Automatic Extraction of Pinna Edges for Binaural Audio Customization

Simone Spagnol [#1], Davide Rocchesso [*2], Michele Geronazzo [#3], Federico Avanzini [#4]

[#] *Department of Information Engineering, University of Padova*
Padova, Italy
[1] spagnols@dei.unipd.it
[3] geronazzo@dei.unipd.it
[4] avanzini@dei.unipd.it

[*] *Iuav University of Venice*
Venice, Italy
[2] roc@iuav.it

***Abstract*—The contribution of the external ear to the head-related transfer function (HRTF) heavily depends on the listener's unique anthropometry. In particular, the shape of the most prominent contours of the pinna defines the frequency location of the HRTF spectral notches along the elevation of the sound source. This paper addresses the issue of automatically estimating the location of pinna edges starting from a set of pictures produced by a multi-flash imaging device. A basic image processing algorithm designed to obtain the principal edges and their distance from the ear canal entrance is described. The effectiveness of the developed hardware and software is preliminarily evaluated on a small number of test subjects.**

I. Motivation and Research Background

The soundwaves produced by everyday sound sources are subject to diverse transformations along their path towards the listener's eardrums. One critical transformation is provided by the listener himself: as a matter of fact, sound waves are influenced by the active role of the listener's body, thanks to which he/she can collect salient information on the spatial location of the sound source. Auditory cues produced by the human body include both binaural cues, such as interaural level and time differences, and monaural cues, such as the spectral coloration resulting from filtering effects of the external ear. All these features are summarized into the so-called *Head-Related Transfer Functions (HRTFs)* [1], i.e. the free-field compensated frequency- and space-dependent acoustic transfer functions between the sound source and the eardrum. Binaural spatial sound can be synthesized by convolving an anechoic sound signal with the corresponding left and right HRTFs, and presented through a pair of suitably compensated headphones.

HRTFs are strictly personal. When individual HRTFs are used, the direction of a simulated sound source is perceived by the listener almost as precisely as a real sound source placed in the same direction [2]. Unfortunately, obtaining personal HRTF data strictly requires specific hardware, anechoic spaces, and long collection times [3]. On the other hand, when non-individual HRTFs are used localization errors such as front/back reversals, elevation angle misperception, and inside-the-head localization are commonly experienced [4], [5]. The reason for such bad behaviour of non-individual HRTFs mainly resides in the listener's unique anthropometry, and especially into the shape of the auricle (or pinna).

The pinna plays a fundamental role in the shaping of HRTFs. It introduces peaks and notches in the high-frequency spectrum, whose center frequency, amplitude, and bandwidth greatly depend on the elevation angle of the sound source. The relative importance of major peaks and notches in typical HRTFs in elevation perception has been disputed over the past years; still, both seem to play an important function in vertical localization of a sound source. Recently [6], [7] the authors found that while resonance peaks are similar among different subjects, frequency notch locations are critically subject-dependent. In the same works, a simple ray-tracing law was used to strengthen the hypothesis that in frontal median-plane HRTFs the frequency of spectral notches, each assumed to be caused by a single reflection path, is related to the distance of the most prominent pinna edges to the ear canal (or meatus) entrance.

Such finding allows for a very attractive approach to the parametrization of the HRTF based on individual anthropometry, i.e. extrapolating the most relevant parameters that characterize the HRTF spectral shape from a representation of the principal pinna edges, which need to be in turn estimated from a picture. Having outlined this basic yet unexplored idea, the challenge addressed by this paper resides in the computational image processing side of our research flow and may be summed up in the following question: *how to automatically derive a robust representation of the most prominent pinna edges from one or more side face pictures of a person?*

II. Pinna Edge Extraction: Research

It is commonly accepted that no two human beings have identical pinnae, and that the structure of the pinna does not

MMSP'13, Sept. 30 - Oct. 2, 2013, Pula (Sardinia), Italy.
978-1-4799-0125-8/13/$31.00

change radically over time [8]. These two simple statements are at the basis of a recently growing interest in the field of biometrics in using the pinna as an alternative to face-, eye- or fingerprint-based subject recognition. A multitude of works addressing ear biometrics has surfaced in the last 15 years starting from Burge and Burger's rediscovery [9] of the work of Iannarelli [8], the pioneer of ear recognition systems. These new works generally address the study and design of all the building blocks making up a complete recognition system, including ear localization in images or video, feature extraction and matching. A comprehensive review of the state-of-the-art in ear biometrics up to 2010 can be found in [10].

