Head Modeling with Camera Auto-calibration and Deformation

Reza Hassanpour
Department of Computer Engineering,
Cankaya 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

A 3D head modeling method from a sequence of 2D images is described. The views from which the input images are acquired are not calibrated. Therefore, an auto-calibration method for a sequence of images with small rotations and translation is developed. For this purpose, we have modified an already existing auto-calibration algorithm to incorporate known aspect ratio and skew values to make it applicable for small rotation around a single axis. We apply this auto-calibration technique to head (face) modeling. Three dimensional positions of known facial features computed from two dimensional images are used to deform a generic head model by using a spring based energy minimization method.

1 Introduction

There is a significant body of work on three dimensional (3D) analysis of images in recent years [1,2,3,4,5]. Many application areas such as computer animation, medical imaging and teleconferencing require 3D information of the environment that can be simulated on our machines. The first step in 3D analysis is modeling 3D objects. This step, in general has been proven to be a difficult task to accomplish. The reason for the difficulty is the need for very sensitive and reliable measurements that can either be obtained by using complicated measuring devices to find the range data (depth information) or by developing algorithms to extract this information from two dimensional (2D) images using different features of the objects. To deal with these problems, the researchers have either limited the 3D objects to be modeled to a small and known class of natural or artificial entities [4] or imposed restrictions on the parameters of the problem [2,3]. These restrictions make possible to incorporate the knowledge about the camera into the system. Knowledge about the camera includes its physical properties (intrinsic parameters) and its position and orientation (extrinsic properties). Between two images, the position and orientation of the camera may change. The change in position is a translation and orientation. Orientation of camera may change in 3 axes of the coordinate system. These changes altogether let us to find the unknown camera parameters under the assumption that all or at least some of the camera parameters remain constant. However, if the camera rotation is not about at least two non-parallel axes, it is not possible to find some of the camera intrinsic parameters, such as the aspect ratio of scene and camera. In order to solve this problem, some restrictions such as known aspect ratio or knowledge about scene should be incorporated. In this study, assuming fixed values for camera internal parameters, we develop an algorithm to compute intrinsic parameter matrix when the rotation is only about the vertical axis and translation is small. This type of rotation and translation is quite common when the images are taken with a handheld camera by a person standing in (almost) a fixed place. In this study, we describe an auto-calibration algorithm with fixed but unknown camera parameters. Employed auto-calibration algorithm is initially proposed by Triggs [2] and we have modified Triggs’ algorithm to incorporate known aspect ratio and skew values to make it applicable for small rotation around a single axis. We apply this auto-calibration technique to head (face) modeling. In the major studies for head modeling, either calibration has not been performed [4] or calibration is totally or partially performed on images other than the ones used for head modeling [12].

The organization of the paper is as follows. Section 2 introduces image formation in a pinhole camera which is followed by the camera model and its parameters in Section 3. A brief introduction to auto-calibration and reconstruction algorithms is given in Section 4. We present the details of our auto-calibration algorithm and the experimental results in the two consequent sections. The details of applying the algorithm in deforming a head model are given in sections 7 and 8. Finally, Section 9 has the experimental results of 3D face modeling.

2 Image Formation Process

A single image, despite its rich content, does not contain enough information to reconstruct the 3D scene. This is because of the image formation process which maps a 3D scene onto a 2D image. In this mapping process, the depth data is lost, however if two or more images are available, the 3D point can be obtained from the intersection of sight lines as demonstrated in Figure 1.

Figure 1: 3D location of a point from the intersection of sight lines. M is a point in 3D and C and C’ are the center of projection of two cameras.

In this process, we need to know:
- corresponding points in the images,
- orientation and relative location of the camera,
- internal camera parameters.

Corresponding points in the images may be found by detecting interest points in both images and matching them by means of some similarity measure [6]. Camera orientation and location, is called as the extrinsic parameters of camera, which relate the external position of camera to the structure of the corresponding sets of image points. Intrinsic camera parameters define the physical properties of the camera.
3 Camera Model

A simple model of a CCD camera may be defined by the pinhole camera. The geometric process for image formation in a pinhole camera is illustrated in Figure 2. In the simplest model, we can assume that the center of projection is the origin of a Euclidean coordinate system and the plane on which the image is created lies at distance $f$ along the $z$ axis from the $x$-$y$ plane and parallel to it. This plane is called image plane. In this model, the center of projection is called the camera center, the line passing from the camera center and perpendicular to image plane is called the principal axis.

