D coordinate position of a tracked object as input and generates a synthesized sound as output. The goal is to represent tracked trajectories as sounds to help visually impaired people - non birth-blind that already have a concept of shapes and movements formed - to "see" simple geometric shapes and understand movements in real-time while listening to the sound. Such model can be used as educational tool in geometry and arts lessons, for example, as an intermodal interactive environment in games, as computer-human interface and so on.

Combining object tracking with sound can give to a visually impaired person a new experience to recognize movements and be able to interact and play (with other people or computers) based on sound feedbacks. The object's trajectory here is extracted using a simple computer vision based approach [Braun et al. 2009], better described in the Section 3, and also could be used with other inputs like a mouse or touch screen devices. The properties of music theory are used in this work to make such relationship - trajectory $\times$ sound, with notes and octaves, associated to a pre-defined object/screen position, also presented in Section 3.

The main contribution of the proposed model is an intermodal mapping strategy that converts 2D trajectory information into musical notes, proposing a new way to show and explain trajectories and movements to visually impaired people through sound.

The rest of the paper is organized as follow: next we present some general human-computer interaction approaches and methods used to help visually impaired people as well as sensory substitution devices with similar purposes (Section 2). The proposed model is described in details in Section 3. Section 4 and Section 5 show some experimental results and our final considerations, respectively.

2 Related work

The access to information is a right for all. Some people, by physical limitations, are deprived of such access in different ways. This is the case of the visually impaired. In a study recently published in Nature [Amedi et al. 2007], the author uses a Sensory Substitution Device (SSD) to report that the lateral-occipital tactile-visual area - which is activated when objects are recognized by vision or touch - is also activated in sighted and blind humans using sound.

Nowadays there are some free software that perform screen reading allowing the visually impaired to interact with computers through speech synthesis, like *Dosvox*[1] and *Jaws*[2]. But how about giving information of image and videos? Previous research using specific devices and general frameworks with different techniques have already addressed this question, mainly through non-speech sound [Back et al. 1998; Levy-Tzedek et al. 2012b; Rampichini 2004; Banf and Blanz 2012].

In gaming field, the role of sound goes beyond background music and effects. The Karaoke is a classic example where the sound is used as input - the user sings and a score is given according to how melody is followed. Tic Tac Toe, as in [Targett and Fernström 2003], can be playble and entertaining using audio as output, with the sound feedback telling what is within each square. Several other audio game designs were proposed - from composing music to listening and hunting bears - but the field has not been fully explored yet [Parker and Heerema 2008].

[1] http://intervox.nce.ufrj.br/DOSVOX
[2] http://www.freedomscientific.com/products/fs/jaws-product-page.asp

In psychology, some works show that people are familiar with sounds related to shapes. As tested in [Maurer et al. 2006], adults map words with rounded vowels to rounded shape and words with unrounded vowels to angular shapes, showing that this correspondence between sound and object is not completely arbitrary. This mapping seems to be natural. On the other hand, with devices where the sound is computer generated, users must learn how the map from image to sound works.

A more intuitive approach that maps image to sound was proposed as Acousmetry [Rampichini 2004]. Acousmetry is neologism to establish a two-way link between graphic and acoustic sign, based on the music-image relationships of volume to distance, frequency to vertical position and stereo to horizontal position. Acousmetry presents a new linguistic code focusing artistic creation with visual and sound perceptions.

To address helping visually impaired people, the mappings are a little more specific. The SSD used to compare sighted and blind cerebral activity in [Amedi et al. 2007] encodes image shape, shade and texture, mapping vertical locations to frequency, horizontal locations to stereo (pan), and brightness to loudness, sounding the columns of an image sequentially from left to right, as defined in [Back et al. 1998]. As result, user hearing this sound can identify objects and features in the environment image, guided by the sound [Merabet et al. 2009; Levy-Tzedek et al. 2012a; Striem-Amit et al. 2012].

In [Levy-Tzedek et al. 2012b], the *EyeMusic* SSD was applied to study cross-sensory transfer of sensory-motor information, mapping picture brightness to loudness, colors to instruments and the vertical position of the pixel to pitch.

In [Doel et al. 2004], *SoundView* converts image onto a virtual surface based on color and user's movement with a pointer over the rough colored image texture. In [Banf and Blanz 2012] - extended in [Banf and Blanz 2013b; Banf and Blanz 2013a], the authors give information about color, textures, orientation and edges in still images through MIDI sounds, so users can explore and analyze the image content. A similar work [Sanchez 2010] uses a mouse/pointer over a shape and generates a sound as the user approaches the curves - sound intensity increases as the distance to the curve decreases.