Radically different approaches to the definition of a feature model that uniquely and robustly defines the pinna morphology from 2D or 3D [11] images in ear recognition systems have been proposed. Some of these directly transport the input ear image to a different domain, e.g. using a 3D elliptic Fourier transform [12] that compactly represents the pinna shape or a force field transformation [13] that treats pixels as an array of mutually attracting particles acting as the source of a Gaussian force field. Others extract an edge map from the original image containing the pinna; thanks to such map either the pinna is localized into the image or distinctive features are extracted. Since we are interested in the extraction of pinna contours from 2D images, we now give a brief review of these latter approaches.

The most obvious way of extracting edges from a generic image implies the use of standard intensity edge detection techniques such as the Canny method [14]. This method was exploited by Burge and Burger [9] to obtain a Voronoi diagram of the pinna edges, from which an adjacency graph was built for matching. Ansari and Gupta [15] also used the Canny method as the starting point towards extraction of the outer pinna edge for localization of the ear in side face pictures. However, in neither of the two works the effectiveness of the Canny method in the extraction of all pinna edges was made clear.

An analogous approach was adopted by Moreno *et al.* [16]. In order to obtain a profile of an ear picture, Sobel filters were applied both in the horizontal and vertical directions of a grayscale image. Then the most marked intensity discontinuities were derived from each resulting image by standard thresholding, and the sum of the thresholded images gave the profile. This was used either to automatically extract a biometric vector of feature points of the pinna or to compute a morphology vector capturing its shape; these vectors were the input for a perceptron performing classification. Thanks to such heuristic procedure, 90% of feature points were reported to be correctly found.

An alternative to the Canny and Sobel methods was proposed by Choraś [17]. Edge detection in a grayscale image of the pinna was performed through a pixelwise method which examined illumination changes within each 3×3 pixel window. Based on the minimum and maximum values of pixel intensities in such window and on an adaptive threshold value, the center pixel was either labeled as edge or background.

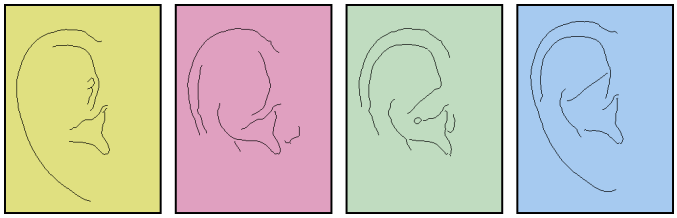

Fig. 1. Canny method ($\sigma = 2$, $t_l = 0.2$, $t_h = 0.5$) applied to four different pictures of the same pinna taken with different light sources.

Feature extraction from the edge map was then fulfilled by tracing concentric circles around the edge map's centroid and computing the intersections between circles and edges. Still, no quantitative results were given for the accuracy of both the edge map computation and the final classification.

Jeges and Máté [18] endorsed a very similar method, where the obtained edge map was used in conjunction with an orientation index image to detect the position of the ear in a video frame sequence and adapt a deformable template model (*active contour method*) to the ear itself. Similarly, in a very recent work Gonzalez *et al.* [19] used adaptation of an active contour model and ovoid fitting to localize the ear in side face pictures and estimate features under the form of distances between the outer and inner pinna edges and the inner edge centroid. No detail on how these edges were extracted is provided.

Jeges' edge extraction algorithm was also a critical component of the reconstruction method by Dellepiane *et al.* [20] which interactively adapted a 3D head model to a specific user starting from a set of pictures of the head and pinna. Following a complementary approach to our anthropometry-based HRTF customization techniques, the resulting model was fed to a simplified boundary element method solver in order to simulate custom HRTFs for that user. Regrettably, few data supported the accuracy of this method for HRTF simulation.

