15th IAPR International Conference on Machine Vision Applications (MVA)
Nagoya University, Nagoya, Japan, May 8-12, 2017.

Human ear Structure From Motion

Salah Eddine KABBOUR
CentraleSupelec
Avenue de la Boulaie 35576 Rennes France
salah7ddine@gmail.com

Pierre-Yves RICHARD
CentraleSupelec
Avenue de la Boulaie 35576 Rennes France
Pierre-Yves.Richard@centralesupelec.fr

Abstract

This paper proposes a simple automatic 3D ear reconstruction method using a video selfie while having no knowledge about the scene. First we use the EXIF data stored within the images to estimate the intrinsic matrix. Then, standard structure from motion techniques are applied to obtain a reconstruction for every couple of images which will be merged into one reconstruction that represents the whole scene using an original procedure. Finally, the quasi-dense model is obtained by using the ASIFT and ZNCC descriptors. Our approach is simple to implement and only requires the use of a smart-phone camera. Also, the experimental results showed that this approach is very promising compared to the other methods used to solve this problem.

1 Introduction

Human ear 3D structure from motion (SFM) piqued interest in recent years, its importance in the domain of recognition systems and 3D modeling is beyond questioning. Anika Pflug *et al.* showed [1] that the three dimensional ear models could be the solution to the current challenges in 2D ear recognition, especially problems concerning pose variation and variation in camera position. There is also a new research utilizing 3D ear models for making a 3D sound that gives a better sensation of sound direction and immersion, this new upcoming technology is based on an adaptive filter that takes the human morphology into consideration, hence the use of 3D ear models.

For the last decade, researchers have been working on ear-based identification technology using a variety of approaches. Steven Cadavid *et al.* [2] were one of first group of people who attempted to establish human recognition system based on 3D ear reconstruction . He extracted different video frames, then he applied a modified structure from shading technique on every frame independently to get multiple 3D models. Among these models, he then chose the one which shares the greatest similarity to the rest of 3D models set. The downside on this approach is its vulnerability to luminosity variation even to a small degree.

Heng liu *et al.* tried a different aproach [3], their goal was to come up with an automatic multi-view 3D ear reconstruction method using a device that controls precisely the position and the angle of the camera while allowing them to keep a constant brightness as well. Firstly they used harris corner detector [4] to extract ear feature points and RANSAC to filter the outliers, but the results proved that changes must be made since they only obtained few matching points; so they proposed a semiautomatic way to select the matching points among the photos which allowed them to obtain between 300 and 600 vertices.

Other methods were invented to solve this problem, hui zeng *et.al.* [5] used binocular stereo vision to obtain 3D ear points, his method is based on finding dense correspondence points by applying SIFT followed by applying a match propagation algorithm combined with the knowledge of the epipolar geometry constraints.

In this work we try to obtain a 3D reconstruction of the ear while having zero knowledge about the scene. Unlike the previous works in this field where they use systems that give them control over the brightness or the knowledge of the camera position and/or angle, we propose an approach based on a standard smart-phone camera that gives a dense reconstruction of the ear and which can be used by any individual.

2 Estimating the intrinsic parameters

In our experiments we used a smart-phone camera with a standard CMOS image sensor but these results could be obtained for any other kind of camera. The intention of this work is to automatize the whole procedure to obtain the 3D model, so we're going to use a simple autonomous method.

In order to find the focal length we used the EXIF information provided within the photo:

$$f_x[pixels] = ImgW[pixels] * \frac{F_l[mm]}{CSW[mm]}$$

$ImgW$, F_l and CSW refer to the image width in pixels, Focal length and camera sensor width in millimeters. Also, there is an alternative to this formula that can be used, since most of modern cameras nowadays provide the equivalent to the focal length for a 35 millimeters film, one can substitute $\frac{F_l[mm]}{CSW[mm]}$ by $\frac{F_{35mm}}{36}$.