The common characteristic between these mappings is that the user may take a long time in the learning step to understand how the mapping works. In [Striem-Amit et al. 2012], for example, a 10-hours training was applied to teach users to read using an SSD. These techniques are useful analyzing image content, searching for features and characteristics, but the authors do not focus on movements and relationships in image sequences.

In this paper we explore a different approach to give information to visually impaired people: like we usually draw paths, lines or shapes in a paper - or even in the air - to interact or better explain an idea to someone, these lines are converted into sound in real time using a relationship between its position and a specific sound frequency. The proposed model is described next.

3 The model

This section describes the proposed approach to generate sound based on a tracked trajectory as well as the computer vision technique used to perform it.

3.1 Object tracking

We use the *CVMouse* proposed in the work of Braun et al. [Braun et al. 2009] to track a colored object in a controlled environment. The objective of *CVMouse* is to work as a pointer/scratch interface improving the human-computer interaction. Using a webcam and computer vision algorithms, *CVMouse* is able to simulate the states of the mouse. However, in this work we use only the captured position (x, y coordinates) of the object in time. *CVMouse* was developed to work with the YCbCr color space as color representation, but probably many other color spaces could be used, such as HSV or Lab. The idea is to use a color invariant feature to be more robust with illumination changes, as mentioned by the authors.

Learning the color model The color of the object to be tracked is learned using the first f frames of the video (as in [Braun et al. 2009]). Basically, for each pixel coordinate, the median and standard deviation of the channels Cb and Cr are computed. It is important to notice that the color of the adopted object must not be visible in any other location of the scene, once the tracking system will track the largest object with the detected color. On the other hand, it does not need a fixed camera, like usual background subtraction models, because the model will track a specific color pattern. Let S be a vector (with size = f) composed of the pixel values captured over a predefined region, in the Cb channel. Firstly we compute their median and standard deviation values (λ and σ). As described in [Braun et al. 2009], some outliers are removed and then the color model is composed of $C(\lambda_1, \sigma_1, \lambda_2, \sigma_2)$, where λ_1 is the median of channel 1 and σ_1 is their standard deviation (respectively for the second channel).

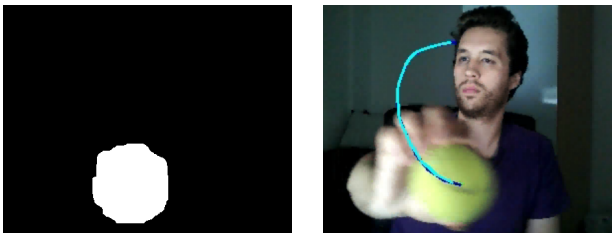
Segmentation and tracking For each pixel (x, y) in the whole image captured from the camera, the absolute difference is computed over the median values (for the channels Cb and Cr). We define the foreground pixels as the ones whose difference are lower than a predefined threshold (K = 6, obtained by experiments), as seen in Equation 1.

$$B_{Cb(x,y)} = \begin{cases} 1 & \text{if } ||Cb(x,y) - \lambda_1|| < K\sigma_1 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The final binary image is composed of an AND operator over the two channels (B_{Cb} and B_{Cr}). Some morphological operators (closing and opening) are used to eliminate small artifacts and to close small holes in the resulting binary image. Illumination changes, shadows, and other factors could influence the background subtraction results, so the obtained trajectory unfortunately can be noisy. To deal with it, we apply the moving average filter for each spatial coordinate (x and y) at each time step (n), as described on the Equation below:

$$x'(n) = \frac{1}{4}x(n) + \frac{1}{4}x(n-1) + \frac{1}{4}x(n-2) + \frac{1}{4}x(n-3), \quad (2)$$

where x is the original data. Figure 2 illustrates the object segmentation and the tracking result.



(a) **(b)**

Figure 2: *Illustration of object segmentation (a) and tracking (b).*

3.2 Music from trajectory

Music is an order that is constructed of sounds and noise. The musical notes are used to represent the music. Each of the seven musical notes in the diatonic scale - *C*, *D*, *E*, *F*, *G*, *A* and *B* - and its semitones - *C*♯, *D*♯, *F*♯, *G*♯ e *A*♯ - is associated with one frequency. This frequency must be between 20 *hertz* and 20 *kilohertz* to be audible to humans and represent, respectively, the bass and treble sounds.

The frequency of each note can vary according to the octave in which it is played: a *C* note, for example, has the frequency of 32.70 *hertz* in the second octave (in a scale 0 - 10). In the third

octave, it will double its frequency (65.40 *hertz*). Also, each instrument (or human voice) is responsible for one timbre, which will produce notes on a given frequency spectrum.

