Inexpensive Construction of a 3D Face Model from Stereo Images

M. Shahriar Hossain, Monika Akbar and J. Denbigh Starkey
Department of Computer Science, Montana State University, Bozeman, MT 59717, USA.
e-mail: {*mshossain, monika, starkey*}*@cs.montana.edu*

***Abstract*— Construction of a three dimensional face model from stereo images is a challenging task. Most of the currently available systems for reconstruction of 3D models require special hardware for calibration. In this paper, we illustrate a mechanism to construct a three dimensional face model from two stereo images. The developed mechanism does not require any special devices to calibrate the stereo images. We used a hand-held inexpensive digital camera to take the stereo images of a face. We did not use any camera-stand to fix and measure the camera system geometry. The stereo images were taken holding the camera in hand and moving it to two slightly different viewpoints. We constructed a depth map from these two stereo images and utilized this depth map to reconstruct the three dimensional face model. The 3D face model reconstruction process described in this paper uses some existing theories and combines them to develop a new system to generate the depth map. The system requires minimal user interaction for the reconstruction.**

***Keywords*— Image reconstruction, Depth map, Stereo images.**

I. INTRODUCTION

Extraction of three dimensional structures from two dimensional images is an important research area for scientists working in the field of computer vision technology. There has been extensive work in this field [1–12] and while most of the existing techniques work well with geometric objects, they require training dataset images, take advantage of expensive external hardware for calibration, or use many expensive cameras for three dimensional visualization effects. In this paper, we exploit different ideas to extract a depth map of a face from stereo images with the aim of reconstructing a 3D face model. The approach discussed in this paper involves only two handheld stereo images and an inexpensive digital camera. Our aims are to employ minimum user interaction for the reconstruction process while still creating high quality 3D reconstructions.

The generation of 3D face models from stereo images requires two cameras capturing an image of a scene at the same time from two slightly separated viewpoints. The images are rectified such that the image rows are aligned between the two images. Each pixel is then matched from one image to the corresponding pixel in the alternate stereo image. Generally matching a single pixel proves inadequate and a window of surrounding pixels is used. A measure of degree of correspondence between the image windows is used to select the best suited region from the search space. Some popular techniques are Sum of Absolute Differences (SAD), Sum of Squared Differences (SSD) and Normalized Cross Correlation (NCC) [13]. In our work, we have used a single camera to take photographs from two slightly different handheld viewpoints. We use window based pixel by pixel matching and simple mechanisms to omit expensive devices for calibration using the epipolar concept.

Calibration is the process of determining the camera system's external geometry (e.g., the relative positions and orientations of each camera) and internal geometry (e.g., focal lengths, optical centers, and lens distortions). Accurate estimates of these geometries are necessary in order to relate image information to an external world coordinate system. Calibrating stereo cameras is usually dealt with by calibrating each camera independently and then applying geometric transformations of the external parameters to find out the geometry of the stereo setting. We developed our depth map construction approach, keeping the assumption as a constraint of the work that only two face stereo images are available, and there is no information about relative positions of the camera and face objects, angles, focal length and optical centers.

We used the concept of epipolar geometry (see section III of this paper) to find the correspondence between two stereo images. Our approach involves a divide and conquer strategy to retrieve the correspondence between two images and measure the disparity between them. This step involves minimal user interaction for the whole mechanism. The rest of the system is automatic.

The organization of the paper is as follows: related literature is described in section II, a brief overview of our system is in section III, section IV contains the implementation details, and we give conclusions in section V.