3 Detecting the ear

Before applying any method of searching for matches, we must restrict the area of the search in order to prevent finding matches at infinity. For this task we used a Viola Jones detector provided by [6] where he uses 5000 positive images and 15000 negative images to train his detector.

4 Feature extraction and matching

There exist a dozen feature extracting methods, SIFT is one of the most known and the most used techniques for this purpose. In our work we used a variant of the SIFT method called ASIFT [7] which proved to be more robust than the original one, and it gives more matching points, which is extremely useful especially in our case since human ears don't usually provide a lot of texture information.

5 3D reconstruction

The main two methods to obtain a sparse 3D reconstruction of a scene from multiple images are the following:
-Using the SFM techniques on two images (three in case you're using the trifocal tensor) then add the rest of the views one by one.

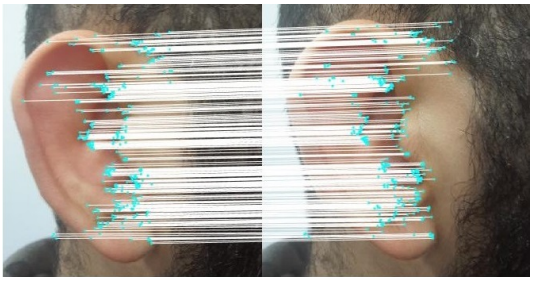


Figure 1: *ASIFT results on two consecutive frames, 332 point correspondences have been found .*

-Using the SFM techniques on pairs of images (triplets in case of trifocal tensor) to get multiple 3D reconstructions, then merge all into one 3D reconstruction.

Each one of these methods has it ups and downs, The first one is very dependent of the initial 3D Reconstruction and the second method is more computationally expensive. Though we select the latter because we prefer robustness over computation speed; in either method, going for the trifocal tensor brings more stability to the system , but we choose to use the fundamental matrix for the sake of simplicity.
First and foremost, we're going to robustly estimate the fundamental matrix for every consecutive couple of images using the RANSAC technique which will allow us to get rid of the outliers eventually.
Then we will use Hartley's method which he described in his famous book [8] chapter 9 where he went over the method to calculate the camera matrices P_i and P_{i+1} using the fundamental matrix and the way to choose one of the four possible solutions for the rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{t}$. He then uses triangulation on these matrices to obtain the 3D points (see chapter 12). it should be noticed in this regard that among the four solutions we should choose the one that results in all the points being in front of the two cameras. But it does not necessarily provide a unique solution. Sometimes one may end up with two or more solutions with the same number of points in front of the cameras. This may happen when the RANSAC fails to find the accurate fundamental matrix or for other reasons. In this case it is preferable to discard one of the two current views and replace it with the next one.

So far we should have the reconstructed 3D points for each point correspondence $x_i \leftrightarrow x_{i+1}$ in two consecutive views:

$$\alpha_i \mathbf{x_i} = \mathbf{P_i X} \qquad \alpha_{i+1}\mathbf{x_{i+1}} = \mathbf{P_{i+1}X}$$

Where α_i and α_{i+1} are non-zero scale factors.

6 Merging 3D reconstructions

By now we should have 3D reconstructed points for every pair of consecutive images. the question that arises is how to merge all of these into one 3D reconstruction.

Modern work on multi-view structure from motion using global methods [9] and [10] attempts to estimate the global rotation and the motion before calculating the position of the 3D points, they also try to match each image multiple times with different images which is computationally expensive. This is considered as an overkill for a small set of images (from 3 to 8 images), so we developed a simple and easy method to implement for this kind of problem. We propose a novel algorithm that estimates the rotation, translation and the 3D points position at the same time.

In this section we are going to describe the original method proposed to merge two sets of camera views, these sets may contain more then just two views each; obviously we need to have at least one camera in common between these two sets.