![Figure 2: Image formation in a pinhole camera.](image)

The intersection of principal axis and image plane is called principal point. If we represent the coordinates in homogeneous forms, then the projection in the camera can be expressed in matrix form as:

$$
\mathbf{x} = \mathbf{P} \mathbf{X}
$$

where $\mathbf{x}$ is the image of world point $\mathbf{X}$ and $\mathbf{P}$ is the camera projection matrix.

Camera projection matrix associates the world coordinates of points to the image coordinates. This means that the association process is affected by the physical properties of the camera. First property in this regard is the focal length $f$ of the camera. In addition, the coordinates on the image plane depends on the scanning process in the camera and shape and size of the pixels. Camera pixels may be non-rectangular which then includes a skew effect in the projection. Furthermore, the projection center may not correspond to the origin of the coordinate system in the image plane. Therefore, the image coordinates should be scaled with pixel width and height. All these physical properties of the camera are called intrinsic parameters. A projection matrix considering the intrinsic parameters is given by Equation 2.

$$
\mathbf{K} = \begin{bmatrix}
\alpha_x & s & x_0 \\
0 & \alpha_y & y_0 \\
0 & 0 & 1
\end{bmatrix}
$$

where $\alpha_x$ and $\alpha_y$ are the focal length of the camera in terms of pixel dimensions in $x$ and $y$ directions respectively, $s$ is the skew parameter and $x_0$ and $y_0$ are location of the principal point. The coordinate system of the camera is related to the world coordinate system via a translation and a rotation. The coordinates of a point in the camera frame is related to its coordinates in the world frame with the Equation 3 where $\mathbf{C}$ is the coordinates of camera projection center in world coordinates and $\mathbf{R}$ is rotation matrix.

$$
\mathbf{X}_{\text{rec}} = \mathbf{R} (s \mathbf{C})
$$

The location and direction of the camera are given by the extrinsic parameters.

Considering both camera position and orientation and its physical features, the mapping from world coordinates to image plane coordinates is given by the following equation:

$$
\mathbf{P} = \mathbf{K} \mathbf{R} [\mathbf{l} \mid -\mathbf{C}] 
$$

which can be simplified as:

$$
\mathbf{P} = \mathbf{K} [\mathbf{R} \mid \mathbf{t}]
$$

where $\mathbf{t}$ indicates the translation. Equation 6 combines the intrinsic and extrinsic parameters of a projective camera and is called camera matrix. $\mathbf{K}$ and $\mathbf{R}$ matrices can be found from camera matrix by decomposing it to an upper-triangulation matrix and an orthogonal matrix using RQ decomposition.

4 3D Reconstruction and Auto-calibration

Given two images and a set of matching points such as $\mathbf{x}_i$ and $\mathbf{x}_i'$, we want to find point set $\mathbf{X}_i$ such that its mappings in first and second images are $\mathbf{x}_i$ and $\mathbf{x}_i'$ respectively. From definition of a camera matrix we have $\mathbf{x}_i = \mathbf{P}_i \mathbf{X}_i$ and $\mathbf{x}_i' = \mathbf{P}_i' \mathbf{X}_i$. Therefore, our first step should be the computation of camera matrices $\mathbf{P}$ and $\mathbf{P}'$. To find these matrices, we need to establish a geometrical relation between the corresponding points in the images. This relation is given by a matrix called fundamental matrix. Mathematically the matrix is defined as:

$$
\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0 
$$

In this relation, $\mathbf{F}$(fundamental matrix) is a $3 \times 3$ matrix and $\mathbf{x}_i$ and $\mathbf{x}_i'$ are the points from the images. This equation shows that given any point from first image, its corresponding point in the second image should lie on a line which is called epipolar line. The reason for this restriction is that the camera centers $\mathbf{C}$ and $\mathbf{C}'$ and world point $\mathbf{X}$ lie in a plane, epipolar plane which also includes the sight lines from $\mathbf{X}$ to image planes and the line connecting two camera centers together (baseline) as shown in Figure 3.