II. LITERATURE REVIEW

Computational stereo for the extraction of three dimensional scene structures has been an intense area of research for decades [13]. Systems have been developed over the last decade to fine-tune the three dimensional scene. Besides, there are mechanisms to retrieve the depth map from a single image. Hassner and Basri [1] propose such a novel solution to the problem of depth map reconstruction from a single image, but the mechanism addresses an example based synthesis approach. Their method uses a database of objects from a single class (e.g., hands, human figures) containing example patches of feasible mappings from the appearance to the depth of each object. Given an image of an object, the system combines the known depths of patches from similar objects to produce a plausible depth estimate. Although the approach performs well on structured rigid objects like faces, it requires a training data set of predefined depth maps. Hoiem *et al.* [2] propose another solution for

creating a 3D model from a single photograph. The work concentrates on three dimensional structures of the outdoor environment, rather than concentrating on fine rigid objects. It presents a method for creating virtual walkthroughs that is completely automatic and requires only a single photograph as input. The approach is similar to the creation of a pop-up illustration in a children's book: the image is laid on the ground plane and then the regions that are deemed to be vertical are automatically "popped up" onto vertical planes. Just like the paper pop-ups, the resulting 3D model is quite basic, missing many details. The reconstruction process is restricted to outdoor scenes, and the mechanism requires a training data set to label objects in the image. Horry *et al.*, Kang *et al.* and Pollefeys *et al.* [3–5] describe other similar works.

Debevec *et al.* [6] propose a process that requires a lot of user interaction, and the system depends more on model based geometry, rather than trusting the regular image. Cipolla *et al.* [7] describe a mechanism for retrieving three dimensional architecture from uncalibrated images that does not need *a priori* information about the cameras being used, but it requires user selection of a set of image-edges that are parallel or perpendicular in the world. Ziegler *et al.* [8] describe another mechanism where the user has to define a set of polygonal regions with corresponding labels in each image using familiar 2D photo-editing tools. Their reconstruction algorithm computes the 3D model with maximum volume that is consistent with the set of regions in the input images. Their system works well with geometric objects, but it is not suitable for reconstructing a 3D face model. Liebowitz *et al.* [9] present methods for creating 3D graphical models of scenes from a limited number of images, where no scene co-ordinate measurements are available. The methods employ constraints available from geometric relationships that are common in architectural scenes – such as, parallelism and orthogonality – together with constraints available from the camera. As a result the method works well for outdoor scenes, like houses, buildings, buses, cars, etc. It is not suitable for the reconstruction of three dimensional face models.

Taeone *et al.* [10] propose a method for locating the 3D position of a soccer ball from a monocular image sequence of soccer games which is highly domain dependent, and is not suitable for human face model reconstruction. With similar aims, Criminisi *et al.* [11] describe how three dimensional affine measurements may be computed from a single perspective view of a scene given only minimal geometric information determined from the image. This minimal information is typically the vanishing line of a reference plane, and a vanishing point for a direction not parallel to the plane. The work shows that affine scene structure may be determined from the image, without knowledge of the camera's internal calibration, or the explicit relation between camera and world. The system is not suitable for face model reconstruction, because it determines perspective view from vanishing lines, which is not possible for a face image.

Li-An and Huang [12] propose an approach to the automatic construction of 3D human face models using a generic face model and several 2D face images. A template matching based algorithm is developed to automatically extract all necessary facial features from the front and side profile face images. Then the generic face model is fitted to these feature points by geometric transforms. Finally, texture mapping is performed to achieve realistic results. The authors show that their system generates good quality 3D face models, but the system is dependent on the generic face model used as background knowledge. As a result, the quality of the generated face model depends on the architecture of the generic face model that is utilized for 3D face model reconstruction.

There are other papers ([13–14]) that portray surveys on computational stereo and face recognition techniques. In our paper, we describe a simple method to reconstruct the face model from stereo images without using any background knowledge, expensive equipment, or calibration information about the camera. Our system requires minimal user interaction for reconstruction.

III. System Overview

In this section, we briefly describe our technique for 3D face model reconstruction. We will fill in the details in section IV. The technique is illustrated in Fig. 1. The only user interaction is at the beginning of the process. The user marks four points on a rectangle placed behind the face model in each stereo image. The user eliminates unnecessary parts from the images and creates two new stereo images that contain only the face. Images with marked points are used in the next step for the retrieval of epipolar lines. Images containing only the face of the model (after the elimination of the unnecessary parts) are used for the Optimized Block Matching step of Fig. 1.