Let Γ_A and Γ_B be two sets of camera matrices, each set corresponding to views which either have been simultaneously used to reconstruct 3D points, or all result from a previous merging; in either case, each set includes matrices expressed in the same world coordinates:

$$\Gamma_A = \{\mathbf{P_{Ak}}\}_{k \in S_A}$$
$$\Gamma_B = \{\mathbf{P_{Bl}}\}_{l \in S_B}$$

Where $S_A \subset \mathbb{N}$ and $S_B \subset \mathbb{N}$.
Also let Λ_A and Λ_B be two sets of the 3D points associated with Γ_A and Γ_B.

We consider an $i \in \mathbb{N}$ so that $\mathbf{P_{Ai}} \in \Gamma_A$ and $\mathbf{P_{Bi}} \in \Gamma_B$, meaning that $\mathbf{P_{Ai}}$ and $\mathbf{P_{Bi}}$ represent the same camera view in the two sets.

The next step would be transforming all the camera matrices that exist within Γ_B so that they become in the same world coordinates as the ones in the set Γ_A, and do the same for the 3D points in Λ_B. Knowing that the scene was static during the time the photos were taken, we are sure that there exists an invertible transformation matrix $\mathbf{T}$ that satisfies: $\forall \mathbf{X_B} \in \Lambda_B$, $\mathbf{T^{-1}X_B}$ is expressed in Λ_A world coordinates. This transformation is composed of a rotation, a translation and a scale.

We can notice that applying this transformation doesn't change the projection of the 3D point, $\mathbf{x} = \mathbf{P_{Bi}X_B} = (\mathbf{P_{Bi}T})(\mathbf{T^{-1}X_B}) = \mathbf{P_{Ai}X_A}$

We know from before that every camera matrix $\mathbf{P}$ should be written as $\mathbf{P} = [\mathbf{R}|\mathbf{t}]$ where $\mathbf{R}$ is the rotation matrix and $\mathbf{t}$ is the translation vector; all rotation matrices by definition satisfy $\mathbf{R}^{-1} = \mathbf{R}^T$. Now let's go back to our camera view i, knowing that $\mathbf{P_{Ai}} = [\mathbf{R_{Ai}}|\mathbf{t_{Ai}}]$ and $\mathbf{P_{Bi}} = [\mathbf{R_{Bi}}|\mathbf{t_{Bi}}]$ one can easily verify that:

$$\mathbf{P_{Bi}} \left[\begin{array}{ccc:c} & R_{Bi}^T R_{Ai} & & R_{Bi}^T t_{Ai} - s R_{Bi}^T t_{Bi} \\ 0 & 0 & 0 & s \end{array}\right] = \mathbf{P_{Ai}}$$

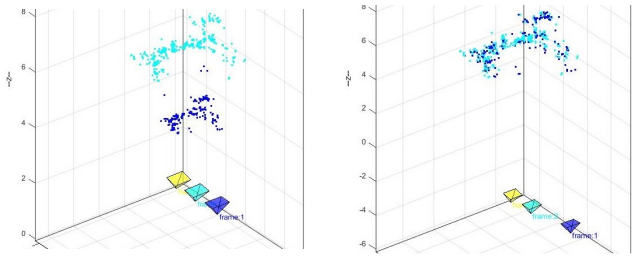
This relation holds true $\forall s \in \mathbb{R}^{+*}$ which represents the global scale. In order to estimate this scale, we propose to search for a common 3D point that has been reconstructed in both Λ_A and Λ_B while visible in the common view i noted $\mathbf{X_c^{Ai}}$ and $\mathbf{X_c^{Bi}}$, then the scale is approximately equal to $s = d(\mathbf{X_c^{Ai}}, \mathbf{C_{Ai}}) / d(\mathbf{X_c^{Bi}}, \mathbf{C_{Bi}})$ where $d(\mathbf{X_c^{Ai}}, \mathbf{C_{Ai}})$ is the distance between the 3D point and the center of the common camera i.