Such properties can be synthesized by a computer to generate sounds through a protocol like *Musical Instrument Digital Interface* (MIDI). As specified in MIDI Protocol[3], each note in its particular octave has a *MIDI number* from 0 to 128, programmatically played. A relationship between the data obtained in the image processing stage (object tracking) and these *MIDI numbers* can so be established.

We designed a normalized mapping to deal with different image resolutions. All coordinates of the image received from the tracker are converted to a point in the Cartesian plane, according to the ratio between image resolution and plane dimension, as the number of *MIDI number* is relatively small when compared to the image resolution. Each point in this plane will be mapped to a note - or semi-tone - in the X axis, starting from *A* (x = 0) to $G\sharp$, and to an octave, in the Y axis, starting from Octave 1 (y = 0) to Octave 7, as shown in Figure 3(a). So we have 84 *MIDI numbers*, 12 notes in 7 octaves. Starting from *A*=33 MIDI (excluding some too bass sound), the plane specifies a range that encompasses frequencies from 55 *hertz* to 6.64 *kilohertz*.

We can imagine such frequency-associated mapping by bass sounds on bottom-left (Cartesian plan origin) and treble sounds on top-right (X-max, Y-max) extremity, as illustrated in Figure 3(b).

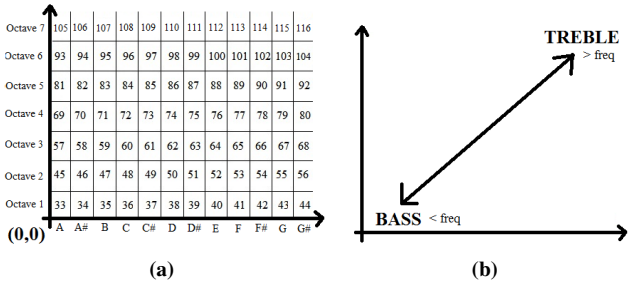


Figure 3: *Displacement of the musical notes (through its MIDI Number) according to the Cartesian plane X-note and Y-octave (a) and bass-treble (frequency) map result (b).*

As the number of acquired points of each tracked trajectory is associated to the frame rate of the tracker (in this case 30 frames per second), we need to skip some points to get a more spaced and distinct sound. Our algorithm handles these points according to the angle of the vectors computed in the head of the trajectory: smaller the angle variation is (approximately a straight line), more spaced the sounds are; greater the angle variation, faster the sound. This feature highlights angle changes in the trajectory, so the user can figure out more accurately the new direction, build an image of the movement and, in the best case, memorize where (and when) it has occurred. Let us define the variables used to process the trajectory:

- t, the frame ahead of the trajectory, the current captured point in computer vision step;
- $buffer_t$, buffer with the last N_{pmax} captured points from where the sounds are going to be played. Indeed, $buffer_t$ is associated with frame t since it represents a memory of last tracked points before t. Currently, $N_{pmax} = 15$, meaning that frame t plus 15 points, from frame $t' = t - 15$ to t, are stored in $buffer_t$;
- t', $buffer_t$ includes points from t' to t;
- α_t, the angle computed using some of the points (N) in the $buffer_t$ that should represent the curvature of the trajectory at the frame t; and
- S_p, the variable spacing between the sounds.

[3] http://www.midi.org/techspecs/

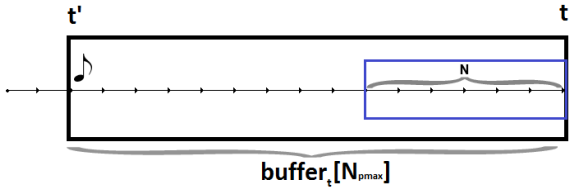


Figure 4: *The $buffer_t$ and the variables used to compute the angles and play the sounds. The N points in the head of the trajectory will represent the curvature of the trajectory.*

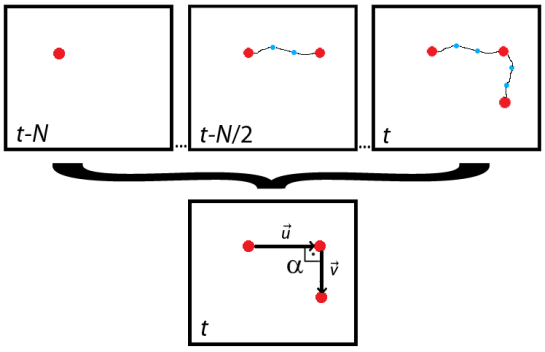


Figure 5: *The angle between vectors $\vec{u}$ and $\vec{v}$ being calculated at frame t plus the first $N = 6$ points of $buffer_t$.*

Figure 4 illustrates the variables used to play the sounds. Variables α_t and S_p are explained in the next paragraphs, as do N_{pmax} and N and their values.