![Figure 3: Camera centers $\mathbf{C}$ and $\mathbf{C}'$, and sight lines lie in the epipolar plane defined by $\mathbf{CXC}'$. $e$ and $e'$ are the epipoles.](image)

Given a point $\mathbf{x}$ and the camera centers, it is possible to find the intersection line of epipolar plane with the second image plane which is a line passing through point $\mathbf{x}'$. The relationship defined by fundamental matrix depends only on projective coordinates of the points in the images and not on the Euclidean measurements such as angles and lengths. This means that $\mathbf{F}$ does not depend on the choice of a coordinate frame and any projective transformation of world points does not have any effect. But, these transformations affect camera projection matrices $\mathbf{P}$ and $\mathbf{P}'$. Therefore, although a pair of camera matrices like $\mathbf{P}$ and $\mathbf{P}'$ uniquely determine the fundamental matrix, the reverse is not true. Therefore, given a set of corresponding points in the two images, the camera matrices and 3D scene can be reconstructed up to a projective transform. To upgrade a projective reconstruction to a more realistic one with a geometric structure we need an orthogonal transformation plus a translation. This transform leaves the plane at infinity unchanged and transforms the absolute conic which is a specific conic, into itself. The absolute conic is defined as follows:

The absolute conic $\Omega$ is a degenerate point conic on the plane at infinity $\pi_\infty$. Since $\pi_\infty$ is defined as $[0 \ 0 \ 0 \ 1]^T$ the points on $\Omega$ satisfy

$$
\mathbf{x}'^T \mathbf{x} = 0
$$

$w=0$
Every known angle or ratio of lengths imposes a constraint on the absolute conic and if enough constraints are available, it can be uniquely determined. If the absolute conic is identified it can be used to upgrade an Affine structure to a metric one. The matrix for upgrading from a projective reconstruction to a metric one is called a Homography matrix. Dual of the absolute conic $DAC$ is related to the camera intrinsic parameters via Equation 9.

$$\Omega^* = K K^T$$

$DAC$ and $\Omega_\alpha$ are encoded in a concise form using a degenerate dual quadric which is called the absolute dual quadric $Q_{\inf}^*$. The relationship between $\Omega$ and $Q_{\inf}^*$ is given in Equation 10. $\Omega_{inf}$ is the null vector of $Q_{\inf}^*$.

$$\Omega = P Q_{\inf}^* P^T$$

### 5 Proposed Auto-calibration Method

Camera projection matrix maps absolute dual quadric to $DAC$ via Equation 10. This means that any assumption about the intrinsic parameters of the camera can be transferred to a restriction on the value of the entries of $DAC$ and absolute dual quadric. 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 be used to find the camera intrinsic parameters. The only problem here is that, the method is applicable if there are rotations about at least two non-parallel axes in the images with at least 30 degrees in each rotation axis and some translations in each direction. To avoid this problem, we have added two more restrictions: skew is 0 and aspect ratio is 1. These restrictions are reasonable assumptions in most real imaging systems. The relationship between the elements of $\Omega^*$ and $Q_{\inf}^*$ are given in Equation 11.

$$\Omega_{\inf} = (P Q_{\inf}^* P^T)_{ij}$$

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 type Equation 11. Since the Equation 14 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 $\Omega$ 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 in Equation 12.

$$\sum \left( \Omega_{\inf}^* \wedge P \right) = 0$$

where $\wedge$ denotes the cross multiplication to eliminate scale factor. The restrictions imposed on the equations are:

- rank 3 restriction for $Q_{\inf}^*$ by putting its determinant equal to zero,
- $\|\Omega\| = 1$,
- $\|Q_{\inf}^*\| = 1$,
- skew is 0,
- aspect ratio is known and equal to 1.

After adding skew=0 and aspect ratio=1 restrictions the $\Omega$ matrix have the following values in terms of camera intrinsic parameters

$$\begin{bmatrix}
\alpha^2 + x_0^2 & x_0 y_0 & x_0 \\
x_0 y_0 & \alpha^2 + y_0^2 & y_0 \\
x_0 & y_0 & 1
\end{bmatrix}$$

where $x_0$ and $y_0$ are camera principle point coordinates and $\alpha$ is the camera resolution in the unit of length in both $x$ and $y$ directions. (we have assumed that they are equal). In order to solve this set of equations, we have used quadratic sequential programming.

### 6 Experimental Results for Auto-calibration

The method described in the previous section is implemented using MATLAB. We have imposed the restrictions as equality constraints and used quadratic sequential programming to solve them. The method is tested on a sequence of images taken from an object on a rotating plate. In each image, we have rotated the plate for 20 degrees and change the location of the camera to have some translation (although small) in the location of the object. Three views chosen from the sequence are shown in Figure 4.