Fig. 2 illustrates the concept of epipolar lines in computational stereo analysis. P indicates the point object, where C_L and C_R indicate the optical centers of the left and right camera. P is projected at x and x' respectively in the left and right image. The plane PC_LC_R is called the epipolar plane, and xe and $x'e'$, which are the intersections of the epipolar plane with the two image planes, are called epipolar lines. xe and $x'e'$ correspond to each other in the left and right image. The goal of the *Epipolar Line Detection* step of Fig. 1 is to predict these epipolar lines,

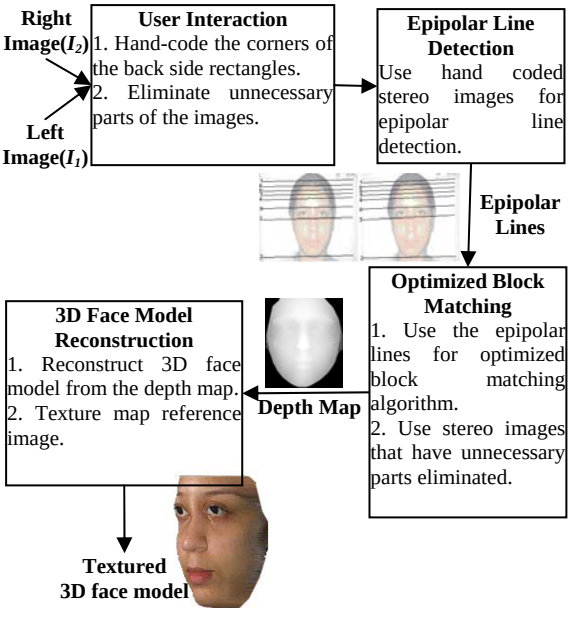


Fig. 1. The 3D face model reconstruction process.

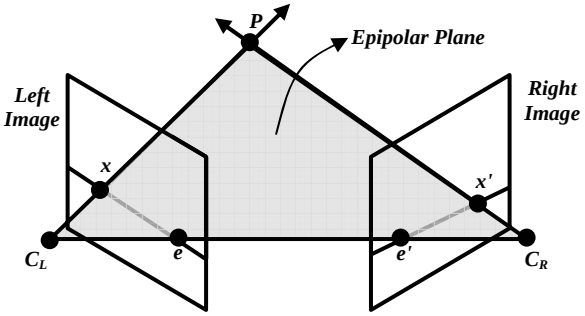


Fig. 2. Epipolar plane and epipolar lines.

so that they can be used in the *Optimized Block Matching* step, which generates the optimized depth map that is used for 3D face model reconstruction. The details of these steps are described in the following section.

IV. Implementation Details

In this section, we describe the detailed mechanism for the 3D face model reconstruction that was shown in Fig. 1. The section is divided into four subsections describing the steps of the reconstruction process – (A) depth map calculation, (B). epipolar line detection, (C) optimized block matching strategy, and (D) reconstruction of the 3D model and texture mapping.

A. Depth Map Calculation

There are two types of correspondence methods for depth map calculations [13], the *local correspondence method*, and the *global correspondence method*. We conducted experiments with the depth map construction strategy for this work. We used local correspondence methods, because local correspondence methods are faster than those of global correspondence methods. Our approach for depth map calculation is called the *block-matching approach* [13]. We can define the problem as follows: we are given two images, and, from the information contained in these images, we must compute disparities. The correspondence problem consists of determining the locations in each camera image that are the projection of the same physical point in space. All correspondence methods attempt to match pixels in one image with their corresponding pixels in the other image. The block matching method seeks to estimate the disparity at a point in one image by comparing a small region about that point (the template of I_1 in Fig. 3) with a series of small regions extracted from the other image (the search region of I_2 in Fig. 3). We used two metrics for block matching: *intensity difference* and *rank metrics*. We found that intensity difference is better in our case. We used the