Now that we found our matrix $\mathbf{T}$, we can use it to transform every camera matrix and 3D point that belongs to Λ_B . With a few index changing we should have our merged set ready, but first we have to verify that $\mathbf{T}$ is indeed invertible. We notice that $det(\mathbf{T}) =$

$sdet(\mathbf{R_{Bi}^T R_{Ai}})$, and we know that $det(\mathbf{R_{Bi}^T}) = 1$ and $det(\mathbf{R_{Ai}}) = 1$, thus $det(\mathbf{T}) = s$.

Using this technique allows to reconstruct the 3D sparse model of the whole scene eventually by merging the model obtained from views 1 and 2 with the one obtained from 2 and 3, which will result in a model containing the views from cameras 1,2 and 3, then merge this one with the 3D model coming from views 3 and 4; and so on. Again, we are suggesting a very basic policy to merge the 3D models.

The results of merging two 3D models is rarely satisfying, most of the time merging will tend to produce two separated clusters of 3D points (an example is shown in figure 2). This is why it is indisputably necessary to apply bundle adjustment.



(a) (b)

Figure 2: *The reconstructed scene from frame 1,2 and 3 after merging two reconstructions, in dark blue is the reconstructed 3D points from frame 1 and 2, in cyan from frame 2 and 3.(a) represent the reconstruction right after the merging while (b) represent the results after optimizing using the Levenberg-Marquardt algorithm.*

7 Bundle adjustment

Before going forth, it's in our interest to refine our 3D points coordinate. this step is crucial, and keep merging 3D reconstructions without optimizing them will only make the error grow exponentially.

Firstly, we will define the reprojection error that should be minimized. Let $\mathbf{x_{ij}}$ be the features detected on a certain frame, and $\mathbf{X_i}$ the 3D reconstructed points of this feature with respect to the projection matrix $\mathbf{P_j}$. Let $\hat{\mathbf{x}}_{\mathbf{ij}}$ be the reprojection of the point $\mathbf{i}$ on image $\mathbf{j}$:

$$\hat{\mathbf{x}}_{\mathbf{ij}} = \mathbf{P_j X_i}$$

That has been said, the reprojection error that needs to be minimized is :

$$\sum_{i=0}^{n}\sum_{j=0}^{m} d(\hat{\mathbf{x}}_{\mathbf{ij}}, \mathbf{x_{ij}})^2$$

Where d is the euclidean distance.

The question arises of what are the parameters that we are going to minimize and what method we are going to use, also there is the problem of avoiding local minima. Several approaches have been proposed to solve this problem, B Triggsc *et al.* [11] went in depth on a variety of methods starting by problem parameterisation, error modeling and the different implementation strategies, We decided to implement one of the simplest forms of bundle adjustment: a first order Levenberg-Marquardt algorithm, MaNolis I. [12] wrote a brief Description of this technique.

The parameters that we need to optimize are: $\mathbf{X_i}$, the rotation $\mathbf{R_j}$ and the translation $\mathbf{t_j}$ of every projection camera $\mathbf{P_j}$. The changing of these parameters is going to hopefully produce a better reprojection error, but changing the rotation matrix might be tricky, because we are not sure that the optimized value of $\mathbf{R}$ is going to keep its rotation matrix properties, So we decided to compute Euler angles α_j, β_j and γ_j from the rotation matrix $\mathbf{R_j}$, and optimize these 3 angles, then compute the rotation matrix from the optimized angles, this solution is also much less computationally expensive.

We described in the section above the method used to implement bundle adjustment, so the key solution is to use bundle adjustment after every time we merge two 3D reconstruction, after that we will also remove the points that kept high reprojection error even after applying bundle adjustment, we choose an arbitrary value of 10 pixels as a ceiling to filter all the ill reprojected points. Finally we will rerun bundle adjustment. Also it's worth noting that there exists an open source algorithm developed by Google called Ceres solver which is widely used for bundle adjustment purposes.

8 3D quasi-dense reconstruction

Our aim in this section is to obtain a dense reconstruction from the sparse matches between our pairs of images, for this task, we choose to use the match propagation algorithm, it will allows us to search for a large number of matching points between our images, and since we already know all the information regarding the scene we will just re-triangulate these new matches to obtain the dense 3D reconstructed points.