III. A Multi-Flash Camera-Based Approach to Pinna Edge Extraction

Even though edge detection through intensity-based methods seems to be a valid choice in the extraction of pinna edges from 2D images, it is not the sole nor the most efficient option. We initially tried to process pictures with the Canny method, yet it turned out that it fails in low-contrast areas such as the pinna, and especially in cases where shadows are not projected below the considered edge. Fig. 1 shows an example of Canny edge extraction (with standard deviation of the Gaussian filter $\sigma = 2$, lower hysteresis threshold $t_l = 0.2$, and upper hysteresis threshold $t_h = 0.5$) on four pictures of the same pinna taken with different light sources. It can be easily noticed that while in some cases the extraction is acceptable (rightmost image), in all other cases either some important depth edges are lost or some minor depth edges or specular highlights are extracted.

A more robust depth edge extraction can instead be achieved

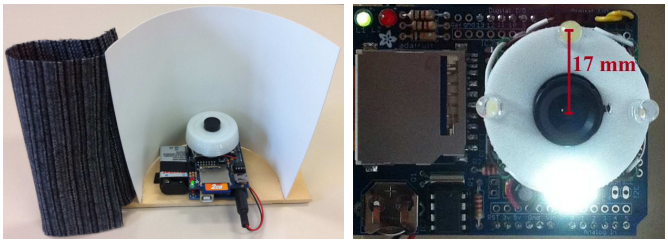

Fig. 2. The multi-flash device: full prototype and electronic parts.

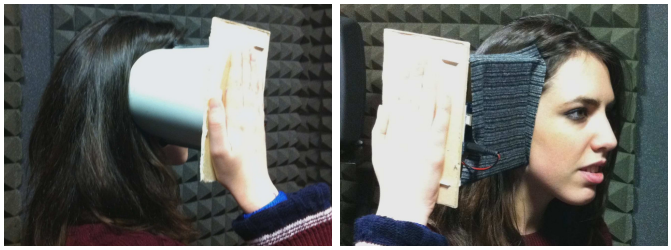

Fig. 3. Subject during acquisition of pinna images.

through a technique known as *multi-flash imaging* [21]. The central concept to multi-flash imaging, which was born as a technical solution to non-photorealistic image rendering, is the exploitation of a camera with N flashes strategically positioned around the lens to cast shadows along depth discontinuities in the scene. For each of the N flashes a picture is taken; the location of the shadows abutting depth discontinuities, appearing only for a strict subset of the N pictures, represents a robust cue to create a depth edge map. Thus, thanks to this simple and computationally efficient method, one can robustly distinguish depth edges from texture edges due to reflectance changes or material discontinuities.

To the best of the authors' knowledge, the method has never been systematically applied to pinna edge detection before. Since the pinna has a uniform texture, the main purpose of multi-flash imaging would reduce to the extraction of the *most marked* depth discontinuities, that usually correspond to the outer helix border, inner helix border, concha wall/antitragus border, and tragus border (see Section III-C for definitions of these anatomical components). We now describe the hardware and software components that implement our multi-flash camera-based pinna edge extractor.

A. The Multi-Flash Camera Prototype

A multi-flash camera prototype was custom built by the authors. The main electronic components building up the device, pictured in Fig. 2, are:

- a battery-powered Arduino UNO microcontroller board;
- an Arduino data logging shield;
- a TTL serial JPEG camera;
- four Super Bright White LEDs;
- a common SD card.

The data logging shield manages data transmission from the camera to the SD card. The four LEDs, which represent the four flashes of our multi-flash camera, are symmetrically positioned around the camera lens along a 17mm-radius circumference and can be turned on independently by the microcontroller. As we will later see, their positions with respect to the pictured scene, i.e. in the up, down, left, and right directions, allow simplification of the post-processing phase. Since the light emitted by each LED has a high directional component that clearly appears in pictures, the application of a punched and reversed paper glass bottom right above the LEDs allows projection of a more diffuse light field.

