A Robust Head Pose Estimation System for Uncalibrated Monocular Videos

B H Pawan Prasad*
Department of EE
IIT Madras
Chennai, India
mail3pawan@smail.iitm.ac.in

R Aravind
Department of EE
IIT Madras
Chennai, India
aravind@ee.iitm.ac.in

ABSTRACT

We present a robust head pose estimation system that is capable of estimating the 3D pose of a human head in video sequences captured using a single camera. The proposed system is able to accurately estimate the 3D pose parameters even without the knowledge of camera parameters. The face is modelled using a parametrized face mask in 3D. SIFT is used to match consecutive image frames. We propose a novel interpolation technique that captures the 3D movement of feature points to estimate the 2D−3D correspondences between the 3D model and the face image. The pose is established using the POSIT algorithm in a RANSAC framework that fits a 3D deformable face model onto the given face image. We evaluate the performance of the proposed scheme on standard test datasets. The mean absolute errors of estimated pitch, yaw and roll are found comparable and in some cases better than the results reported in literature.

Keywords

Candide, 2D−3D correspondences, RANSAC, POSIT

1. INTRODUCTION

Head pose estimation is defined as the process of determining the orientation and position of the human head with respect to the view of the camera. The human head can rotate with three degrees of freedom, namely yaw, pitch and roll. Head pose estimation can be performed with static images to estimate the pose on a coarse scale [1] or with video sequences by exploiting the temporal continuity to track the head in 3D and estimate pose on a fine scale [2]. Tracking the human head in 3D yields much more information compared to 2D tracking. Head tracking in 3D has multitude of applications such as human computer interaction, driver assistance systems, video conferencing, and many more. A recent survey of head pose estimation techniques is contained in [3].

*Corresponding author

ICVGIP '10, December 12-15, 2010, Chennai, India

The face is modelled as a cylindrical model in [4], a ellipsoidal model in [5]. It is represented as a sparse set of 3D points as in [6]. Pose estimation is then achieved by tracking feature points in video frames and establishing correspondence with the model points. Correspondence can be established either by deterministic methods where certain motion constraints are utilised [7], or by statistical methods which take into account the measurement and model uncertainties to estimate the object state [4], or by combining the two methods [8]. The Euler angles of yaw, pitch and roll are recovered using weak perspective geometry [9].

In this paper, we present a robust algorithm to perform 3D tracking of the human head in a video sequence captured using a monocular camera with unknown camera parameters. We use parametrized face mask Candide-3 to represent the face in 3D. We employ the Scale Invariant Feature Transform (SIFT) [10] to establish correspondence between facial feature points and model points. Estimation of pose from a set of 2D−3D points is achieved using the POSIT [11] algorithm in a RANSAC [12] framework that fits a 3D deformable model onto the given face image. Our overall solution includes methods for estimating the pose even when the number of SIFT matching points are low. This can occur due to various reasons such as fast motion, occlusion, etc.

The rest of the paper is organised as follows. Section 2 describes the concept of 3D pose estimation. Section 3 details the initialization of the tracking algorithm. Section 4 introduces the proposed head pose tracking algorithm applied to a video sequence. The experimental results that evaluate the performance of the proposed algorithm are presented in Section 5. Finally conclusions are presented in Section 6.

2. 3D POSE ESTIMATION

2.1 Head Modelling

Here we consider a 3D-based head model that aim at synthesizing head appearances. A human head can be modelled in 3D using a parametrized mask such as Candide-3 [13]. It consists of 113 vertices and 168 surfaces. The 3D model formed by connecting the model points is of standard size. Hence the model has to initialized by scaling it appropriately. The horizontal scaling factor h and the vertical scaling factor v are to be computed. The scaling factors h and v determine the amount by which the model has to be scaled in the x and the y dimensions respectively so as to fit the mesh to the given face. A 3D reference head model such as Candide-3 can be represented by a $N \times 3$ matrix G with each

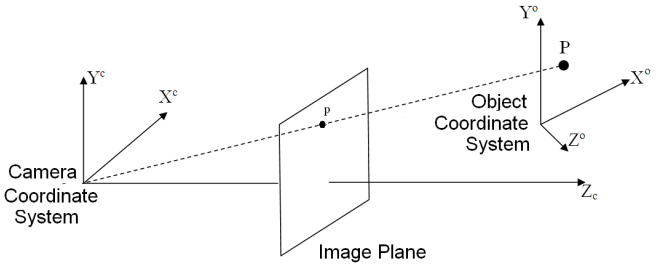


Figure 1: Transformation from Object coordinate system to Camera coordinate system

row representing a facial feature point $\{(X_i^o, Y_i^o, Z_i^o)\}_{i=1}^N$ on the face model.