The camera calibration matrix $K$ is computed after finding the fundamental matrix from matching points using 8 points algorithm [10] and then determining the camera projection matrices [11]. Camera intrinsic matrix $K$ is given below.

$$\begin{bmatrix}
1234 & -0.002 & 657.1 \\
0 & 1234 & 519.6 \\
0 & 0 & 1
\end{bmatrix}$$

We have also reconstructed some corner points of the object and find their 3D coordinate values. The homography matrix converting projective camera matrices into a metric one is found from the Equation 13.

$$Q_{\inf}^* = H H^T$$

$H^{-1}$ obtained from Equation 13 is a 3D point homography taking the projective coordinate to Euclidean coordinates. Decomposition of $Q_{\inf}^*$ may be performed by eigenvalue decomposition [11]. The homography matrix $H$ is given as follows.

$$\begin{bmatrix}
15478 & 0.4867 & -0.9853 & -0.2779 \\
-89451 & 0.8472 & -0.1704 & 0.3032 \\
280.9 & 235 & -0.0008 & 182.7 \\
-563.2 & -1214.8 & -0.0017 & 35.351
\end{bmatrix}$$

Figure 5 shows the corner points we have selected to find their corresponding 3D coordinates. These points in our experiment are given manually. The reconstructed values of the coordinates are as follows.

### Experimental Results for Auto-calibration

![Three samples from the sequence of input images.](image-url)
Metric reconstruction is valid up to a scale factor which may be found only from some a-priori knowledge about the objects in the image. This fact is clear from the values given above, but other than this, despite the small rotation angle about only a single axis and a small translation value, the results are satisfactory. If the matching pairs are reliable (which is the case in natural images with a good texture) this method may be followed by a bundle adjustment to increase accuracy in point location in 3D.

Figure 5: Location of reconstructed points.

Figure 6 demonstrates reconstructed corner points marked in Figure 5 from front and top views.

Figure 6: Reconstructed points form front and top views.

7 Mesh Deformation

To create the 3D model of the head of a specific person, we need the coordinates of all points covering the outer surface of his/her face. However, such data is hard to obtain, since most of these points are either not visible in all input images or due to a uniform texture or light changes, it is almost impossible to find matching pairs in the images. Having some sort of default and approximate values for the points can lead to find the 3D coordinates of only a limited number of major feature points and interpolate remaining ones based on the values of the computed points and the default values. On the other hand, these default values can be used to reject erroneous coordinate values and to keep the general form of the face and later deformations in an acceptable range. In this study, we use a generic mesh of human face consisting of 113 vertices and 168 triangles as the default values. We intend to use a more dense and accurate mesh in the future but for the time being, this sparse mesh can simplify the implementation and testing of our algorithms. To deform this generic mesh and make it look like the face of a specific person, we proceed with the following steps:
1. global scaling of the mesh,
2. facial feature extraction and local scaling,
3. determining main facial points and face feature deformation.

Global scaling of the mesh is performed by considering the bounding box covering the area between topmost, bottommost, leftmost, and rightmost points of the face and scaling the mesh in $x$ and $y$ directions accordingly. This boundary is obtained by segmenting the face using a skin filter and fitting the best ellipse to the segmented region [13]. Global scaling in $z$ direction is obtained by the average of $x$ and $y$ scale factors. Figure 7 shows the boundaries of global scaling for a sample input face image.

Figure 7: Bounding box of a face.

Different parts of the face may be in different forms and sizes in each face. For example, some people have wide forehead while some other have very narrow one. The lengths of nose and chin also can be different. The next step in our scaling algorithm divides the face into five regions. The first region is between the start of forehead and uppermost point of eyebrows. The second region is between lower border of the first region and the corners of the eyes. Third region starts from the lower border of the second region and ends at the nose tip. Fourth region is between the lower border of the third region and the line connecting corners of lip. Finally, the last region is below the fourth region and above the tip of chin. These regions are shown in Figure 8 for the sample input face.

Figure 8: Location of local scaling regions.

The points which are used to determine the borders of these local scaling regions are given in Figure 9.