M. Lhuillier *et al.* [13] proposed algorithm that utilizes the ZNCC descriptor (zero-mean Normalized Cross-Correlation) with cross-consistency check, the propagation is initiated by a set of matches$\{(\mathbf{x}, \mathbf{x}')\}_j$ between the images I and I', these initial matches called *seed points.*

The ZNCC descriptor of the pair $(\mathbf{x}, \mathbf{x}')$ is defined as:

$$\frac{\sum_i \Big(I(\mathbf{x}+i) - \mu_I(\mathbf{x})\Big)\Big(I'(\mathbf{x}'+i) - \mu_{I'}(\mathbf{x}')\Big)}{\sqrt{\sum_i \Big(I(\mathbf{x}+i) - \mu_I(\mathbf{x})\Big)^2 \sum_i \Big(I'(\mathbf{x}'+i) - \mu_{I'}(\mathbf{x}')\Big)^2}}$$

where $\mathbf{x} + i$ is index of the neighbor pixel in a given windows around the center $\mathbf{x}$, and $\mu_I(\mathbf{x})$ is the mean value on that window.

The propagation algorithm follows 3 simple steps:

- search for the best pair $(\mathbf{x}, \mathbf{x}')$ in terms of $ZNCC$ score, and remove it from the list of *seed points*
- search for pairs $(\mathbf{u}, \mathbf{u}')$ in the neighboring window of $(\mathbf{x}, \mathbf{x}')$ that exceeds a $ZNCC$ threshold and also satisfy a certain number of constraints.
- store these pairs in the disparity map and in the list of *seed points*

This process is repeated until the list of seeds is empty. The neighboring window of $(\mathbf{x}, \mathbf{x}')$ is defined as:

$$\mathcal{N}(\mathbf{x}, \mathbf{x}') = \{(\mathbf{u}, \mathbf{u}') | \mathbf{u} \in \mathcal{N}(\mathbf{x}), \mathbf{u}' \in \mathcal{N}(\mathbf{x}')\}$$

where

$$\mathcal{N}(\mathbf{x}) = \{\mathbf{u} | (\mathbf{u} - \mathbf{x}) \in [-N, N]^2\}$$
$$\mathcal{N}(\mathbf{x}') = \{\mathbf{u}' | (\mathbf{u}' - \mathbf{x}') \in [-N, N]^2\}$$

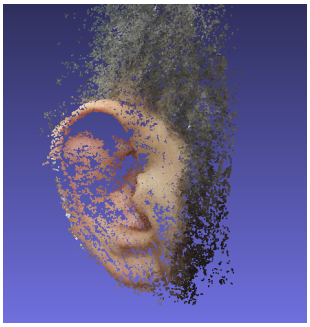


Figure 3: *An example of 3D ear reconstruction result using 5 images, number of vertices 93114*

Also there is a number of constraints that will be applied to prevent the algorithm from finding false matches. The first one, called the disparity constraint, which will be used to insure that the transformation vector from the seed point to the new candidate has the same direction in both images, so a seed point that satisfies the disparity constraint is defined as:

$$\{(\mathbf{u},\mathbf{u}')|\mathbf{u}\in\mathcal{N}(\mathbf{x}),\mathbf{u}'\in\mathcal{N}(\mathbf{x}'),\\ ||(\mathbf{u}'-\mathbf{x}')-(\mathbf{u}-\mathbf{x})||_\infty\leq\epsilon\}$$

furthermore, since we already know the fundamental matrix, we can check if a new candidate $(\mathbf{u},\mathbf{u}')$ satisfies the epipolar constraint, which means that $\mathbf{u}$ belongs to the epipolar line of $\mathbf{u}'$ and vice versa.
There are also other constraints that can be implied to improve the dense matching results or prevent the propagation into too uniform areas.