2.2 Perspective Projection Model

The 3D coordinates of the Candide-3 face model can be represented in object coordinate system as $\mathbf{X_i^o} = (X_i^o, Y_i^o, Z_i^o)$. The camera coordinates are then obtained as $\mathbf{X}_i^c = R\mathbf{X}_i^o + \mathbf{t}$, where R is the rotation matrix and $\mathbf{t}$ is the translation vector. If f is the focal length of the camera, the image coordinates (x_i, y_i) of a point on the 3D reference model is obtained using perspective projection. Since we do not assume camera calibration, the camera focal length is unknown. If we assume that the depth variations in the object are small compared to its distance from the camera, the image coordinates is given by

$$x_i \approx \frac{fX_i^c}{Z^c} \quad , \quad y_i \approx \frac{fY_i^c}{Z^c} \tag{1}$$

Eqn. 1 represents the weak perspective model. Z^c is the distance between any one point on the face mesh and the camera origin. The estimation R and $\mathbf{t}$ using Eqn. 1 is robust to errors in the choice of the focal length f [9].

2.3 Pose Estimation

The goal of pose estimation is to determine the rotation matrix R and translation vector $\mathbf{t}$, which is accomplished using the POSIT algorithm [11]. Once we find the rotation matrix, the Euler angles θ_y, θ_r and θ_p can be computed.

We are given an object that is positioned in the field of view of the camera consisting of feature points $\mathbf{X}_0^o, \mathbf{X}_1^o, ..., \mathbf{X}_N^o$, each point with known object coordinates $\mathbf{X}_i^o = [X_i^o \;\; Y_i^o \;\; Z_i^o]^T$. The object here is the 3D reference model. We are also given a set of $M < N$ image points $\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_M$ with known image coordinates $\mathbf{x}_i = [x_i \;\; y_i]^T$. The image points are the feature points present on the face of a person in the given image. The correspondence between the 2D and 3D points are also given. The goal is to find the rotation matrix R and the translation vector $\mathbf{t}$. To compute R and $\mathbf{t}$, we use the POSIT (Pose from Orthography and Scaling with Iterations) algorithm which requires at least four 2D−3D correspondences. In our work, we always estimate the pose with six 2D−3D correspondences.

The Euler angles are computed using R and hence we can define the pose of any rigid 3D object with six degrees of freedom (three rotational and three translational) as

$$\mathbf{b} = [\theta_y \;\; \theta_r \;\; \theta_p \;\; t_x \;\; t_y \;\; t_z]^T \tag{2}$$

3. FACE MESH INITIALIZATION

In this section, the initialization of the head pose estimation system is described. For any tracking algorithm, the initialization step is very crucial. The performance of the tracking algorithm depends on the robustness of the initialization step. A very small variation in the estimation of the facial feature locations during initialization leads to quite large errors in the tracking process. We perform manual initialization where in the location of the facial feature points (eye and mouth corners) are manually marked. These feature points are then used to compute an initial pose.

To track the pose of the head of a person in the given video sequence, the generic 3D reference model has to be adapted to the person's face. The adapted face model denoted as $\{\mathbf{X}_i^f\}_{i=1}^N = (X_i^f, Y_i^f, Z_i^f)$ is to be computed. Face mesh initialization is usually carried out using the first frame of the video sequence which shows the person in a frontal view facing the camera. The frontal view is considered to be oriented with zero Euler angles with respect to the camera center. Unless the person maintains a frontal view, the HPE system is not initialized. The 3D reference model when projected onto the image plane shows a frontal view. The projected face mesh is of standard size. Hence it has to be scaled appropriately to adapt to a person's face.

The objective of the initialization step is to first compute the horizontal and the vertical scaling factors h and v, which determine the amount by which the face mesh has to be scaled in the x and the y dimensions respectively on the image plane. Given an image frame with the person's head in the frontal view, the two eye corners (m_1, n_1) and (m_2, n_2) and one mouth corner (m_3, n_3) are selected. The corresponding eye corners (x_1, y_1) and (x_2, y_2) and one mouth corner (x_3, y_3) on the projected face mesh are also selected. We then compute h and v as

$$h = \frac{|x_1 - x_2|}{|m_1 - m_2|} \quad , \quad v = \frac{|y_1 - y_3|}{|n_1 - n_3|} \tag{3}$$

We compute the camera coordinates $\{\mathbf{X}_i^c\}_{i=1}^N$ of the 3D reference model using $R = I$ and $\mathbf{t} = [0 \;\; 0 \;\; Z^c]^T$. Next, we project the reference model onto the image plane to get $\{(x_i, y_i)\}_{i=1}^N$ and scale them to get $(x_i^s, y_i^s) = (hx_i, vy_i)$. We then compute the horizontal and the vertical translations d_x and d_y on the image plane using the eye corner (x_1, y_1) needed to translate the face mesh to fit exactly onto the face. The translation in the horizontal direction is given by $d_x = x_1^s - m_1$ and the translation in the y direction is given by $d_y = y_1^s - n_1$. The coordinates of the adapted face model are then computed using inverse perspective projection as

$$\begin{aligned} X_i^f &= \left(\frac{Z^c + Z_i^o}{f}\right)(x_i^s - d_x) \\ Y_i^f &= \left(\frac{Z^c + Z_i^o}{f}\right)(y_i^s - d_y) \quad ; \quad i = 1, 2, ...N \\ Z_i^f &= Z_i^o \end{aligned} \tag{4}$$

Face mesh initialization is summarized in Algorithm 1. In Step. 1 of the algorithm, we use R as the identity matrix so that the reference model when projected onto the image plane shows a frontal view. Weak perspective projection in steps 3 and 8 are computed using $f = 1000$ and $Z^c = 50000$. The values of f and Z^c are determined during the calibration stage by experimental evaluations by performing several trial runs and comparing the results with the available