Angle computation Consider the current object position at frame t and the object positions in the last N frames, it is, the N+1 top points inside the $buffer_t$. We use an early captured point $ep = (t - N)$, the point in the middle $mp = (t - N/2)$ and the last one (pt on time t) to create the vectors $\vec{u} = (mp - ep)$ and $\vec{v} = (pt - mp)$. The smallest angle between $\vec{u}$ and $\vec{v}$ is computed to measure the angle variation (α_t) at the current frame t. The N parameter is configurable to determine how smooth the trajectory variation will be represented. If N is too high, some small curves can disappear. If it is too small, little variances will make a big change in the angles. In this model we calculated $\vec{u}$ and $\vec{v}$ as illustrated in Figure 5, with N = 6, trying to keep the main variations and discarding unwanted imprecisions in the drawn trajectories. In the next section it is detailed how the α_t is used to define how faster the sound will be played, it means, the sound spacing between notes.

Sound spacing To start this step, we need to define N_{pmax}. The N_{pmax} parameter is a ratio between the frame rate of the tracker and the velocity we want the sound. We choose $N_{pmax} = 15$, using a frame rate of ~30 frames per second, assuming it will play, at minimum, 2 notes per second - which we considered not too slow/fast to detect sound variations. With different frame rates or very fast/slow trajectories drawn, N_{pmax} can also be configured to adjust the sound velocity according to our needs. However, these 2 notes per second can be changed to indicate a high curvature of the trajectory. That is why we check for each computed α_t if the sound spacing, here represented as variable S_p, should be increased/decreased or not.

As the time passes by, for each frame t, we create the $buffer_t$ with t and the last N_{pmax} captured points, calculate α_t and define how many points should be played from such buffer. In other words, we will play a sound at each S_p points inside the $buffer_t$ (from t' to t), according to the α_t obtained with the top $N+1$ points of this buffer. Table 1 shows the S_p value defined as a function of α_t interval and N_{pmax} value. The second column shows the spacing, in points (S_p), between each played note, according to the computed angle α. The third column shows de quantity of notes played per second in each case, using $N_{pmax} = 15$. Summarizing, angles smaller than

Table 1: *The sound spacing according to the angle α and the quantity of notes played per second.*

Angle (α)	S_p	Notes per second
$\alpha \leq 35^\circ$	N_{pmax}	2
$35^\circ < \alpha \leq 45^\circ$	1	30
$45^\circ < \alpha \leq 90^\circ$	N_{pmax}/5	10
$90^\circ < \alpha \leq 150^\circ$	N_{pmax}/3	6
$\alpha > 150^\circ$	N_{pmax}	2

35° or greater than 150° generate more spaced and, consequently, slower sounds, i.e. 2 notes played in one second. Angles from 35° to 45° generate 30 notes (all captured points) played in one second, and successively. These values have been empirically defined to sound faster in the angles greater than 35° or smaller than 150°, as we considered these angles more important to highlight changes in trajectory direction.

If α_t and α_{t-1} lie in different intervals, considering the first column of Table 1, S_p is updated and two conditions should be verified:

- if $S_p < N_{pmax}$ (happens when $35^\circ < \alpha \leq 150^\circ$), the points inside $buffer_t$ (from t' to t) are played at each S_p points until the $buffer_t$ is consumed, and no new S_p values will be updated. It implies that no other variation will influence in the sound spacing until the end of $buffer_t$. Afterwards, new t' is the current t, new t is the current frame processed by computer vision (the head of the trajectory) and S_p can now be updated according to the new α_t.
- if $S_p = N_{pmax}$ (happens in remaining cases), the S_p is always updated. The difference is that, in this case, we let S_p causes sound spacing variation inside the $buffer_t$. In other words, cases when $S_p = N_{pmax}$ were considered less important to characterize angle variations and can be interrupted.

This way we can listen to faster or slower sounds as the trajectory changes its direction. In the next section we analyze the results obtained using the model in an experimental test, asking for information based on the sound presented.

4 Experimental Results

We validate our technique through a test session applied to sighted people. Although the users could see the trajectories, we choose by apply it like they were visually impaired: the learning step and the experiments are completely blind, using just sounds to explain and test, without showing the mapping nor trajectories or images.