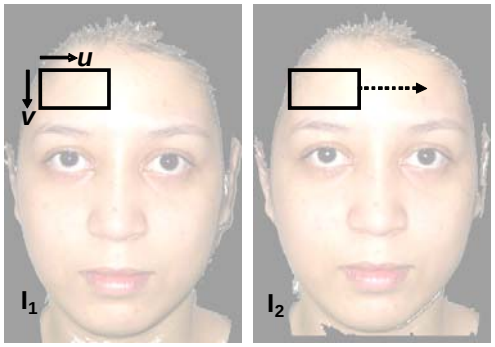


Fig. 3. Block matching approach for depth map calculation.

Fig. 4. Effect of different template size for the block matching scheme.

two most popular statistical formulas for this purpose: (1) Sum of Squared Differences (SSD), and (2) Sum of Absolute Differences (SAD). The equations are [15]:

$$\sum_{u,v} (I_1(u,v) - I_2(u+d,v))^2 \quad (1)$$

$$\sum_{u,v} |I_1(u,v) - I_2(u+d,v)| \quad (2)$$

We examined depth maps using these two formulas and found that SSD is slightly better than SAD, and so for further experiments we used only SSD.

In practice the disparity measurement formula is not the only issue for generating a good depth map. The template size is also an important parameter for depth map calculation. Fig. 4 shows a comparison of generated depth maps between four different template sizes for the block matching algorithm. Naturally, the larger the template size the better the accuracy of the depth map. Unfortunately, these depth maps are not suitable for high quality 3D face model reconstructions. This is because for the depth maps of Fig. 4 we did not use any kind of calibration; our algorithm just scanned the corresponding pixels of the images instead of scanning through the corresponding epipolar lines. Epipolar lines between two images are the actual physical correspondence between two images [13]. The following subsection describes how we measured the epipolar lines without using any kind of external hardware.

B. Epipolar Line Detection

We used a very simple technique for the detection of epipolar lines. We placed a rectangular board behind the model. The board itself contained a black rectangle in such a way that the face height fits in the rectangle. But the black lines of the rectangle did not have strong black intensity, i.e., RGB(0,0,0), in the digital form of the images. So we hand-coded any extreme color on the four corners of the rectangles of the input images (Fig. 5).

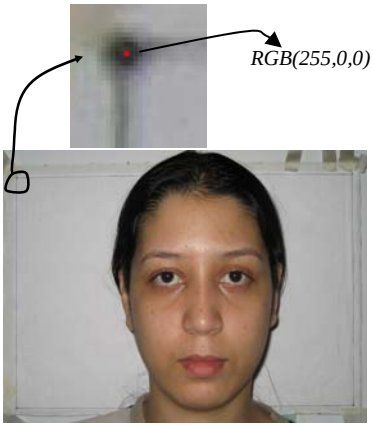


Fig. 5. Corners are hand-coded by red (RGB(255,0,0)).

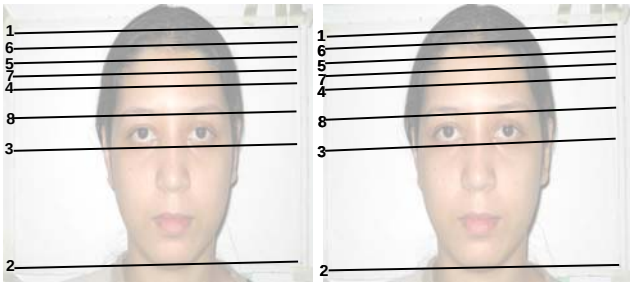


Fig. 6. Divide and conquer strategy to discover the epipolar lines.

