Delaunay Triangulation based 3D Human Face Modeling from Uncalibrated Images

Reza Hassanpour

Department of Computer Engineering
Çankaya University
Ankara Turkey
reza@cankaya.edu.tr

Volkan Atalay

Department of Computer Engineering
Middle East Technical University
Ankara Turkey
volkan@ceng.metu.edu.tr

Abstract

In this paper, we describe an algorithm for generating three dimensional models of human faces from uncalibrated images. Input images are taken by a camera generally with a small rotation around a single axis which may cause degenerate solutions during auto-calibration. We describe a solution to this problem by a priori assumptions on the camera. To generate a specific person's head, a generic human head model is deformed according to the 3D coordinates of points obtained by reconstructing the scene using images calibrated with our algorithm. The deformation process is based on a physical based massless spring model and it requires local re-triangulation in the areas with high curvatures. This is achieved by locally applying Delaunay triangulation method. However, there may occur degeneracies in Delaunay triangulation such as encroaching of edges. We describe an algorithm for removing the degeneracies during triangulation by modifying the definition of the Delaunay cavity. This algorithm has also the effect of preserving the curvature in the face area. We have compared the models generated with our algorithm with the models obtained using cyberscanners. The RMS geometric error in these comparisons are less than 1.8×10^{-2}.

1 Introduction

3D head modeling is a special case of the general problem of 3D object representation. Appropriate combinations of geometric descriptors, surface texture and animation capabilities permit modeling of human heads. Graphical modeling of faces has obvious applications in the entertainment industry for character animation, in simulations involving people, man-machine interaction, and security as well. 3D head modeling involves three parts: geometrical, surface and deformation modeling. One of the geometrical modeling methods is the use of a group of discrete 3D points sampled from the outer surface of the head and face. 3D sample points may be grouped into polygons and the head model is then given by a mesh of these polygons. The main advantage of using a mesh is its simplicity of implementation and availability of hardware facilities for effective rendering. The use of a mesh also provides linear interpolated values for unavailable data points. 3D sample points may be found from direct measurements on the head and face such as by range scan data, or obtained from the images taken by digital cameras [20, 13, 9]. In range scanned data, a large number of points makes rendering process and animation slow while the number of sample points obtained from 2D images is generally low. A generic mesh with a limited number of vertices can solve the problems in both cases. However, the correlation between the facial features and the relationships in the geometric properties of the vertices makes the morphing of a generic mesh into a specific one a difficult task. We may group face modeling algorithms using 3D sample points as generic mesh deforming algorithms, interpolating algorithms, and mesh creating algorithms. In the generic mesh deforming algorithms, the locations of vertices in the generic mesh are modified by moving them towards the sampled points and the remaining vertices are adjusted using either an interpolation algorithm or simulating deformations by external forces [6, 7, 8]. The interpolating algorithms argue that a specific head is a weighted average of a group of base head models [19, 17]. Only weight coefficients are unknown in these algorithms. The mesh creating algorithms are based on the argument that a mesh can always be created whenever a set of 3D vertices is available [2, 15]. Generally these groups of algorithms use generic surface modeling methods. The main disadvantage of these methods is that when ap-

plied to human face modeling , there is uncertainty in locating main facial feature vertices.

The main aim of this study is to generate 3D face models from uncalibrated images. Our method is a generic mesh deforming algorithm based on the use of 3D points sampled from the outer surface of the head and face. In the first part of our method, the camera auto-calibration is accomplished. We start with matching points from different views. In the matching step, a set of feature points provided by the user is used to reduce erroneous correspondence. In the second part, we employ the physical parameters of the camera obtained in the calibration part and we reconstruct a 3D cloud of random points. These points are then used in deforming a generic mesh. The final step is adding texture values to the model. We use an approach similar to that of Pighin *et al.* [11] to generate a view independent texture map. Some usual restrictions on the image acquisition process and facial feature extraction are imposed. All images are taken from the same elevation as the face and no tilt to left, right, forward or backward is present in the face position. Face should have a neutral expression and no movement should exist from one image to the other. An important aspect of our study is that the calibration and the deformation are integrated for the construction of a head model. Besides, in the deformation step, we use a Delaunay triangulation based algorithm to add new vertices and re-triangulate the mesh. In the previous studies, Delaunay triangulation was used only for moving the positions of vertices with respect to obtained 3D points (not for adding a new vertex) and the degeneracy has not been considered either.