Algorithm 1: Face Mesh Initialization

Input: Eye and mouth corners on the image $\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3$, 3D reference model G
Output: $\mathbf{X}_i^f, \quad (x_i^1, \quad y_i^1), i = 1, 2, ...N$

Set $R = I$ and $\mathbf{t} = [0 \;\; 0 \;\; Z^c]^T$
$\mathbf{X}_i^c = R\mathbf{X}_i^o + \mathbf{t}$, $i = 1, 2, 3$
Compute $[x_i \;\; y_i]^T$ using Eqn. 1
Compute h and v using Eqn. 3
$[x_i^s \;\; y_i^s]^T = [hx_i \;\; vy_i]^T$, $i = 1, 2, ...N$
$d_x = x_1^s - m_1, \quad d_y = y_1^s - n_1$
Compute $\mathbf{X}_i^f = [X_i^f \;\; Y_i^f \;\; Z_i^f]^T$ using Eqn. 4
The adapted face mesh is given by $(x_i^1, \quad y_i^1) = (x_i^s - d_x, \quad y_i^s - d_y)$.

ground truth. The values of f and Z^c are only needed to be determined once in the calibration step. Once these values are set, we can use the same values for estimating head pose in any other video sequence.

4. HEAD POSE TRACKING IN A VIDEO SEQUENCE

We formulate the head tracking algorithm as the correspondence between facial feature points across frames. The feature points between two successive frames are detected and matched using the Scale Invariant Feature Transform (SIFT). Association of these feature points to the object model is based on the previous object state. The object state is represented by the vector $\mathbf{b}$ given in Eqn. 2. The goal of the tracking algorithm is to estimate the location of a set of facial feature points present in the incoming frame that correspond to the 3D reference head model. This is achieved by using the information available from the previous frame and the SIFT matching points.

The head pose tracking can be divided into three stages. The first stage involves extraction of SIFT matching points in two successive frames to represent the face. In the second stage, we establish the 2D−3D correspondences using these matching points and the previous object state $\mathbf{b^{n-1}}$. The final stage involves estimation of the current pose $\mathbf{b^n}$ at frame n using the set of 2D−3D correspondences computed in the previous stage.

4.1 Scale Invariant Feature Transform

Scale Invariant Feature Transform (SIFT) [10] is a method for extracting distinctive image features from images. These features are useful in performing reliable matching between different views of an object. These features are invariant to image scale and rotation, change in illumination, and also provide robust matching across a limited change in 3D viewpoint.

Given two successive frames of a video sequence $n-1$ and n, applying SIFT we obtain a set of P matching points $\{\mathbf{p}_i\}_{i=1}^P = [p_i^x \;\; p_i^y]^T$ and $\{\mathbf{q}_i\}_{i=1}^P = [q_i^x \;\; q_i^y]^T$ between frames $n-1$ and n respectively. We are interested in determining only the $M \leq P$ points $\{\mathbf{m}_i\}_{i=1}^M = [m_i^x \;\; m_i^y]^T$ and $\{\mathbf{n}_i\}_{i=1}^M = [n_i^x \;\; n_i^y]^T$ that are present inside the face region. For brevity, we denote the number of SIFT matching points $P^{n-1,n}$ between frames $n-1$ and n as P and the number of matching points present inside the face regions as M instead of $M^{n-1,n}$.

4.2 Determination of SIFT points inside face region

To determine the SIFT matching points between frames $n-1$ and n present inside the face region, we first need to compute the location of the face region in frame $n-1$ using the pose estimate $\mathbf{b}^{n-1}$. Then, the SIFT points $\{\mathbf{m}_i\}_{i=1}^M$ present inside the face regions are determined. Since there is a one-to-one correspondence between $\{\mathbf{m}_i\}_{i=1}^M$ and $\{\mathbf{n}_i\}_{i=1}^M$, the SIFT points of frame n are automatically determined.

Given the pose $\mathbf{b}^{n-1}$ of frame $n-1$, we compute the rotation matrix R^{n-1} and the translation vector $\mathbf{t}^{n-1}$. The adapted 3D model $\{\mathbf{X}_i^f\}_{i=1}^N$ obtained from mesh initialization of Section 3 is then rotated and translated to get the 3D coordinates of the model for the frame $n-1$.

$$\mathbf{X}_i^{n-1} = \begin{bmatrix} X_i^{n-1} \\ Y_i^{n-1} \\ Z_i^{n-1} \end{bmatrix} = R^{n-1} \begin{bmatrix} X_i^f \\ Y_i^f \\ Z_i^f \end{bmatrix} + \mathbf{t}^{n-1} \tag{5}$$

We then project these 3D points onto the image plane using Eqn. 1 to get $\{\mathbf{x}_i^{n-1}\}_{i=1}^N$. The SIFT matching points $\{\mathbf{m}_i\}_{i=1}^M$, $\{\mathbf{n}_i\}_{i=1}^M$ and the projected points $\{\mathbf{x}_i^{n-1}\}_{i=1}^N$ are now in the same 2D coordinate system. The projected points $\{\mathbf{x}_i^{n-1}\}_{i=1}^N$ form a set S. Given this finite non-empty set of N points on the image plane, the convex hull is represented by a sequence of $N_p \leq N$ vertices that form a polygon. The number of vertices of the polygon N_p depends on the scattering of the points $\mathbf{x}_i^{n-1}$. For an example of $N = 3$ non-collinear points, the polygon is a triangle with $N_p = 3$ vertices. The vertices of the convex hull are determined using the Matlab function *convhull*, which is based on the Quickhull algorithm [14]. We represent the polygon by a set of N_p vertices $\{\mathbf{c}_i\}_{i=1}^{N_p}$.