The electronic components are secured to a rigid board by four long pins and enclosed in a hemi-cylindrical PVC foil, whose shape affords correct orientation as referred to the pinna. The height of the half-cylinder (15 cm) was chosen so as to entirely fit a big-sized pinna (8 cm height) in the pictured frame. Furthermore, the fixed distance between the lens and the pinna allows to maintain consistency among the dimensions of different pinnae. Lastly, because a dark environment is desirable to better project shadows, the open side of the hemi-cylinder is closed by a square of dark cloth with Velcro fastening strips before data acquisition.

Acquisition of the required data is managed as follows. By connecting the battery to the Arduino board, an Arduino sketch performing the following operations is run:

```
while no motion detected do
  wait; {wait for motion detection}
end while
delay 10 s;
for k = 1 to 4 do
  led_k ← turn on;
  take picture;
  led_k ← turn off;
  img_k.jpg ← picture; {save to SD card}
end for
```

When the cap is removed from the lens, motion detection is triggered. During the following 10s pause, the subject presses the open top side of the device around his/her left or right ear trying to avoid hair occlusion and aligning the hemi-cylinder with the pinna (see Fig. 3). Afterwards, four pictures – each synchronized with a different flash light – are acquired. Because of the required storage time of our current prototype this basic procedure takes approximately 30 seconds, during which the subject tries to hold the device as still as possible with respect to the pinna. The four pictures, stored in the SD card as 320×240 pixel JPEG files, are then passed to a PC for processing.

B. Depth Edge Map Computation

After having associated each picture to the position of the corresponding flash light (i_1 = left, i_2 = right, i_3 = up, i_4 = down) depending on whether the left or right pinna has been acquired, the picture set is fed to a MATLAB processing script. The procedure it implements is divided into a pre-processing

phase and an automatic depth map computation phase, whose core is the algorithm described in [21].

The pre-processing phase consists of the following steps:

1) *grayscale conversion*: the four images are converted to grayscale;
2) *intensity normalization*: the four grayscale images are normalized with respect to their mean intensity;
3) *motion correction*: images are first rotated and then translated for the best possible relative alignment according to a standard 2-D correlation function;
4) *low-pass filtering*: each motion-corrected image is median-filtered using a 7-by-7 neighbourhood.

Motion correction is critical in those cases where the subject moved during image acquisition. The best rotation is first calculated by rotating each image i_k, $k = 2, 3, 4$ in 1° increments, cropping the rotated image to fit the original image size, and finding the rotation of i_k that maximizes the correlation coefficient with respect to i_1. The best translation is instead calculated by considering each possible $(320-w_c)\times(240-w_c)$ pixel window of i_k, where w_c is an even positive user-defined parameter that we typically set to $w_c = 20$, and finding the window that maximizes the correlation coefficient with respect to the centrally $(320 - w_c)\times(240 - w_c)$ pixel cropped section of i_1. Finally, low-pass filtering was introduced *a posteriori* to remove the inherent noise introduced by hair in depth maps.

Shadows are now detected by taking a ratio r_k of each image with the pixelwise maximum of all images. Sharp transitions in r_k along the epipolar ray, i.e. the ray connecting the light epipole (defined as the position of the flash light with respect to the taken picture) to the shadowed area, are then marked as depth edges. In our case, since the four flash lights are in the plane parallel to the image plane that contains the camera lens, each light epipole is at infinity and the corresponding epipolar rays are parallel and aligned with the pixel grid. This reduces our problem to the detection of sharp transitions along the horizontal and vertical directions of the ratio images, that can be managed by standard Sobel filters.

More in detail, the depth edge map is calculated as follows:

- for all pixels x, create $i_{max}(x) = \max_k(i_k(x))$, $k = 1, \ldots, 4$;
- for each k, create ratio image $r_k(x) = i_k(x)/i_{max}(x)$;
- calculate e_k, $k = 1, \ldots, 4$ by applying a horizontal Sobel filter to r_1 and r_2 and a vertical Sobel filter to r_3 and r_4;
- keep only the negative transitions in e_1 and e_3 and the positive transitions in e_2 and e_4;
- extract the main depth edges from e_k, $k = 1, \ldots, 4$ through a Canny-like hysteresis thresholding, with upper threshold t_h defined by the user and lower threshold $t_l = 0.4t_h$;
- combine all the edges into a single depth edge map.