(a) **(b)**

Fig. 7. Input images and constructed depth map.

In our case, for both the stereo images, we hand-coded the four corners of the rectangle with extreme red (RGB(255,0,0)) because any real image generally does not contain any kind of extreme color. The algorithm for the detection of epipolar lines between two stereo images first looks for extreme red points in the images and uses them to generate two epipolar lines per image. These two lines are marked 1 and 2 in the stereo images of Fig. 6. The algorithm then detects the other epipolar lines using a recursive divide and conquer strategy. Every time the algorithm detects a line that is at the middle position between two input lines. The corresponding epipolar lines for the other image are also detected in the same way simultaneously. Fig. 6 shows some of the epipolar lines with the detection sequence. The algorithm repeats its recursion until the difference between any corresponding endpoints of the two input lines in the y-direction becomes 1 pixel height or corresponding endpoints overlap. The same number of epipolar lines are selected from two images. These epipolar lines are used as scan-lines for the block matching algorithm discussed in the previous subsection. Before using block matching, we eliminate unnecessary parts (parts other than the face) from both the images. Such input images and the constructed depth map found after using the calibration are given in Fig. 7(a) and (b).

Two images from the reconstructed three dimensional model using the discovered depth map are given in Fig. 8. The constructed three dimensional face model is not satisfactory, especially at the nose, and also since the depth map has lots of noise, the three dimensional model does not look perfect. So, we apply an optimization for the block matching algorithm. The optimization is described in the following subsection.

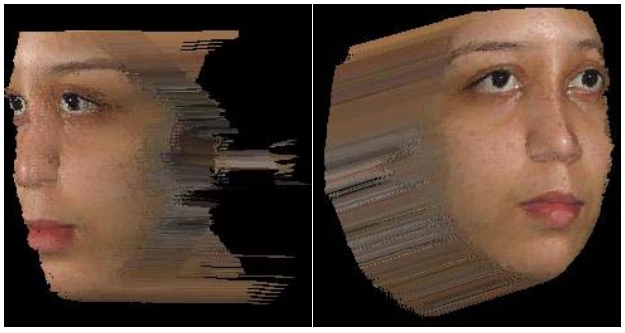


Fig. 8. Snapshot of the constructed three dimensional model (without optimization).

C. Optimized Block Matching Strategy

Our optimization approach has some similarity with the approach used by Hassner and Basri [1], although they generate depth maps from an existing example database. In contrary, we apply an optimization technique for stereo images using the assumption that the left side and right side of a face are relatively symmetrical.

Our observation is that some of the patches/templates from the left side of the face can match with the right side of the other image generating a noisy depth map. Let *h* and *w* be the height and width of the stereo images (both

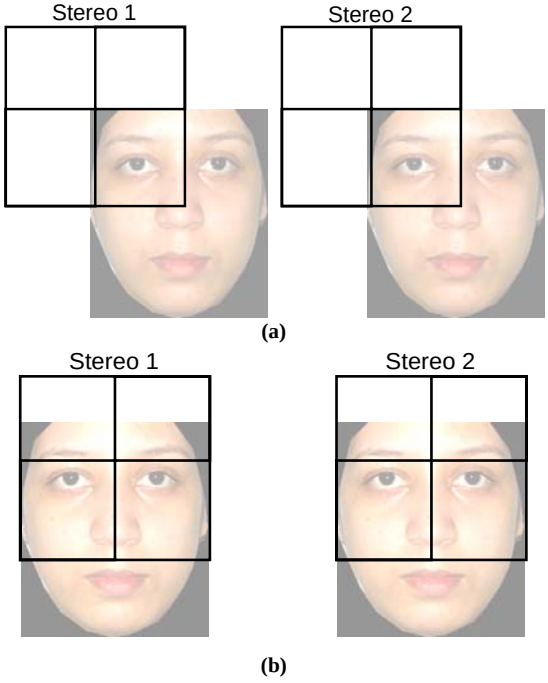
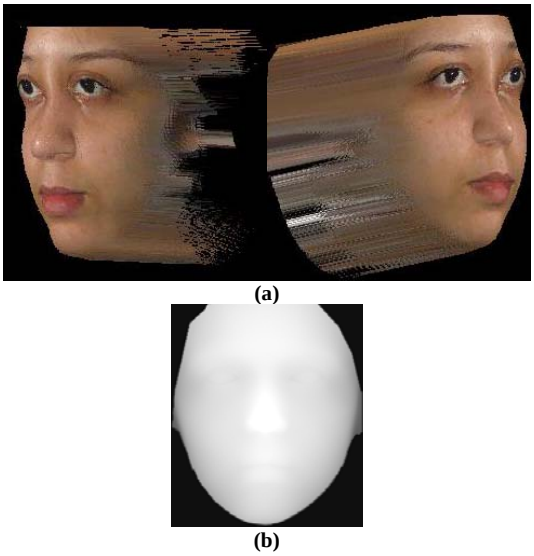