We can now determine the SIFT matching points $\{\mathbf{m}_i\}_{i=1}^M$ that are present inside the polygon region that represents the boundary of the convex hull. This is a fundamental Point-Location problem of computational geometry which has been solved in many ways [15]. We use the Matlab function *inpolygon* to determine if each of the P SIFT points lies inside the polygon. Extraction of SIFT matching points present inside the polygonal face region is summarized in Algorithm 2.

Algorithm 2: Extraction of SIFT matching points present inside the polygonal face region

Input: $\mathbf{X}_i^f$, $\{\mathbf{p}_i\}_{i=1}^P$, $\{\mathbf{q}_i\}_{i=1}^P$, and $\mathbf{b}_{n-1}$
Output: $\{\mathbf{m}_i\}_{i=1}^M$, $\{\mathbf{n}_i\}_{i=1}^M$

Compute $\{\mathbf{X}_i^{n-1}\}_{i=1}^N$ using Eqn. 5 ;
Project $\{\mathbf{X}_i^{n-1}\}_{i=1}^N$ onto the image plane using Eqn. 1 to get $\{\mathbf{x}_i^{n-1}\}_{i=1}^N$;
Determine the vertices $\{\mathbf{c}_i\}_{i=1}^{N_p}$ of the convex hull ;
$k = 1$;
for $i \leftarrow 1$ **to** P **do**
    if $[\mathbf{p}_i$ *is inside the polygon* $\{\mathbf{c_j}\}_{j=1}^{N_p}]$ **then**
        $\mathbf{m}_k = \mathbf{p}_i$; $\mathbf{n}_k = \mathbf{q}_i$; $k = k + 1$;
    end if
end for

4.3 Estimation of 2D-3D correspondence

The SIFT matching points $\{\mathbf{m}_i\}_{i=1}^M$, $\{\mathbf{n}_i\}_{i=1}^M$ present inside the face region may or may not correspond to the 3D points of the face model. Hence the goal of this section

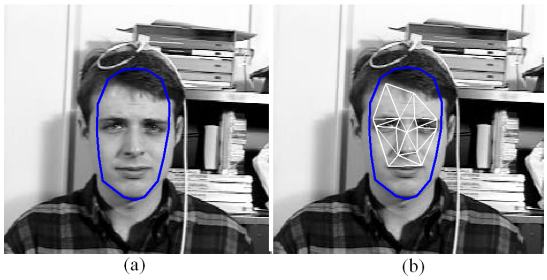
is to determine the location of the mesh points $\{\mathbf{x}\}_{i=1}^{N}$ at frame n that correspond to the 3D points on the face model. However, normalised cross correlation as in [4] or nearest neighbour translation as in [8] would have been possible if a cylindrical face model with densely lying points was used. Since we use the Candide-3 face model, both these methods leads to accumulation of errors. To achieve this, we propose a strategy to establish a set of 2D-3D point correspondences using the pose $\mathbf{b}^{n-1}$ of the frame $n-1$ and the SIFT matching points. Using this set of correspondences, the pose of the current frame n can be computed using the POSIT algorithm as discussed in Section 2.3. The motivation for computing the 3D points corresponding to mesh 2D points and not the SIFT matching points is that, we know the 3D values of the mesh points from the face model, but we do not know the 3D points corresponding to the SIFT matching points $\{\mathbf{m}_i\}_{i=1}^{M}$. The other important motivation is that, when the number of SIFT matches are small, computing the pose from a small set of matching points containing outliers leads to errors. This in turn leads to accumulation of errors in subsequent frames. The number of SIFT matching points can be low under the circumstances such as fast motion, occlusion, face being only partially visible due to large yaw angle, etc.

The boundary points $\{\mathbf{c}_i\}_{i=1}^{N_p}$ of the convex hull of a set of mesh points $\{\mathbf{x}_i^{n-1}\}_{i=1}^{N}$ that form a polygon with N_p vertices encloses the M SIFT points $\{\mathbf{m}_i\}_{i=1}^{M}$. The points present on the boundary of the polygon are also considered enclosed. We then determine the vertices of the new convex hull formed using these M SIFT points at frame $n-1$. Hence, K out the N mesh points are enclosed inside this new convex hull. Let us denote this subset of K mesh points as $\{\mathbf{u}_i\}_{i=1}^{K}$. The corresponding subset of 3D points from the adapted face model $\{\mathbf{X}_i^f\}_{i=1}^{N}$ is then denoted as $\{\mathbf{U}_i^f\}_{i=1}^{K}$.