The final depth edge map is a $(320 - w_c)\times(240 - w_c)$ binary matrix whose black pixels represent the most prominent depth discontinuities of the pictured scene. As we will later see, the choice of t_h has a non-negligible impact on the extracted edges and on the final results. Fig. 4 reports an example of depth edge map extraction for two subjects with parameters $t_h = 0.35$ and $w_c = 20$.

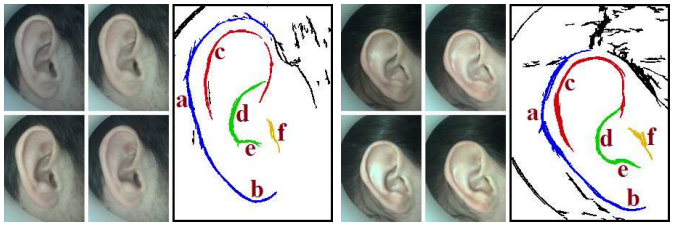

Fig. 4. Depth edge maps of two subjects. The outer helix/lobule (a/b), inner helix (c), concha wall/antitragus (d/e), and tragus (f) borders are highlighted in blue, red, green, and yellow respectively.

C. The Pinna Edge Extraction Algorithm

The depth edge map of the subject's pinna allows extraction of the relevant features that characterize an individual acoustic response. The information contained in the depth edge map that reflects such characterization is included in the Euclidean distance from the points that form the outer helix, inner helix, and concha wall/antitragus borders to a point approximately situated at the meatus entrance, that we conventionally assume to be located in the upper segment of the tragus border (definitions of all borders are given in Fig. 4).

In order to compute distance values, a second MATLAB script that sequentially executes the following steps is run:

1) *map refinement*: the connected components containing less than 100 pixels, i.e. the smallest blobs that usually correspond to spurious hair edges, are deleted;
2) *tragus detection*: the tragus edge is heuristically identified as the connected component lying in the central 200×150 pixel section of the depth edge map whose distance to the bottom left corner (left pinna) or bottom right corner (right pinna) of the map is the least;
3) *meatus point*: the tragus component is subtracted pixelwise to its convex hull and the northwestern/northeastern (left/right pinna) pixel is labeled as the meatus entrance point;
4) *radial sweep*: for each elevation angle $\phi \in [-90^\circ, 90^\circ]$ in 1° steps, all the transitions to a depth edge along the ray originating from the meatus point and heading towards the pinna edges with $-\phi$ inclination are stored as distances (in pixels);
5) *edge tracking*: a partial tracking algorithm [22], originally used in audio signal processing to temporally group sinusoidal partials, is exploited to group distances (i.e. edges) along consecutive frames into spatial tracks, where each frame corresponds to an elevation angle;[1]
6) *pinna edge detection*: the two longest tracks in increasing order of distance value as identified by the edge tracking algorithm, that we call d_1 and d_3, correspond to the concha wall and outer helix border respectively,

[1]The maximum difference between two distances to allow grouping along consecutive frames is set to 5 pixels, while the maximum number of frames before a track being declared dead is set to 10.

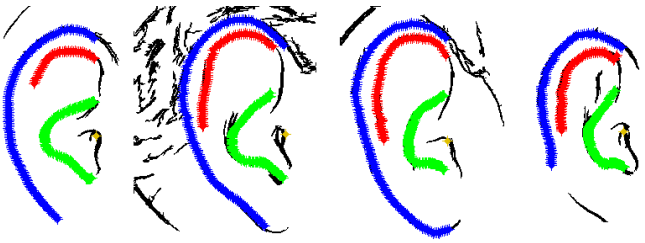

Fig. 5. Edge extraction of the authors' right pinna images.