2 3D Reconstruction and Autocalibration

3D reconstruction of a scene using uncalibrated images provides a projective reconstruction which does not preserve the ratio of lengths, parallelism and ratio of areas. [12]. A metric reconstruction however, preserves all geometrical features of the scene except for an overall scale factor. To find the homography converting a projective reconstruction to a metric one, we need intrinsic camera parameters. Intrinsic camera parameters are related to the dual of absolute conic Ω^* via $\Omega^* = \mathbf{K}\mathbf{K}^T$. Ω^* and π_∞ are encoded in a concise form using a degenerate dual quadric which is called the absolute dual quadric $\mathbf{Q}^*_\infty$. The relationship between Ω^* and $\mathbf{Q}^*_\infty$ is given by $\Omega^* = \mathbf{P}\mathbf{Q}^*_\infty\mathbf{P}^T$. Remark that π_∞ is the null vector of $\mathbf{Q}^*_\infty$ which shows how it encodes Ω^* and π_∞. The above equation also gives the mapping between $\mathbf{Q}^*_\infty$ and Ω^* by the camera matrix. This means that any assumption about the intrinsic parameters of the camera can be transferred to a restriction on the value for the entries of Ω^* and $\mathbf{Q}^*_\infty$[5]. By imposing constraints on the entries of camera intrinsic matrix such as constant values for focal length, skew, aspect ratio and so on, we may define equations on unknown dual absolute quadric entries. This set of equations, if solved gives the absolute dual quadric which may then be used to find the camera intrinsic parameters. This algorithm which first introduced by Triggs [16] is a quadratic method but compared to other non-linear methods such as Kruppa equations [1] or modulus constraints [14], solving the equations is easier [3, 10, 18]. One of the problems here is that, the method is applicable if there are sufficient rotations about at least two non-parallel axes in the images and some translations in each direction. The ambiguity can be described as follows: since absolute conic lies on the plane at infinity, its images on the different views are related to each other by a planar infinite homography $\mathbf{H}_\infty$. This relationship is given in Equation 1.

$$\Omega^* = \mathbf{H}_\infty\Omega^*\mathbf{H}^T_\infty \tag{1}$$

However, $\mathbf{H}_\infty$ is related to camera internal and external parameters by $\mathbf{H}_\infty = \mathbf{K}\mathbf{R}^i\mathbf{K}^{-1}$. Assuming $\mathbf{d}_r$ to be the unit eigenvector of $\mathbf{R}$ we have $\mathbf{R}\mathbf{d}_r = \mathbf{d}_r$ and the eigenvector of $\mathbf{H}_\infty$ is v=Kdr; that is a one parameter family of solutions such as $\mathbf{v}\mathbf{v}^T$ satisfies Equation 1 and the solution is degenerate and not unique. To obtain a unique solution, we add two more restrictions: skew s is 0 and aspect ratio γ is 1. These restrictions are reasonable assumptions in most real imaging systems. Assuming $s = 0$ gives a unique solution when the rotation is about an axis which is not parallel to one of the axis of the camera. On the other hand, if the rotation is about an axis parallel to $\mathbf{y}$ axis, assuming $s = 0$, Ω^* becomes as:

$$\Omega^*(\mu) = \begin{bmatrix} \alpha^2_{xy} + x^2_0 & x_0y_0 & x_0 \\ x_0y_0 & \alpha^2_{xy}(1+\mu) + y^2_0 & y_0 \\ x_0 & y_y & 1 \end{bmatrix} \tag{2}$$

where x_0 and y_0 are camera principle point coordinates and α_{xy} is the camera resolution in the unit of length in both x and y directions (we have assumed that they are equal)[4]. In order to find the camera internal parameters, a set of equations are established among camera matrices and $\mathbf{Q}^*_\infty$. The relationship between the elements of and $\mathbf{Q}^*_\infty$ are rewritten as $\Omega^*_{ij} = (\mathbf{P}\mathbf{Q}^*_\infty\mathbf{P}^T)_{ij}$ where i and j subscripts show the element at row i and column j. For each image pair, we have a set of equations of the above type. Since the relation between the absolute conic and dual absolute quadric is given

in homogeneous coordinates, it is valid only up to a scale factor. In order to eliminate this scale factor, cross product of the entries of can be used. In this study, in order to solve the equations in an iterative form, we prefer to use a cost function and minimize it subject to some restrictions. The cost function is given by $\sum_i \Omega^*_{ij} \bigwedge \mathbf{P}_i \mathbf{Q}^*_\infty \mathbf{P}_i^T$ where $\bigwedge$ denotes the cross multiplication to eliminate scale factor. The restrictions imposed on the equations are: rank 3 restriction for $\mathbf{Q}^*_\infty$ by making its determinant to be equal to zero, $\|\Omega^*\| = 1$, $\|\mathbf{Q}^*_\infty\| = 1$, $s = 0$, $\gamma = 1$. Then Ω^* becomes as follows.