The K mesh points $\{\mathbf{u}_i\}_{i=1}^{K}$ of frame $n-1$ have one-to-one correspondence with K points in frame n. We denote this set of K points at frame n as $\{\mathbf{v}_i\}_{i=1}^{K}$. The locations of these points are unknown because, the pose at frame n is unknown. In this section, we develop an algorithm to estimate the location of these K points at frame n using the K mesh points at frame $n-1$ and SIFT matching points.

Now let us consider a mesh point $\mathbf{u}_i$ present inside the new convex hull formed by the points $\{\mathbf{m}_i\}_{i=1}^{M}$. This point can be considered to be enclosed by a triangle formed by three vertices from the set $\{\mathbf{m}_i\}_{i=1}^{M}$. In order to estimate the location of $\mathbf{v}_i$, we first compute the barycentric coordinates of the mesh point $\mathbf{u}_i$ of frame $n-1$. Since computation of barycentric coordinates of any point enclosed inside a triangle is easier compared to any other polygon, we use Delaunay triangulation [16] to connect the M SIFT points as shown in Fig. 2(b). Delaunay triangulation of a set of M points on a plane is a triangulation such that none of the M points are present inside the circumcircle of any triangle in the triangulation and none of the edges intersect. We use the Matlab function *delaunay* that returns a matrix S of size $J \times 3$, where each row of the matrix S contain the indices of the SIFT points $\{\mathbf{m}_i\}_{i=1}^{M}$ that represent each of the J triangles. The number of triangles J in frame $n-1$ depend on the scattering of the M SIFT points $\{\mathbf{m}_i\}_{i=1}^{M}$ on the image plane.

Each of the K mesh points $\{\mathbf{u}_i\}_{i=1}^{K}$ of frame $n-1$ is typically present inside only one of the J triangles. Hence we need to first determine the triangles in which each of these K mesh points is present. This is achieved in a similar way as discussed earlier, where each point $\mathbf{u}_i$ is tested against all the triangles one by one using the Matlab function *inpolygon*. Now that we know the vertices of the triangle inside which each mesh point is located, we can compute the barycentric coordinates [17] of these mesh points $\mathbf{u}_i$ as follows.



(a) (b)

Figure 2: Estimation of 2D - 3D correspondence (a) Polygonal face region of frame $n-1$ (b) Delaunay Triangulation of the SIFT matching points $\{\mathbf{m}_i\}_{i=1}^{M}$ of frame $n-1$

Let us denote the barycentric coordinates of a mesh point $\mathbf{u}_i = (u_i^x,\ u_i^y)$ as $\Lambda_i = [\lambda_{i,1}\ \ \lambda_{i,2}]^T$. We obtain the vertices of the triangle inside which $\mathbf{u}_i$ lies as say, $\mathbf{r}_i = [r_i^x\ \ r_i^y]^T$, $\mathbf{s}_i = [s_i^x\ \ s_i^y]^T$ and $\mathbf{t}_i = [t_i^x\ \ t_i^y]^T$ from the matrix S. Hence $\mathbf{u}_i$ can be written as a weighted sum of these three vertices [17] as

$$\begin{aligned}\begin{bmatrix} u_i^x \\ u_i^y \end{bmatrix} &= \lambda_{i,1}\begin{bmatrix} r_i^x \\ r_i^y \end{bmatrix} + \lambda_{i,2}\begin{bmatrix} s_i^x \\ s_i^y \end{bmatrix} \\ &\quad + \ (1-\lambda_{i,1}-\lambda_{i,2})\begin{bmatrix} t_i^x \\ t_i^y \end{bmatrix} \\ &= \begin{bmatrix} r_i^x - t_i^x & s_i^x - t_i^x \\ r_i^y - t_i^y & s_i^y - t_i^y \end{bmatrix}\begin{bmatrix} \lambda_{i,1} \\ \lambda_{i,2} \end{bmatrix} + \begin{bmatrix} t_i^x \\ t_i^y \end{bmatrix} \quad (6)\end{aligned}$$

Let us define a matrix $\mathcal{A}_i$ as follows.

$$\mathcal{A}_i = \begin{bmatrix} r_i^x - t_i^x & s_i^x - t_i^x \\ r_i^y - t_i^y & s_i^y - t_i^y \end{bmatrix} \qquad 1 \le i \le K \quad (7)$$

The matrix $\mathcal{A}_i$ is always invertible because, if it were not to be invertible, the three vertices will be collinear and does not form a triangle [17]. Substituting Eqn. 7 in Eqn. 6, we get

$$\begin{aligned}\mathbf{u}_i &= \mathcal{A}_i\Lambda_i + \mathbf{t}_i \qquad 1 \le i \le K \quad (8)\\ \Rightarrow \Lambda_i &= (\mathcal{A}_i)^{-1}(\mathbf{u}_i - \mathbf{t}_i) \quad (9)\end{aligned}$$