The test session has the following framework: a learning step would teach user how the mapping works. Then, three experiments - with three questions each - are applied to evaluate the user's understanding of the trajectories heard.

4.1 Learning

The learning process consists of a series of instructions and samples where trajectories, directions and angles are explained to the users. Users need to imagine and memorize the positions of the notes, understanding the sound distribution in the Cartesian plane.

A 7-minute audio was given explaining how the map works, teaching what is a bass and a treble sound. The audio so explains the Cartesian plane with its notes in the X axis and octaves in the Y. Sampling this map with computer-generated lines, three horizontal lines are played forward and backward, one in the bottom (Octave 1), one in the middle (Octave 4) and one in the top of the screen (Octave 7) showing all notes within an octave, as shown in Figure 6(a). Then three vertical lines are played, one in the left extremity (note *A*), one in the middle (*D*) and one in the right extremity of the screen ($G\sharp$), showing the note difference between octaves (see Figure 6(b)). This sample address to teach sound variances between low and high notes in the axes.

In the last step of the learning process, the user is trained about the angles. A horizontal line parallel to the X axis in the bottom of the screen, followed by a vertical line in the right extremity of the screen is played, making the sound of a 90° angle, as illustrated in Figure 6(c). Once heard these steps, the user is encouraged to start the experiment.

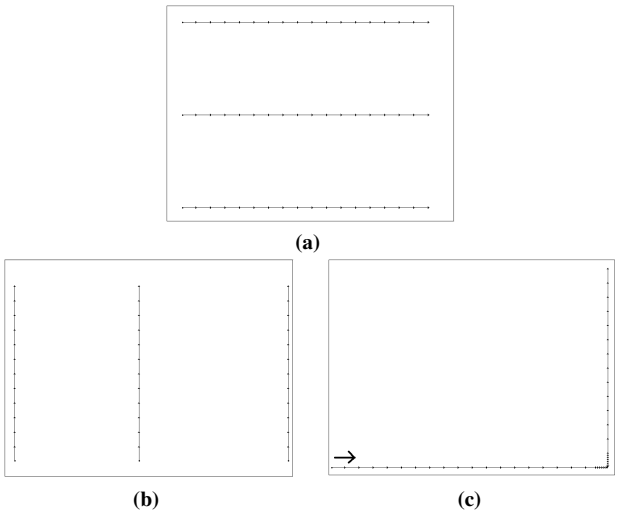


Figure 6: *The three steps of the learning process: i) the computer-generated horizontal lines in the first, fourth and seventh octaves (respectively, from bottom to top), played from left to right and from right to left (a); ii) the computer-generated vertical lines with the notes A, D and G♯ (respectively, from left to right), played from bottom to top and from top to bottom (b); iii) the computer-generated lines making a 90° angle, with the arrow showing the beginning of the trajectory and its direction (c).*

4.2 First experiment

The first experiment was composed of three challenges involving straight lines. These lines are computer-generated (totally straight) and the user was asked about horizontal or vertical, left/right and up/down movements. Each line was played three times. The first line was horizontal, crossing the screen from right to left, in the sixth octave, as show in Figure 7(a). The second line, shown in Figure 7(b), was another horizontal line, but now from left to right and played in the second octave. The last line was a vertical one, played with a *C*, bottom-top, as shown in Figure 7(c). After each sequence of repetitions, the user should answer horizontal or vertical and the direction of the trajectory.

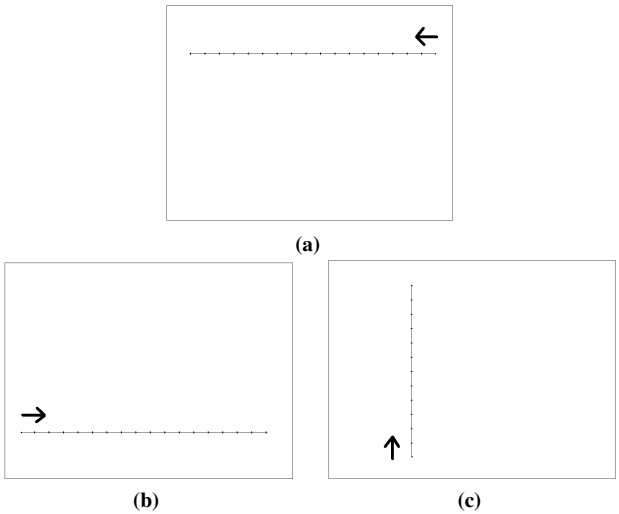


Figure 7: *First experiment: the computer-generated horizontal lines in the sixth octave (a), in the second octave (b) and a vertical line with the note C (c). The arrows show the beginning of the trajectories and their directions.*

4.3 Second experiment

In the second experiment the user is asked about the geometric shape represented by the sound of the trajectory. Three shapes were drawn in the air, using the image tracking described in Section 3 to retrieve the points and generate the sound. As the trajectory of each shape is continuous, it will have angles and may have a little variance inherit from the trajectory drawn (a not precise draw). In each figure, the trajectory was recorded in order to apply the same sound for every user. Four alternatives were given in each sound. After three repetitions of each sound, the user can choose the corresponding alternative.