$$\Omega^*(\mu) = \begin{bmatrix} \alpha_{xy}^2 + x_0^2 & x_0 y_0 & x_0 \\ x_0 y_0 & \alpha_{xy}^2 + y_0^2 & y_0 \\ x_0 & y_y & 1 \end{bmatrix} \quad (3)$$

The homography matrix converting projective camera matrices into a metric one is found from $\mathbf{Q}^*_\infty = \mathbf{H}\hat{\mathbf{I}}\mathbf{H}^T$ where $\mathbf{H}^{-1}$ is a 3D point homography taking the projective coordinate to Euclidean coordinates. Decomposition of $\mathbf{Q}^*_\infty$ may be performed by eigenvalue decomposition [3].

3 Deforming a Generic Head Model

The geometric deformation of the generic head model which is in fact a mesh is done in five steps: global and local scaling of the generic mesh, coarse and fine deformation, and mesh improvement. In the first two steps, the generic mesh is scaled so that each face segment between facial features corresponds to the size of the same region in the images. Step 3 and Step 4 deforms features and face surface and Step 5 refines the mesh. The deformation process requires local re-triangulation in the areas with high curvatures. This is achieved by locally applying Delaunay triangulation method. The inputs are two sets of points, a generic model of the human head in the form of a 3D mesh and texture values of the specific face given in cylindrical format. The output is the textured 3D model of the specific head.

3.1 Input Data

From the images of a specific person, two sets S_1 and S_2 are formed. These sets are composed of 3D points from the surface of the head and the 3D coordinates reconstructed by using the calibrated images as mentioned in Section 2. The elements of S_1 are associated and labeled with the facial features and they indicate the position of the point on the face. The points in S_1 are obtained by locating and matching the facial feature points and by reconstructing their 3D coordinates. Locating the feature points is done manually in the images and then these locations are refined by searching in a 7×7 window around each point and finding best matches using normalized cross-correlation. The second set of input points S_2 is composed of random points from facial areas in the images and these points are not labeled. The points in this group are obtained by applying an interest point detector (Harris corner detector) and then finding the corresponding point in another image by using the disparity consistency and cross correlation. The 3D points are reconstructed from the correspondences.

3.2 Global and Local Scaling

Since the mesh model and the input points may be in different sizes, the first step performs a scaling operation based on the maximum and minimum coordinate values of the points in the input set S_1. No local displacement of the facial features is done in this step. The locations and sizes of facial features may vary individually. Therefore, the generic mesh is assumed to be divided horizontally into 4 parts which are separated by the boundary points as the corners of lips, the nose tip, and the corners of eyes. The vertices corresponding to these boundary points, that are called as boundary vertices are displaced such that their projections on the images overlap with the positions of their corresponding facial features. The locations of the remaining intermediate vertices are computed by interpolating over the displacement of the boundary vertices. Remark that all of the boundary points are the elements of S_1.

3.3 Coarse Deformation

In this step, the points from S_1 are used to deform the generic mesh. Since the points are labeled and their 3D coordinates are available, we displace the corresponding vertices and bring them to the 3D locations given by these input points. Two main issues that are considered in this displacement are as follows. Anthropological relationship between face features should be preserved. Displacing a vertex should not create unrealistic warps such as corners with sharp curvatures. In order to impose both of the restrictions, we consider the mesh as an interconnected set of vertices in which displacing a vertex affects other (especially neighboring) vertices. This force is simulated by considering the edges connecting the vertices as mass-less spring in equilibrium states. Total force ξ generated by the springs around a vertex determines the new position

of a vertex. In this specific mesh deformation application, we have the advantage of knowing *a priori* the general form of the face. Therefore, in order to preserve the topology of the mesh, a second restriction is added based on the principal curvatures at a given point on a surface which measures the maximum and minimum bending of the surface at that point. To find out the principal curvatures, we compare the normals to facets about a vertex and the minimum and maximum angles between these normals define the curvature $\zeta = Max(Diff(n_i, n_j))$ where n_k is a normal about vertex k. Considering both issues mentioned above, a cost function is defined to adjust the location of other vertices in the mesh whenever a vertex is moved to a new position. The cost C function is defined as $C = \Delta(\zeta) + \Delta(\xi)$.