We estimate the location of the mesh points $\{\mathbf{v}_i\}_{i=1}^{K}$ in frame n as follows. Let us denote the three points at frame n that have SIFT correspondence to the triangle vertices $\mathbf{r}_i$, $\mathbf{s}_i$ and $\mathbf{t}_i$ as $\mathbf{w}_i$, $\mathbf{y}_i$ and $\mathbf{z}_i$ respectively. Here we assume that $\mathbf{v}_i$ lies inside the triangle formed by the three vertices $\mathbf{w}_i$, $\mathbf{y}_i$ and $\mathbf{z}_i$. Hence we can write $\mathbf{v}_i$ as a weighted sum of these three vertices as before. To estimate $\mathbf{v}_i$ at frame n, we use the same barycentric coordinates Λ_i of $\mathbf{u}_i$ computed at frame $n-1$. Therefore,

$$\begin{aligned}\mathbf{v}_i &= \mathcal{B}_i\Lambda_i + \mathbf{z}_i \qquad 1 \le i \le K \\ &= \mathcal{B}_i(\mathcal{A}_i)^{-1}(\mathbf{u}_i - \mathbf{t}_i) + \mathbf{z}_i \quad (10)\end{aligned}$$

where $\mathcal{B}_i$ is computed as in Eqn. 7 using the vertices $\mathbf{w}_i$, $\mathbf{y}_i$ and $\mathbf{z}_i$ in place of $\mathbf{r}_i$, $\mathbf{s}_i$ and $\mathbf{t}_i$. Eqn. 10 holds true

if the following three constraints are satisfied. Firstly,the SIFT matching points are accurate. Secondly, the face does not undergo any local deformation caused due to facial appearance changes. Thirdly, the head pose at frame $n-1$ is precisely known.

Any one of the above constraints not holding true, leads to the occurrence of outliers. Handling of outliers is discussed in the next section. Estimation of 2D−3D correspondences is summarized in Algorithm 3. This set of 2D−3D correspondences $\{\mathbf{v}_i\}_{i=1}^{K}$ and $\{\mathbf{U}_i^f\}_{i=1}^{K}$ are used to determine the pose at frame n as described in the next section.

Algorithm 3: Estimation of 2D - 3D correspondences at frame n

Input: $[X_i^f \;\; Y_i^f \;\; Z_i^f]^T$, $\{\mathbf{m}_i\}_{i=1}^{M}$, $\{\mathbf{n}_i\}_{i=1}^{M}$
Output: $\{\mathbf{v}_i\}_{i=1}^{K}$, $\{\mathbf{U}_i^f\}_{i=1}^{K} = [U_i^f \;\; V_i^f \;\; W_i^f]^T$

Determine the vertices $\{\mathbf{e}_i\}_{i=1}^{M_p}$ of the convex hull using $\{\mathbf{m}_i\}_{i=1}^{M}$;
Compute the delaunay triangulation matrix T of size $J \times 3$ using $\{\mathbf{m}_i\}_{i=1}^{M}$;
$k = 1$;
for $i \leftarrow 1$ **to** N **do**
  if $\left[\mathbf{x}_i^{n-1}$ *is inside the new convex hull* $\right]$ **then**
    $\mathbf{u}_k = \mathbf{x}_i^{n-1}$; $\mathbf{U}_k^f = \mathbf{X}_i^f$; $k = k+1$;
  end if
end for
$K = k-1$;
$k = 1$;
for $i \leftarrow 1$ **to** K **do**
  for $j \leftarrow 1$ **to** J **do**
    $a = S(j,1)$; $b = S(j,2)$; $c = S(j,3)$;
    if $\left[\mathbf{u}_i$ *is inside triangle with vertices* $\mathbf{r}_j$, $\mathbf{s}_j$, $\mathbf{t}_j\right]$ **then**
      Compute $\mathcal{A}_i$ using vertices $\mathbf{r}_j$, $\mathbf{s}_j$, $\mathbf{t}_j$;
      $\Lambda_i = (\mathcal{A}_i)^{-1}(\mathbf{u}_i - \mathbf{t}_j)$;
      Compute $\mathcal{B}_i$ using vertices $\mathbf{w}_j$, $\mathbf{y}_j$, $\mathbf{z}_j$;
      $\mathbf{v}_k = \mathcal{B}_i\Lambda_i + \mathbf{z}_j$; $k = k+1$;
    end if
  end for
end for

4.4 Pose Estimation from 2D-3D Correspondence

As described in Section 2.3 we use the POSIT algorithm to estimate the 3D pose. The 2D−3D point correspondences are established as described in Section 4.3. The number of point correspondences was found to be in the range of 50 to 80 in our experiments. This large set of point correspondences can have outliers resulting from erroneous measurements, noise, etc. Motivated by the presence of this large number of point correspondences, we use the Random Sampling and Consensus (RANSAC) [12] framework to eliminate outliers. However, if original RANSAC is used, there is a strong possibility of error accumulation from frame to frame which will lead to tracking failure. Hence we make use of the adaptation strategy introduced in [18]. The RANSAC based pose estimation involves four stages namely, random sampling, model estimation, texture consistency and consensus.

4.4.1 Random Sampling

Let us denote the number of samples that we choose from the given set of K point correspondences $\{\mathbf{v}_i\}_{i=1}^{K}$ and $\{\mathbf{U}_i^f\}_{i=1}^{K}$ as α. To determine the pose $\mathbf{b}^{(n)}$ at frame n using the POSIT algorithm, we take $\alpha = 6$.