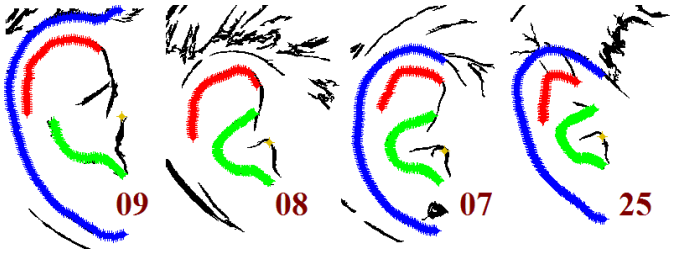

Fig. 6. Edge extraction of right pinna images of four test subjects.

and the longest track falling between these two tracks is called d_2 and corresponds to the inner helix border.

Fig. 5 depicts the results of the edge extraction algorithm as track points superimposed to the refined pinna depth edge maps of the four authors of this paper. This is achieved by simply projecting each point at distance $d_i(\phi)$, $i = 1, 2, 3$ from the yellow meatus point at $-\phi$ inclination.

IV. Results and Discussion

The multi-flash-based approach to pinna edge extraction was tested on a small number of subjects. Right pinna images of 30 volunteers (aged 18 to 60, 12 female and 18 male, caucasian) were acquired with the multi-flash device and then processed. Parameter w_c was set to 20 for all subjects except for 5 of them who required a more substantial motion correction (in these cases, $w_c = 40$). Parameter t_h was set from 0.1 to 0.7 in 0.01 steps in order to look for the range where edge extraction visually outputs the best results. Table I reports this information along with the number of correctly extracted edge tracks for each subject. This means that in the reported t_h range,

- the meatus point is correctly placed in correspondence with the tragus edge and always falls in the same point;
- the three tracks follow the corresponding depth edge in its entirety.

If no t_h value satisfies the latter condition, the reported t_h range refers to two correctly extracted tracks out of three.

One can immediately notice that ranges for t_h are significantly different from subject to subject. The variability among pinna shapes is a first obvious cause of this finding: as an example, subject 17 has a helix that folds into itself almost coming into contact with the antihelix, thus failing to project a consistent shadow. This results into a very shallow depth edge that is not recognized in the reported t_h range. Outside this range, either the number of edges is too high to discriminate the real depth edges from any artifact (low t_h) or some relevant depth edges are lost or broken (high t_h). Another factor that contributes to the determination of the lower t_h bound is the possible connection between the tragus and concha edges, that does not allow correct detection of the meatus point.

TABLE I
Pinna Edge Extraction: Results.

subject	t_h range	# tracks	bad tracks
01	$0.29 - 0.33$	3	
02	$0.43 - 0.47$	3	
03	$0.43 - 0.56$	3	
04	$0.37 - 0.58$	3	
05	$0.23 - 0.34$	3	
06	$0.21 - 0.43$	3	
07	$0.27 - 0.60$	3	
08	$0.24 - 0.40$	2	d_3 missing
09	$0.27 - 0.49$	2	d_1 interrupted
10	$0.23 - 0.31$	3	
11	$0.27 - 0.38$	3	
12	$0.25 - 0.51$	3	
13	$0.25 - 0.32$	3	
14	$0.28 - 0.33$	3	
15	$0.40 - 0.60$	3	
16	$0.29 - 0.40$	3	
17	$0.28 - 0.39$	2	d_2 missing
18	$0.28 - 0.46$	2	d_1 interrupted
19	$0.37 - 0.43$	3	
20	$0.22 - 0.45$	3	
21	$0.24 - 0.50$	3	
22	$0.38 - 0.41$	3	
23	$0.33 - 0.40$	2	d_3 missing
24	$0.19 - 0.38$	3	
25	$0.36 - 0.44$	3	
26	$0.45 - 0.55$	2	d_2 missing
27	$0.20 - 0.57$	3	
28	$0.30 - 0.48$	3	
29	$0.31 - 0.40$	3	
30	$0.27 - 0.32$	3	

Two more examples of how pinna morphology affects the final results are subjects 09 (see Fig. 6) and 18, whose concha wall is not fully extracted. This is due to the shape of the concha itself, resulting in two or more separate and nonintersecting edges (as in the pinna of Fig. 1). Since the grouping conditions of the edge tracking algorithm are not satisfied, no interpolation between these edges is performed and only partial extraction of the concha edge occurs.