3.4 Fine Deformation

Let $\mathbf{P} = \{p_1, p_2, ..., p_n\}$ be a set of points in the Euclidean space. A triangulation $\mathbf{T}$ associated with $\mathbf{P}$ is the Delaunay triangulation of $\mathbf{P}$ iff every $(m-1)$-open ball circumscribed about an m-simplex contains no other point from $\mathbf{P}$. A Delaunay triangulation maximizes the minimum angle of the triangles involved in any triangulation of a set of points $\mathbf{P}$. These properties make Delaunay triangulation a powerful tool for mesh generation. The fine deformation which is based on local modification of the generic mesh using Delaunay triangulation is applied only to the points in S_2. In this step, we do not expect a drastic change in the form of the generic mesh, because it has already undertaken three deformation steps and the new points should add slight modifications. On the other hand, since there is no one-to-one correspondence between the points in S_2 and the mesh vertices (in contrast to the points in S_1), displacing a vertex to a new position may cause encroaching on an edge by another one. Furthermore, this generates small concave holes and artificial crease lines when mapping texture to facets. As a solution to this degeneracy, we can still use the approach explained for the coarse deformation which provides a gradual displacement of a vertex and propagating the displacement to the neighboring vertices. However, we do not have correspondences between the elements of S_2 and mesh vertices, and in fact there may be no corresponding vertex for a point at all. Therefore, partial re-triangulation of the mesh may be required. We start with a search to find the nearest vertex to the 3D position of a point from S_2. We use a threshold for the distance between the 3D point and the nearest vertex. This threshold value bans large changes in the mesh. Hence, if the 3D point is within the open ball $\mathbf{B}(k, t)$ defined by the nearest vertex k and threshold t, then the vertex is moved towards the 3D point using the same algorithm as used in the coarse deformation. Otherwise, a new vertex is added to the mesh and the mesh is re-triangulated. The steps listed above are summarized in Algorithm 1.

Algorithm 1 Steps in the fine deformation of the mesh.

1 For each 3D point $t \in S_2$ **do**
  1.a k = NearestVetex(t)
  1.b **IF** t in $\mathbf{B}(k, t)$ **THEN**
    Move k towards t using springs
  ELSE IF Distance(t,Mesh) $<$ τ **THEN**
    Insert(t)

2 End For

During the insertion a Delaunay cavity is created. Besides, we consider several possible cases depending on the location of the vertex and the topology of the mesh in the particular region. The algorithm to find Delaunay cavity for adding a vertex to the mesh is given in Algorithm 2.

Algorithm 2 Finding Delaunay cavity.

1 For each 3D point t **do**
  1.a Find all facets k which includes the mapping of t in their diametral ball $\mathbf{D}(k)$.
  1.b Find the Delaunay cavity from the union of the facets obtained in step **1.a**.

2 End For

Delaunay cavity in step 1.b of the Algorithm 2 is defined as the union of all facets having the mapping of the new point in their diametral ball. The cavity is created by removing common edges from the facets. Re-triangulation of the mesh is done by adding edges $< t, m_i >$ where m_i is a vertex from the Delaunay cavity. This kind of triangulation preserves the slight changing rate in the curvature when the new vertex is an almost plain surface. In the case of the regions with high curvature, a threshold value is used to reject points far from the surface of the mesh. This threshold value is defined using radius-edge ratio of the facets falling in the Dealunay cavity of that point. Radius-edge ratio of a facet k is defined as the ratio of the length of the radius of $\mathbf{D}(k)$ to the length of the shortest edge of k. Two main degenerate cases may happen in this type of triangulations. First case occurs when an edge added during re-triangulation of a Delaunay

cavity encroaches on a boundary edge of the cavity. It happens when the vertex is not visible from the new vertex. This can be stated as follows. If an empty ball **B** contains n points in its boundary then the n-simplex formed from their convex hull is present in their Delaunay triangulation. However in degenerate cases the empty ball criterion does not determine a triangulation of the input points. Instead it defines decomposition as a unique collection of Delaunay protopes. Second case happens when the sight line from t to a cavity boundary vertex pierces the facets of the cavity area. For the first type of degenerate cases, we have changed the definition of Delaunay cavity to include only those points in its boundary that does not cause an edge encroaching during triangulation. This also preserves the relationship between convex hull and the Delaunay cavity. Second case of degeneracy affects the topology of the mesh. Re-triangulation in this case violates the boundaries of the local deformations. Most unwanted effect of these deformations is creating concave areas in a convex sub-region. Our modification in the definition of the 3D Delaunay cavity also prohibits this type of degeneracy.