Fig. 9. The optimization process.



(a) **(b)**

Fig. 10. Snapshot of the constructed three dimensional model with optimization. (b) the corresponding depth map.

of our input stereo images have the same size). We add the values (*w/2, h/2*) to every point of two images. While comparing against the second stereo image, the template does not cross the boundary of the image-frame that existed before this addition. Boundary image-frames are shown in Fig. 9 with each image for better illustration. This imposes some symmetry constraint on the depth map construction. As, for example, the left side of the first image is compared with only the left side of the second image (Fig. 9(a)). A template with center positioned at the central vertical line gets the highest priority and conducts a full comparison with the biggest search space, i.e., the whole width of the image (Fig. 9(b)). The depth map found after this optimization is given in Fig. 10(b). Two snapshots after the reconstruction of the 3D face using this depth map are shown in Fig. 10(a), which shows that the 3D reconstruction after the optimization we used is far better than the scheme we used without the optimization given in Fig. 8.

D. Reconstruction of the 3D Model and Texture Mapping

We use a simple method for the reconstruction of the 3D model from the depth map. For a 3D point *(x, y, z)* we take *x* and *y* from the first stereo image and *z* from the gray level intensity at *(x, y)*. Hence the gray level intensity of the depth map becomes the *z*-value for the reconstruction process. We have used texture in a different way from traditional texture mapping techniques. As we are not working with geometric objects and not constructing wireframes, we used our own technique for texture mapping. Consider a pixel of the first stereo to be *P(i,j)*. Then, z_1 is the corresponding depth of *P* obtained from the depth map. Let us denote it as: $z_1=depth(i, j)$. Now, the corresponding 3D point is $P_1(i, j, z_1)$. We select two other points in the following way:

(1) Select $P_2(i–k, j–k, z_2)$ where $z_2=depth(i–k, j–k)$

(2) Select $P_3(i–k,j+k,z3)$ where $z_3=depth(i–k, j+k)$.

We construct a triangle using these three points P_1, P_2 and P_3 . The fill color of the triangle is the color of the pixel of the first stereo image at *P(i, j)*. In this approach, *k* is a user defined constant. In our case, *k=2*. For a better 3D illusion it is better to keep the value of *k* small. Both Fig. 8 and Fig. 10(a) display snapshots after the texture map. Fig. 11 shows a snapshot of the 3D reconstruction with optimization but without the triangles and texture map.

Fig. 11. Snapshot of the constructed three dimensional model without any triangle and texture.