Motion correction also plays an important role in the determination of the t_h range. As a matter of fact, linear correction often does not perfectly align the four pinna images. This causes the same edge to be considered twice or thrice in the final depth map in slightly different yet overlapping positions, resulting in thicker depth edges. At the same time, a non-perfect alignment allows extraction of the outer helix border when the back of the ear is surrounded by hair, as shadows on hair are only rarely detected by the multi-flash setup. The second pinna in Fig. 6 shows a case (subject 08) into which a very good alignment is reached yet part of the outer helix border fails to be extracted.

However, if we consider a t_h value included in the reported range for each subject, the meatus point is correctly identified for all subjects, and 84 out of 90 edge tracks are correctly extracted (success rate: 93.3%). Statistically, the t_h value that guarantees a correct extraction of the edge tracks for the highest number of subjects is $t_h = 0.31$. These findings are conditioned by the fixed relation $t_l/t_h = 0.4$, hence further work is needed to check whether a different lower/upper threshold ratio improves the above success rate.

The described edge extraction procedure also seems to be robust in those cases where earrings, glasses or other objects appear in pictures (e.g. subject 07 in Fig. 6). Even small amounts of hair occlusion causing the detection of depth edges due to hair (e.g. subject 25 in Fig. 6) do not corrupt the extracted tracks.

The above results refer to a preliminary study. An extensive analysis on a wide variety of subjects with different pinna sizes, shapes, and albedo values is required to robustly assess the effectiveness of the edge extraction procedure and to study how the t_h parameter can be automatically defined prior to the image post-processing routine. Nevertheless, a more robust motion correction is required before: possible solutions to this issue, whose feasibility still has to be investigated, include

- the exploitation of more complex feature-based image alignment (*image registration*) algorithms;
- *fast shooting* of pictures, in order to reduce the duration of the acquisition routine down to a few seconds and make motion correction become much less critical;
- *single-shot multi-flash* photography [21], [23], a little explored idea according to which four different flash colours can be used to take a single picture of the scene so that the Bayer filter of the camera should be able to decode the separate light wavelenghts and thus derive four different pictures each related to a single flash.

Other improvements need to be introduced to the multi-flash edge extractor, especially at hardware level. For instance, the four flashes can be placed farther from the lens in order to project broader shadows and thus improve the depth edge extraction. A similar result can be achieved by a configuration with more flash lights, e.g. 8. Other working ideas include the improvement of the outer shell of the device and the use of thermogram imagery to robustly detect the meatus location as well as to remove partially occluding hair [9]. Furthermore, at software level, a combination of depth and intensity edge detection techniques will greatly improve extraction of the outer helix border.

The proposed technology was designed so as to be applied to automatic measurements of pinna anthropometry for binaural audio rendering, and represents a low-cost alternative to technologies involving 3D image acquisition (e.g. laser scanners). Nevertheless, ear biometrics represents a natural applicative area for the multi-flash edge extractor, as the feature vectors (distance tracks) it produces share analogies with respect to those used in recent systems, especially [19]. A deeper study of the applicability of this technology to a complete biometric system will disclose its real potential.

REFERENCES

[1] C. I. Cheng and G. H. Wakefield, "Introduction to head-related transfer functions (HRTFs): Representations of HRTFs in time, frequency, and space," *J. Audio Eng. Soc.*, vol. 49, no. 4, pp. 231–249, April 2001.

[2] A. W. Bronkhorst, "Localization of real and virtual sound sources," *J. Acoust. Soc. Am.*, vol. 98, no. 5, pp. 2542–2553, November 1995.

[3] C. P. Brown and R. O. Duda, "A structural model for binaural sound synthesis," *IEEE Trans. Speech Audio Process.*, vol. 6, no. 5, pp. 476–488, September 1998.

[4] E. M. Wenzel, M. Arruda, D. J. Kistler, and F. L. Wightman, "Localization using nonindividualized head-related transfer functions," *J. Acoust. Soc. Am.*, vol. 94, no. 1, pp. 111–123, July 1993.

[5] H. Møller, M. F. Sørensen, C. B. Jensen, and D. Hammershøi, "Binaural technique: Do we need individual recordings?" *J. Audio Eng. Soc.*, vol. 44, no. 6, pp. 451–469, June 1996.