Overall, the reason why we choose the ZNCC descriptor for our propagation method is because it is invariant to linear radiometric changes and more tolerant to noise, but its main problem is that it requires little to no change in camera angles while taking the two images. Juha kannala *et al.* [14] addresses this issue in details, he comes up with an extension to the match propagation algorithm for wide baseline matching.

9 Experimental results

The experiments were run on several sets of photos taken by a smart-phone camera with a resolution of 3264x183. We found out that 5 photos are usually enough to produce the desired results. The whole process takes from 10 to 15 min in average to produce the quasi-dense reconstruction (there is plenty room for optimisation). Our method can produce results of reconstructions that vary from 30 000 to 100 000 vertices, which is much denser compared to the methods proposed by [3] and [5]. Table 1 shows the comparison of different methods.

Table 1: Comparison between different methods

	Methods				
	Range Scan	Structure Light	Multi-view	Hui's method	Our method
Vertices	7000 9000	2000 4000	300 600	2000 3000	30000 90000

The downside of this method is that sometimes it fails to merge effectively the different reconstructions, which becomes a problem for bundle adjustment, due to its huge dependency on a good initialization, and this is often a result of a the lack of correspondences between images.

10 Conclusion

We have presented an automatic method for 3D ear reconstruction based on a video taken by a smart phone camera. Compared to other methods, ours is much simpler to implement and has better results in terms of number of vertices. But still there is place for improvement, especially in choosing the right photos. Also one can utilize the cell phone gyroscope and accelerometer to estimate the camera pose and its rotation.

References

[1] A. Pflug and C. Busch, "Ear biometrics: a survey of detection, feature extraction and recognition methods," *IET biometrics*, vol. 1, no. 2, pp. 114–129, 2012.

[2] S. Cadavid and M. Abdel-Mottaleb, "3-D ear modeling and recognition from video sequences using shape from shading," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 4, pp. 709–718, 2008.

[3] H. Liu and J. Yan, "Multi-view Ear Shape Feature Extraction and Reconstruction." IEEE, Dec. 2007, pp. 652–658.

[4] C. Harris and M. Stephens, "A combined corner and edge detector." in *Alvey vision conference*, vol. 15. Citeseer, 1988, p. 50.

[5] H. Zeng, Z.-C. Mu, K. Wang, and C. Sun, "Automatic 3d ear reconstruction based on binocular stereo vision," in *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on.* IEEE, 2009, pp. 5205–5208.

[6] M. Castrillón Santana, J. L. Navarro, and D. H. Sosa, "An study on ear detection and its applications to face detection," in *Conferencia de la Asociacin Espaola para la Inteligencia Artificial (CAEPIA)*, La Laguna, Spain, November 2011.

[7] J.-M. Morel and G. Yu, "Asift: A new framework for fully affine invariant image comparison," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 438–469, 2009.

[8] R. Hartley and A. Zisserman, "Multiple view geometry in computer vision," pp. 257–260, 2003.

[9] P. Moulon, P. Monasse, and R. Marlet, "Global fusion of relative motions for robust, accurate and scalable structure from motion," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 3248–3255.

[10] M. Arie-Nachimson, S. Z. Kovalsky, I. Kemelmacher-Shlizerman, A. Singer, and R. Basri, "Global motion estimation from point matches," in *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on.* IEEE, 2012, pp. 81–88.

[11] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustmenta modern synthesis," in *International workshop on vision algorithms*, ser. r217. Springer, 1999, pp. 298–372.

[12] M. I. Lourakis, "A brief description of the Levenberg-Marquardt algorithm implemented by levmar," *Foundation of Research and Technology*, vol. 4, pp. 1–6, 2005.

[13] M. Lhuillier and L. Quan, "Match propagation for image-based modeling and rendering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, pp. 1140–1146, 2002.

[14] J. Kannala and S. S. Brandt, "Quasi-dense wide baseline matching using match propagation," in *2007 IEEE Conference on Computer Vision and Pattern Recognition.* IEEE, 2007, pp. 1–8.