3.5 Mesh Improvement

Re-triangulation and displacing vertices may create some small and skinny facets. Skinny and small facets affects deformation algorithm and slows down the rendering process. First step in mesh improvement attempts to eliminate these facets. To detect skinny facets the radius-edge ratio is used. This criteria is defined as the ratio of the length of the radius of diametral ball of each triangle to its shortest edge. Finally, a Laplacian smoothing is applied to the deformed mesh to improve its quality. We have modified the Laplacian smoothing algorithm also to impose restrictions which define a Delaunay triangulation. This restriction can be imposed by adjusting the location of each vertex through the its barycentric coordinates.

4 Experiments and Results

Two meshes are available as models of generic human head on which we apply the described deformation process. The first mesh is a coarse one with 168 vertices and 275 faces while the second mesh includes 1355 vertices and 1739 faces. We have performed two types of experiments. In the first type of experiments, we have used digital images of people and the reconstruction results are evaluated qualitatively. For the quantitative evaluation, the data obtained from a 3D scanner is used only to deform the available dense mesh.

Figure 1: A sample sequence of input images.

For qualitative evaluation, we have modeled several human heads although we present here the information related to only one of them. Figure 1 shows a subset of input images. The 9 points of the facial features which are elements of S_1 are shown in Figure 2 for this person. 274 point pairs are detected as the elements for S_2.

Figure 2: Location of feature points.

However, 31 points are rejected during the deformation due to their distance from the generic mesh. Only 29 points cause a face split in this experiment. Remark that this number depends on the threshold value used in displacing a vertex and the density of vertices in the mesh. A sample set of images projected from the final reconstructed head model is given in Figure 3.

In order to define an objective reliability criteria for the assessment of the algorithm, we also use the 3D data obtained from range scanning of two different faces as ground truth and verify the correctness of the results of our algorithm with respect to them. The data obtained form range scanning two human faces are available in the form of a PLY file which includes both 3D values of vertices and polygons which relate vertices to each other. The files that we are using have 302948 vertices, 605086 polygons and 303380 vertices, 605902 polygons, respectively. The range scan data of one of these faces have been rendered in Figures 4. To verify the accuracy of deformation depth values of some points from different facial areas are compared in our model and the range scanned data. To avoid any error in comparison we connect points form the first set S_1 to each other and then divide the connecting line by 4

Figure 3: Reconstructed head model.

and obtain x and y coordinates of the testing points. The z coordinate values of these points are found from our model and range scan data and compared to each other. The average error in each facial region is given in Table 1. In these experiments, we scale the y coordinate values of the vertices in the generic mesh to range between -1 and 1. x and z coordinates are scaled using the same factor. Considering this fact, 0.01 in error values means that the displacement is 1/200 times of head height. The error values are higher in the first image. Since the generic mesh represents the face model of a male, it may be justified by the need for larger changes in the mesh compared to the second image. The reconstructed faces using the dense mesh are displayed in Figure 5.

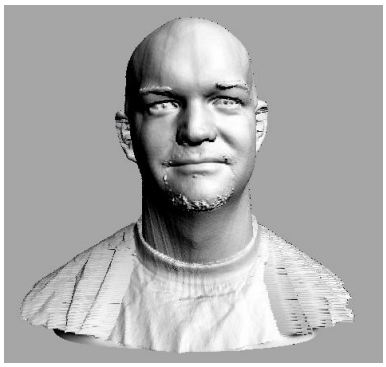

Figure 4: Rendered Cyberware head data.

Figure 5: Reconstructed head model using Cyberware data.

5 Conclusions

In this paper, we describe a method for 3D human head modeling using uncalibrated images. Having uncalibrated images as input makes the method generic and applicable for any system setup on one hand, but introduces the degeneracy problem in estimating internal camera parameters and camera motions on the other hand. We describe a solution to the degeneracy problem in camera calibration for the most common case. The second main issue in to describing geometry of human heads is complexity of the shape and existence of poor texture on the facial skin. Our solution for the reliable reconstruction of facial geometry is based on a generic head model. To obtain the geometry of any particular head, we deform the generic model which consists of a mesh. To propagate the movement of a vertex to its neighboring vertices in a controlled manner, we consider the physical based model in which the edges connecting vertices to be massless springs react-

Table 1: Average error in face modeling using Cyberware data.