In the first shape, we drawn a triangle (Figure 8(a)), making a sound with three main variances in the angles; the alternatives were triangle, rectangle, circle and hexagon. The objective here was investigate if the user can distinguish the quantity of angles and then induce the shape, as each alternative has a different number of angles.

In the second shape, a circle (Figure 8(b)) was played in order to show how it works in a shape without angles; the alternatives were square, rhombus, circle and triangle. Here we tested if the user could induce the shape in the alternatives by the absence of angles.

The last shape was a rectangle (Figure 8(c)) and the alternatives were rhombus, circle, square and rectangle. Here alternatives with the same number of angles were given, with the objective of evaluate if the direction of the trajectory can be understood.

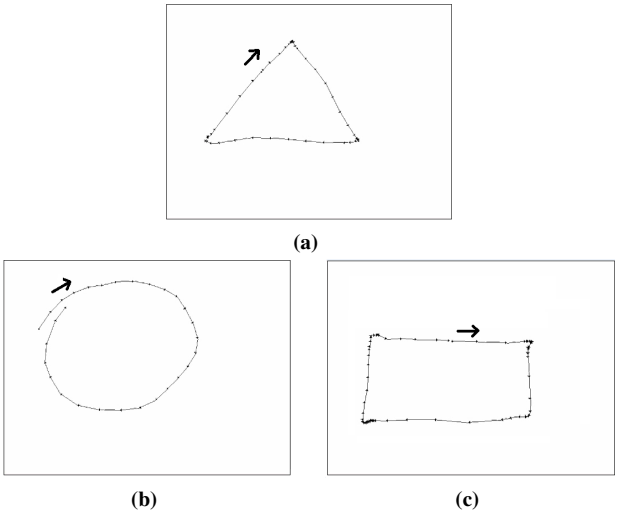


Figure 8: *Second experiment: the hand-drawn trajectory retrieved from image tracking generating the sound of a triangle (a), a circle (b) and a rectangle (c). The arrow shows the beginning of each trajectory and its direction.*

4.4 Third experiment

The last experiment was based on letters drawn in the air: retrieving its trajectory from the tracking and converting it into sound. Each sound will represent a specific letter and, once it depends on how the user draws it, this experiment is more challenging. We choose three random letters - T, W and Z, as shown in Figure 9. Our objective here is to see if the fast training is enough to understand generic trajectories. All letters are continuous and the same principle of the angles is applied. It was prompted that letters (any from the latin alphabet) would be shown and, after listening three times each letter sound, the user would type the letter heard without alternatives. None draw-pattern was teached.

These three experiments were applied to 23 people in a 16-49 age group, without criteria of schooling or gender, although none of them was musician. The recorded audio containing learning and test was distributed via email. The whole process should have took about 22 minutes for each user and the answers were sent through online form or email. None of the answers was required, although

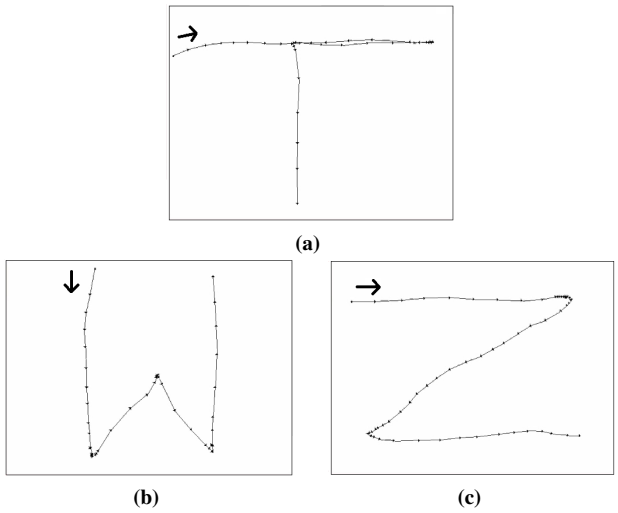


Figure 9: *Third experiment: the hand-drawn trajectory retrieved from image tracking, generating sounds corresponding to the letters T, W and Z. The arrow shows the beginning of each trajectory and its direction.*

users were encouraged to answer all the questions. The obtained answers are presented and discussed next.