4.4.2 Model Estimation using POSIT

We make use of the POSIT algorithm to estimate the rotation matrix $R^{(n)}$ and translation vector $\mathbf{t}^{(n)}$ at frame n using which we can compute the pose parameters $\mathbf{b}^{(n)}$.

$$\mathbf{b}^{(n)} = \text{POSIT}\left[\{\mathbf{v}_i\}_{i=1}^{\alpha}, \{\mathbf{U}_i^f\}_{i=1}^{\alpha}\right] \tag{11}$$

4.4.3 Texture Consistency

Given a lexicographically ordered face image $\boldsymbol{\mu}$ at frame n, let us denote the geometrically normalized face image [19] as $\boldsymbol{\rho}$ and is given by

$$\boldsymbol{\rho} = \boldsymbol{\rho}(\mathbf{b}^{(n)}) = \mathcal{W}(\boldsymbol{\mu}, \mathbf{b}^{(n)}) \tag{12}$$

where $\mathcal{W}$ is the piecewise affine transform [20]. The piecewise affine transforms perform pose normalization for the given face. If the pose parameters $\mathbf{b}^{(n)}$ is a good fit to the given image $\boldsymbol{\mu}$, then the geometrically normalized image $\boldsymbol{\rho}$ will closely resemble a face. Hence $\boldsymbol{\mu}$ will be consistent with the face statistical model. The residual error between the $\boldsymbol{\rho}$ and its projection $\widehat{\boldsymbol{\rho}}$ onto the PCA subspace will be small. The PCA subspace is formed using the standard BioID face database [21] which contains a large variety of face images taken under varying illumination, background and face size. We choose a set of 500 training face images from the database. The face is cropped and resized to 40×40. Applying PCA on this dataset, we obtain the mean texture $\bar{\boldsymbol{\rho}}$ and a set of eigenvectors. We choose the 100 principal eigenvectors and denote the eigenvector matrix of size 1600×100 as E. The projection of $\boldsymbol{\rho}$ onto this PCA subspace is given by

$$\widehat{\boldsymbol{\rho}}(\mathbf{b}^{(n)}) = \bar{\boldsymbol{\rho}} + EE^T(\boldsymbol{\rho} - \bar{\boldsymbol{\rho}}) \tag{13}$$

Hence the measure of a good fit can be estimated by the reconstruction error given by

$$e(\mathbf{b}^{(n)}) = \left|\left|\boldsymbol{\rho}(\mathbf{b}^{(n)}) - \widehat{\boldsymbol{\rho}}(\mathbf{b}^{(n)})\right|\right|^2 \tag{14}$$

4.4.4 Consensus

The above three steps are repeated L times. The pose parameters $\mathbf{b}^{(n)}$ with the least reconstruction error is taken as the final estimate of the pose parameters.

4.4.5 Determination of number of iterations

Since RANSAC involves random sampling, it is unnecessary and computationally infeasible to use every possible sample to estimate the model [22]. Instead we choose the number of samples L to ensure that the probability p that at least one of the random samples α is free from outliers. Let us denote the probability that any selected data point is an inlier as w. Hence the probability that all the α points are inliers is w^α. Therefore $1 - w^\alpha$ is the probability that atleast one of α points is an outlier. This probability to the power L is the probability that the none of the L samples have all inlier points which is same as $1 - p$. Hence we can write $1 - p = (1 - w^\alpha)^L$. Therefore the number of samples

L is given by

$$L = \left\lceil \frac{\log(1-p)}{\log(1-w^{\alpha})} \right\rceil \tag{15}$$

In our work, we set p as 0.95. The percentage of inliers in the data set w for the next frame $n+1$ is approximately found using the pose parameters $\mathbf{b}^{(n)}$ of the current frame as follows. The threshold ϵ we have used is 0.75. We classify the points $\{\mathbf{v}_i\}$ as an inlier if the following condition is satisfied.

$$\left\| \mathbf{v}_i - \frac{f}{Z^c} \left(R^{(n)} \mathbf{U}_i^f + \mathbf{t}^{(n)} \right) \right\| < \epsilon \qquad 1 \le i \le K \tag{16}$$

The event of tracking failure has to be taken into account in any tracking algorithm. Here, we use different criteria to detect it. We first determine the number of SIFT matching points M between the current frame and the previous frame. If $M < 3$, we retain the pose of the previous frame and go ahead with tracking the head in the next frame. This is a natural choice because, we need a minimum of three points to form a triangle using which the mesh points $\{\mathbf{u}_i\}_{i=1}^{K}$ are determined.

Secondly, we determine the change in Euler angles between the current frame and the previous frame. Once these values are larger than the specified threshold of ten degrees, we determine the reconstruction error using Eqn. 14 and verify whether this value is larger than some threshold value. This threshold value is computed using the reconstruction error of the first frame from the initialization step. Once the reconstruction error is greater than this specified threshold, the tracking is stopped and the initialization is started. Since, we rely on manual initialization, the tracking is again restarted only when the face shows a frontal view.

5. EXPERIMENTAL RESULTS

To evaluate the performance of the proposed head pose estimation system, we use the Boston University database [23]. It consists of 72 image sequences of 200 frames each of size 320×240, that contains eight people, each of them appearing in nine videos. Out of the eight sets of nine videos each, five sets were taken under uniform illumination and the rest were taken under varying illumination. The ground truth indicating the Euler angles is available for all the 72 sequences. We have tested the proposed algorithm on all the 72 video sequences to evaluate its robustness under uniform as well as varying illumination.