	Eyes	Mouth	Cheeks
First Image	0.032	0.024	0.051
Second Image	0.028	0.027	0.035

ing to the forces applied onto them. We apply a two stage deformation based on the 3D points computed by matching points from images of a specific person along with the described calibration algorithm. In the first stage, we use a small number of labeled points whose correspondences with the mesh vertices are known. In the second phase, a cloud of non-labeled points is input to Delaunay meshing algorithm to locally increase the mesh density by inserting new vertices while keeping the consistency of the model. The models generated using this method is very suitable for animation and pose independent identification.

References

[1] Faugeras O., Luong Q. , Maybank S., 1992. "Camera self-calibration: Theory and experiments. In proc. European Conference on Computer Vision, LNCS 588, pp. 321-334. Springer-Verlag.

[2] Fua, P., 2000, "Regularized bundle-adjustment to model heads from image sequences without calibration data", International Journal of Computer Vision, pp. 153-171.

[3] Hartley, R, Zisserman, A 2000. Multiple View Geometry in Computer Vision, Cambridge University press.

[4] Hassanpour R. and Atalay V., 2002. "Head Modeling with Camera Auto-calibration and Deformation", 7th Int. Fall Workshop on Vision, Modeling and Visualization-VMV 2002, Erlangen, Germany, pp.3-11

[5] Hassanpour R., Atalay V., 2002. "Camera Auto-Calibration using a Sequence of 2D Images with Small Rotations and Translations", Proceedings of the 17th Int. Symposium on Computer and Information Sciences, Orlando, Florida, USA, pp.215-219, CRC Press.

[6] Akimoto T. , Suenago V., 1993, Automatic creation of 3D facial models, IEEE trans, Computer Graphics and application vol. 3 pp 16-22

[7] Luo S., King W. R., 1996, Automatic Human Face Modeling in Model-Based Facial Image Coding, Proc. of AUstralian New Zealnd Conf. on Intelligent Information Systems, pp 167-171.

[8] Yan J. , Gao W. , 1998 , Deformable Model-Based Generation of Realistic 3-D Specific Human Face, Proceedings of ICSP, pp 857-860.

[9] Lengagne, R, Fua, P, Monga, O, 1998. "3D face modeling from stereo and differential constraints", International Conference on Pattern Recognition, pp. 148-153.

[10] Luong Q., 1992. "Matrice Fondamentale et Autocalibration en Vision par Ordinateur", PhD thesis, Universit de Paris-Sud, France.

[11] Pighin F. Hecker J., 1998 , Synthesizing realistic facial expressions from photographs, Computer Graphics, Annual conference Series, pp 75-84.

[12] Otlu B., Hassanpour R., Atalay V., 2002. "Consistency of projective reconstructions from an image pair", 9th International Workshop on Systems, Signals and Image Processing, Manchester Town Hall, UK. pp. 376-381.

[13] Parke F., 1996, Computer Facial Animation, AKPeters, Wellesley, Massachusetts.

[14] Pollefeys, M., Van Gool L., Oosterlinck A., 1996. "The Modulus Constraint: a new constraint for self-calibration", In Proc. International Conference on Pattern Recognition, pp. 31-42.

[15] Thalman N., Minh H., Angelis M. and Thalman D., 1989, Design transformation and animation of human faces, Visual Computer (5):32-39.

[16] Triggs, B., 1997. "Autocalibration and the absolute quadric", In Proceedings of the Conference on Computer Vision and Pattern Recognition, Puerto Rico, USA, pp. 607-614.

[17] Blanz V., Vetter T., 1999 A morohable model for the synthesis of 3D faces, In computer Graphics, Annual Conference Series,SIGGRAPH, pp 187-194

[18] Zeller C., 1996. "Projective, Affine and Euclidean Calibration in Computer Vision and Application of Three Dimensional Perception, PhD thesis, Robot Vis Group, INRIA Sophia-Antipolis.

[19] Zhang Z. 2001. "Image-based Modeling of Objects and Human Faces", Proceedings of SPIE Vol.4309 , Videometrics and Optical Methods for 3D Shape Measurement, pp. 1-15.

[20] Zhang, Z. and Isono, K. and Akamatsu, S.,1998, "Euclidean Structure from Uncalibrated Images Using Fuzzy Domain Knowledge: Application to Facial Images Synthesis", International Conference on Computer Vision, pp. 784-789.