[6] S. Spagnol, M. Geronazzo, and F. Avanzini, "Fitting pinna-related transfer functions to anthropometry for binaural sound rendering," in *Proc. IEEE Int. Work. Multi. Signal Process. (MMSP'10)*, Saint-Malo, France, October 2010, pp. 194–199.

[7] ——, "On the relation between pinna reflection patterns and head-related transfer function features," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 3, pp. 508–520, March 2013.

[8] A. V. Iannarelli, *Ear Identification*, ser. Forensic Identification. Fremont, CA, USA: Paramount Publishing Company, 1989.

[9] M. Burge and W. Burger, "Ear biometrics in computer vision," in *Proc. 15th IEEE Int. Conf. Pattern Recog.*, Barcelona, Spain, September 2000, pp. 2822–2826.

[10] A. Abaza, A. Ross, C. Hebert, M. A. F. Harrison, and M. S. Nixon, "A survey on ear biometrics," *ACM Trans. Embedded Computing Systems*, vol. 9, no. 4, pp. 39:1–39:33, March 2010.

[11] H. Chen and B. Bhanu, "Contour matching for 3D ear recognition," in *Proc. 7th IEEE Work. Appl. Comp. Vision (WACV/MOTION'05)*, Breckenridge, CO, USA, January 2005, pp. 123–128.

[12] C. Hetherington, A. I. Tew, and Y. Tao, "Three-dimensional elliptic Fourier methods for the parameterization of human pinna shape," in *Proc. 2003 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP 2003)*, Hong Kong, China, April 2003, pp. V–612–V–615.

[13] D. J. Hurley, M. S. Nixon, and J. N. Carter, "Force field feature extraction for ear biometrics," *Comput. Vision Image Understand.*, vol. 98, no. 3, pp. 491–512, June 2005.

[14] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, November 1986.

[15] S. Ansari and P. Gupta, "Localization of ear using outer helix curve of the ear," in *Proc. 17th IEEE Int. Conf. Comput. Theory Appl.*, Kolkata, India, March 2007, pp. 688–692.

[16] B. Moreno, A. Sánchez, and J. F. Vélez, "On the use of outer ear images for personal identification in security applications," in *Proc. IEEE 33rd Int. Carnahan Conf. Security Tech.*, Madrid, Spain, October 1999, pp. 469–476.

[17] M. Choraś, "Ear biometrics based on geometrical feature extraction," *Electron. Lett. Comput. Vision Image Anal.*, vol. 5, no. 3, pp. 84–95, August 2005.

[18] E. Jeges and L. Máté, "Model-based human ear localization and feature extraction," *Int. J. Intell. Comput. Med. Sci. Image Process.*, vol. 1, no. 2, pp. 101–112, 2007.

[19] E. González, L. Alvarez, and L. Mazorra, "Normalization and feature extraction on ear images," in *Proc. IEEE 46th Int. Carnahan Conf. Security Tech.*, Boston, MA, USA, October 2012, pp. 97–104.

[20] M. Dellepiane, N. Pietroni, N. Tsingos, M. Asselot, and R. Scopigno, "Reconstructing head models from photographs for individualized 3D-audio processing," *Comp. Graph. Forum*, vol. 27, no. 7, pp. 1719–1727, 2008.

[21] R. Raskar, K.-H. Tan, R. Feris, J. Yu, and M. Turk, "Non-photorealistic camera: Depth edge detection and stylized rendering using multi-flash imaging," *ACM Trans. Graphics (Proc. SIGGRAPH)*, vol. 23, no. 3, pp. 679–688, August 2004.

[22] R. J. McAulay and T. F. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 34, no. 4, pp. 744–754, August 1986.

[23] D. A. Vaquero, R. Raskar, R. S. Feris, and M. Turk, "A projector-camera setup for geometry-invariant frequency demultiplexing," in *Proc. IEEE Conf. Comput. Vision Pattern Recog. (CVPR 2009)*, Miami, FL, USA, June 2009, pp. 2082–2089.