V. CONCLUSION

In this work, we construct a 3D face model from two stereo images. Our technique does not require any training dataset, expensive calibration equipment, or any kind of camera geometry information. The system requires minimal user interaction. Moreover, snapshots of the stereo images are taken by moving the hand-held camera to slightly different viewpoints. As a result, this system could be used by anyone with a cheap digital camera to build 3D face models. The limitation of the technique is that it is not suitable for large objects (e.g., buildings, cars, aircraft, etc.) because we construct the depth map from the RGB color intensity of the stereo images. Consequently, our depth maps have only 256 units of depth which is sufficient for the 3D model reconstruction of a face or similar objects. Although the reconstructed 3D face produces good quality visualization, the constructed depth map still possesses some noise generating a slightly disfigured 3D face. We only concentrated on the depth map of the face and ignored left, right, back and top sides of the model. As a result, the 3D reconstruction is done only for the face, not for the complete head. The task of discovery of an automatic system to construct a complete head model remains as a topic for future work.

REFERENCES

[1] Hassner T., and Basri R., "Example Based 3D Reconstruction from Single 2D Images", *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 15–22, June 2006.

[2] Hoiem D., Efros A., and Hebert M., "Automatic photo pop-up", *International Conference on Computer Graphics and Interactive Techniques, ACM SIGGRAPH 2005*, Los Angeles, California, pp. 577–584, 2005.

[3] Horry Y., Anjyo K.-I., and Arai K., "Tour into the Picture: Using a Spidery Mesh Interface to Make Animation from a Single Image", *24th annual conference on Computer graphics and interactive techniques, ACM SIGGRAPH'97*, pp. 225–232, 1997.

[4] Kang H., Pyo S., Anjyo K., and Shin S., "Tour into the Picture Using a Vanishing Line and Its Extension to Panoramic Images", *Eurographics*, pp. 132–141, 2001.

[5] Pollefeys M., Gool L. V., Vergauwen M., Verbiest F., Cornelis K., Tops J., and Koch R., "Visual Modeling with a Hand-Held Camera", *International Journal of Computer Vision*, vol. 59, no. 3, pp. 207–232, 2004.

[6] Debevec P. E., Taylor C. J., and Malik, J., "Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach", *International Conference on Computer Graphics and Interactive Techniques, ACM SIGGRAPH*, pp. 11–20, 1996.

[7] Cipolla R., Robertson D., and Boyer E., "Photobuilder–3d Models of Architectural Scenes From Uncalibrated Images", *IEEE International Conference On Multimedia Computing And Systems*, vol. 1, pp. 25–31, 1999.

[8] Ziegler R., Matusik W., Pfister H., and Mcmillan, L., "3d Reconstruction Using Labeled Image Regions", *2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing, ACM International Conference Proceeding Series*, vol. 43, pp. 248–259, 2003.

[9] Liebowitz D., Criminisi A., and Zisserman A., "Creating Architectural Models from Images", *Eurographics*, vol. 18, pp. 39–50, 1999.

[10] Taeone K., Yongduek S., and Ki-Sang H., "Physics-based 3D Position Analysis of a Soccer Ball from Monocularimage Sequences", *6th International Conference on Computer Vision*, Bombay, India, pp. 721–726, 1998.

[11] Criminisi A., Reid I., and Zisserman A., "Single View Metrology", *International Journal of Computer Vision*, vol. 40, no. 2, pp. 123–148, 2000.

[12] Li-An T., and Huang T. S., “Automatic Construction of 3D Human Face Models Based on 2D Images”, *International Conference on Image Processing*, vol. 3, pp. 467–470, September 1996.

[13] Brown M. Z., Burschka D., and Hager G. D., “Advances in computational stereo”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 993–1008, August 2003.

[14] Chellappa R., Wilson C. L., and Sirohey S., “Human and machine recognition of faces: a survey”, *Proceedings of the IEEE*, vol. 83, no. 5, pp. 705–741, May 1995.

[15] Aschwanden P., and Guggenbuhl W., “Experimental Results from a Comparative Study on Correlation-Type Registration Algorithms,” *Robust Computer Vision, Forstner and Ruwiedel, eds.*, pp. 268–289, 1993.