4.5 Discussion

In the first experiment we had the following percentage of correct answers: 56.5% of the users answered correctly “horizontal” and “to the left” in the first line; 82.6% answered correctly “horizontal” and “to the right” in the second line and 86.9% were correct answering “vertical” and “up” in the third line. It shows that “vertical” and “horizontal” trajectories could be easily understood in straight lines and direction can be affirmed with relative certainty. Only in the second line we had blank answers: 8.7%.

In the second experiment, the first shape, a triangle, was guessed by 65.2% of the user. The second, a circle, had 78.3% of hits. The third, a rectangle, 26.1%. As the directions may be harder to understand because of deformities in trajectory drawn in the air, the main point here was the angle. In the triangle, the users could analyze the sound through the three main variances in the trajectory and in the circle, by absence of them. The last shape, the rectangle, had a greater miss with the users confused between rectangle and rhombus (52.2%), fact we assign to the small rectangle drawn in the air. The rate of blank answers obtained in the triangle, circle and rectangle were, respectively, 4.3%, 0% and 8.7%.

The last experiment was, in fact, more challenging with very different results. We had 8.7% of correct answers guessing the first letter (T), 17.4% in the second (W) and 43.5% in the third (Z). The rate of blank answers was more significant here: 21.7% (T), 17.4% (W) and 26.1% (Z). The main point probably was that we could not teach a pattern of how the letters were drawn in function of the short training. The letter T was continuous drawn from left to right, backing to the middle and going down. The backward trajectory in the top to the middle of the letter could be hard to understand and probably too fast to show the difference between the middle of the T and its right extremity. The second letter is not a common letter in the brazilian alphabet (country in which the test was applied) and could be one of the reasons of the high percentage of mistakes. Another cause could be the diagonal lines in this letter that are hardly perceived. The last letter had a greater rate of correct answers, probably because the start and the end of the letter are straight lines and they make two angle variances.

Table 2 summarizes the experimental results of the test session. The second column of the table shows the % of correct answer (correct alternative/letter) given by the users in the test session (described in details earlier). The third column shows the % of participants that left the answer blank or did not know the answer.

Table 2: *Experimental results: the lines of the table describe, each three, one experiment.*

Experiment	Correct answers	Blank answers
Horizontal line, left (←)	56.5%	0%
Horizontal line, right (→)	82.6%	8.7%
Vertical line, up (↑)	86.9%	0%
Triangle	65.2%	4.3%
Circle	78.3%	0%
Rectangle	26.1%	8.7%
Letter T	8.7%	21.7%
Letter W	17.4%	17.4%
Letter Z	43.5%	26.1%

As seen in this test session, considerable results could be obtained with a fast and basic training: directions could be easily perceived and angles be helpful to distinguish simple shapes and direction changes. There is a lack to improvement in more complex shapes or letters, what we believe could be achieved with bigger draws and establishing a pattern of how the trajectories are made. We also believe the hit rate could be improved as a function of learning time, although it will generate a problem with the participants' engagement.

5 Final considerations

In this paper we proposed a method to retrieve image tracking information about trajectories and convert it into sound, making them accessible to visually impaired (non birth-blind) people. We believe the results were satisfactory for simple trajectories: lines and basic geometric shapes could be distinguished with a fast learning step, based on the discussion presented in Section 4.

Taking only 7 minutes, it is a few time when compared with other software that generate sound from image, like in [Striem-Amit et al. 2012], with an average of 73 hours for the complete training. It is important to highlight that the objectives of [Striem-Amit et al. 2012] are quite different from what we are proposing, even with the same final objective of help people with special needs.

A low engagement rate could have decreased the result rate as sighted people may not be enough interested in the project to keep focused while answering the questions. Also, the participants had different ages and schooling, maybe were not familiar enough with musical notes or Cartesian planes. A custom training may get each participant committed, analyzing the ratio between training time and engagement.

In further works we intend to evaluate this model with specific trainings to totally blind people, really interested in the experiment - it was not possible to get a significant quantity of non birth-blind to participate in these experiments. We also believe that the model could be upgraded using multiples timbres - each in a quadrant - in a bigger Cartesian plane. The applicability of this model can address some educational and entertainment issues involving people with special needs, making available a new way of interaction to share and explain details of trajectories, movements and gestures.