Figure 9: Facial feature points used in local scaling.
The last step in our deformation algorithm is to determine the 3D position of certain feature points and displace the neighboring points to have a realistic head (face) model. 3D coordinates of these feature points are obtained by means of their Euclidean reconstructed values using the homography matrix computed as described in the previous section. As mentioned in self calibration section, the coordinate values are true only up to a scale factor. In this study, we obtain the scale factor from the front and profile images. In order to avoid unrealistic changes in facial features, we propagate any change in the position of a vertex to all of its neighbors. This propagation is performed by considering the edges connecting the vertices together to have a spring like reaction to change the lengths. However, since the facial features may not change in any direction freely, we also impose another restriction: the changes in the shape of the features should be minimum. This restriction can be imposed, for example by minimizing the changes in the directions of the normals to facets in the mesh as demonstrated in Figure 10.

\[
C = \sum \left( \delta l_{ix} + \sum \delta l_{ij} \right) + \sum \sum \delta n_{kl}
\]

where \( \delta l_{x} \) is the change of length in the edge connecting vertex i to current vertex x , \( \delta l_{ij} \) is change of length in the edge between vertex i and vertex j and \( \delta n_{kl} \) is the total change of the angles between normal to facet k and its neighboring facet l. The cost function is minimized subject to the following restrictions:
- change of angle between normals of two neighboring facets should be less than a threshold value,
- change of length of an edge should be less than a threshold value,
- the changes are not propagated beyond the borders of a facial feature.

We have implemented the last restriction by splitting the boundary vertices and defining very large spring constants for them. For example a vertex \( v \) in the boundary of nose and chick is split into \( v_1 \) and \( v_2 \) where \( v_1 \) belongs to nose and \( v_2 \) belongs to chick. If we try to change the location of a vertex in the nose, because of large spring constant the change does not propagate to chick.

8 Texture Mapping

Each triangle in the mesh is to be covered with a texture from the input images. For each triangle, the texture is taken from the image with the nearest viewing direction. To find these directions, we use the normals to the triangles. Since the 3D coordinates of the vertices are available, we can find the normal to each of them easily.

9 Implementation and Experimental Results for Head Modeling

The method for head modeling is tested on a sequence of images taken with a camera having small translations. Sample images from the sequence are shown in Figure 11.

Figure 11: Sample input images captured with the same camera.

The camera calibration matrix \( K \), the camera projection matrices and the homography matrix \( H \) are computed as explained in Section 6. Mesh deformation is done only in one level of neighboring vertices. Propagating changes to further vertices is necessary in highly curved areas especially if the generic mesh is a dense one. However, this propagation should be limited by the boundaries of the features and smaller than threshold effects. Local scaling which is carried out as the second step in our deformation algorithm has a major impact on reducing highly propagated changes. Figure 12 and Figure 13 show the results we obtained after local deformation, and face feature deformation steps.

Figure 12: Locally scaled mesh has been super imposed on the image.

Figure 13: Mesh with adjusted feature points super imposed on the front and profile images.

Since we are using a sparse mesh with a few number of vertices in this study, the approximated 3D model is not so exact. For example, in adjusting the position of nose tip, the model fails to show the curvature and a sharp nose tip is generated but the approximation and propagation of deformation to other vertices are acceptable. In addition, we have applied the deformation to
only a limited number of feature points which again is clear from the output images. For example, the lower lip and the curvature below the lower lip in the profile image do not conform with the input image. We plan to use a more dense mesh and apply our deformation algorithm in a larger number of levels for a better 3D structure. Figure 14 shows the texture mapped mesh after all deformations.

Figure 14: Textured mapped mesh after deformation.

10 Conclusion

A 3D head modeling method from a sequence of 2D images is described. The views from which the input images are acquired are not calibrated. Therefore, an auto-calibration method for a sequence of images with small rotations and translation is developed. For this purpose, we have modified an already existing auto-calibration algorithm to incorporate known aspect ratio and skew values to make it applicable for small rotation around a single axis. Some experimental results of this auto-calibration method are demonstrated. The developed auto-calibration method is furthermore applied to three dimensional head modeling. Three dimensional positions of known facial features computed from two dimensional images are used to deform a generic head model by using a spring based energy minimization method. A sparse mesh with a few number of vertices is used in this study and therefore the approximated 3D model is not so exact. In addition, we have applied the deformation to only a limited number of feature points. We intend to use a more dense mesh and apply our deformation algorithm in a larger number of levels for a better 3D structure.

References