Acknowledgements

Authors would like to thank Brazilian agencies FAPERGS and CAPES for their financial support.

References

AMEDI, A., STERN, W. M., CAMPRODON, J. A., BERMPOHL, F., MERABET, L., ROTMAN, S., HEMOND, C., MEIJER, P., AND PASCUAL-LEONE, A. 2007. Shape conveyed by visual-to-auditory sensory substitution activates the lateral occipital complex. *Nature neuroscience 10*, 6, 687–689.

BACK, M., COOK, P. R., BARGAR, R., MEIJER, P. B. L., AND MYNATT, E. 1998. Listen up! realtime auditory interfaces for the real world (panel). In *ACM SIGGRAPH 98 Conference abstracts and applications*, ACM, New York, NY, USA, SIGGRAPH '98, 182–184.

BANF, M., AND BLANZ, V. 2012. A modular computer vision sonification model for the visually impaired. *Proceedings of the 18th International Conference on Auditory Display*.

BANF, M., AND BLANZ, V. 2013. Man made structure detection and verification of object recognition in images for the visually impaired. In *Proceedings of the 6th International Conference on Computer Vision / Computer Graphics Collaboration Techniques and Applications*, ACM, New York, NY, USA, MIRAGE '13, 18:1–18:8.

BANF, M., AND BLANZ, V. 2013. Sonification of images for the visually impaired using a multi-level approach. In *Proceedings of the 4th Augmented Human International Conference*, ACM, New York, NY, USA, AH '13, 162–169.

BRAUN, H., HOCEVAR, R., QUEIROZ, R., COHEN, M., MOREIRA, J., JACQUES, J., BRAUN, A., MUSSE, S., AND SAMADANI, R. 2009. Vhcve: A collaborative virtual environment including facial animation and computer vision. In *Games and Digital Entertainment (SBGAMES), 2009 VIII Brazilian Symposium on*, 207–213.

DOEL, K. V. D., SMILEK, D., BODNAR, A., CHITA, C., CORBETT, R., NEKRASOVSKI, D., AND MCGRENERE, J. 2004. Geometric shape detection with soundview. In *Proceedings of the 10th International Conference on Auditory Display (ICAD2004)*.

KRAMER, G., WALKER, B., BONEBRIGHT, T., COOK, P., FLOWERS, J. H., MINER, N., AND NEUHOFF, J. 2010. Sonification report: Status of the field and research agenda. *DigitalCommons@University of Nebraska - Lincoln*.

LEVY-TZEDEK, S., HANASSY, S., ABBOUD, S., MAIDENBAUM, S., AND AMEDI, A. 2012. Fast, accurate reaching movements with a visual-to-auditory sensory substitution device. *Restorative Neurology and Neuroscience 30*, 4, 313–323.

LEVY-TZEDEK, S., NOVICK, I., ARBEL, R., ABBOUD, S., MAIDENBAUM, S., VAADIA, E., AND AMEDI, A. 2012. Cross-sensory transfer of sensory-motor information: visuomotor learning affects performance on an audiomotor task, using sensory-substitution. *Scientific reports 2*.

MAURER, D., PATHMAN, T., AND MONDLOCH, C. J. 2006. The shape of boubas: Sound–shape correspondences in toddlers and adults. *Developmental science 9*, 3, 316–322.

MERABET, L. B., BATTELLI, L., OBRETENOVA, S., MAGUIRE, S., MEIJER, P., AND PASCUAL-LEONE, A. 2009. Functional recruitment of visual cortex for sound encoded object identification in the blind. *Neuroreport 20*, 2, 132–138.

PARKER, J. R., AND HEEREMA, J. 2008. Audio interaction in computer mediated games. *International Journal of Computer Games Technology 2008*, 1.

RAMPICHINI, F. 2004. *Acusmetria: il suono visibile*, vol. 37. Franco Angeli.

SANCHEZ, J. 2010. Recognizing shapes and gestures using sound as feedback. In *CHI'10 Extended Abstracts on Human Factors in Computing Systems*, ACM, 3063–3068.

STRIEM-AMIT, E., COHEN, L., DEHAENE, S., AND AMEDI, A. 2012. Reading with sounds: Sensory substitution selectively activates the visual word form area in the blind. *Neuron 73*, 3, 640–652.

TARGETT, S., AND FERNSTRÖM, M. 2003. Audio games: Fun for all? all for fun. In *ICAD*.

YILMAZ, A., JAVED, O., AND SHAH, M. 2006. Object tracking: A survey. *ACM Comput. Surv. 38*, 4 (Dec.).