We test our algorithm first on uniform illumination. Fig. 3 shows the face mesh adaptation for the sequence "*Llm1*" using the proposed algorithm. The initialization is carried out using the first frame of the video sequence in which the face shows a frontal view. The tracker is able to successfully estimate the pose of the face in all the frame of the sequeuce "*LLm1*" as shown in Fig. 3. Next we consider the sequence "*Vam5*" as shown in Fig 4. In this sequence the face undergoes large roll rotations and scale variations, and the tracker is able to handle them effectively.

The algorithm is next evaluated for its robustness under varying illumination. Here we consider the sequence "*Jal6*" as shown in Fig. 5, where the tracker is able to successfully track the face under varying illumination. Fig. 6 shows the plot of number of iterations $L^{(n)}$ versus the frame number n. The number of iterations for some frames is very high, computation of which is very time consuming. Hence in such

Figure 3: "*Llm1*" sequence, frames 1, 45, 147, 184

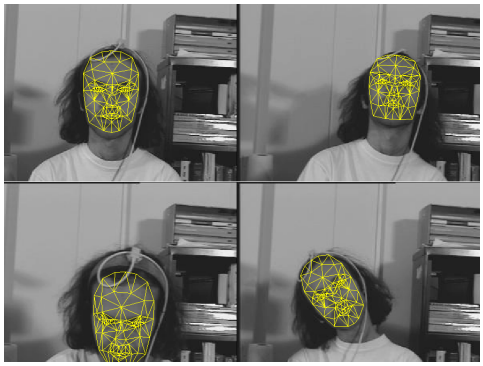


Figure 4: "*Vam5*" sequence, frames 1, 67, 104, 150

Figure 5: "*Jal6*" sequence, frames 1, 39, 91, 160

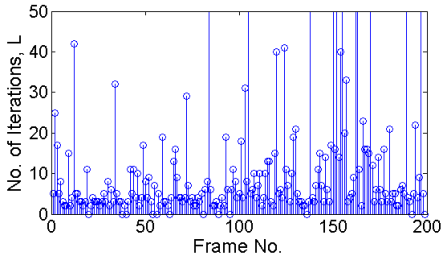


Figure 6: No. of Iterations of model fitting performed on each frame for the sequence "*Vam5*"

Figure 7: "*Dudek*" sequence, frames 1, 174, 213, 270

cases, we restrict the number of iterations and determine the pose from the best fitted model.

As a final example, we consider the "*Dudek*" sequence [24] where there is significant movement of the head as well as the camera along with self occlusions. Although the ground truth is not available for this dataset, we have verified the robustness visually and the tracker output is shown in Fig. 7. The tracker almost loses track at frame 213 because of self occlusion. But the adaptation step prevents error accumulation and restores the tracker at frame 270.

We finally illustrate the robustness of the algorithm in estimating the pose using a small set of SIFT matching points. Fig. 8 shows a plot of number of SIFT matching points and mesh points between frames n and $n-1$ for the *Dudek* sequence. The number of selected mesh points are always much higher compared to the number SIFT points. In some cases, the number of SIFT matching points are as low as three. Since we compute the pose from the mesh points and not from the SIFT points, our algorithm is able to handle such scenarios effectively.

The estimated Euler angles against the ground truth values [23] is shown for four different video sequences in Fig. 9. We then compute the mean absolute error of the Euler angles from the ground truth for all the sequences. The average MAE values for the three Euler angles are tabulated in Table 1. It is evident that, our method is comparable and in some cases perform even better. The performance of the head pose estimation system can be verified from the videos uploaded as supplementary material.

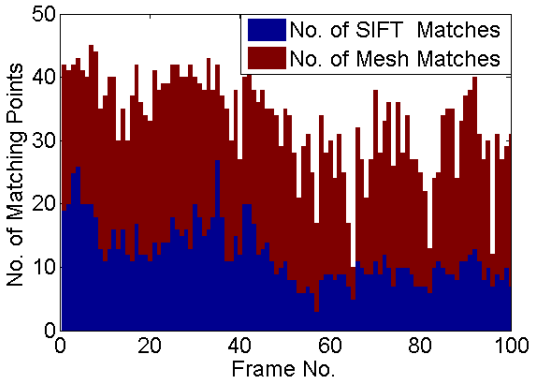


Figure 8: No. of SIFT points and Mesh points between frames n and $n-1$ for *Dudek* sequence

6. CONCLUSIONS

In this paper, we have presented a complete head pose estimate system that is capable of recovering the pose parameters of yaw, pitch and roll of a human head captured using a single camera. The pose parameters were estimated without assuming the knowledge of internal camera parameters. We have also presented a novel algorithm to estimate the 2D−3D correspondences using Delaunay interpolation. The system was tested for its robustness on standard data sets that involved large movements in all the six degrees of freedom. The algorithm was also evaluated on challenging conditions such as varying illumination and camera movement. The tracker was also found to be robust to moderate self occlusions. In this work, the facemesh is manually initialised. Future work will investigate employing automatic initialization to the first frame of the video sequence. This can be accomplished by using a face and feature point detector. The proposed system uses SIFT to determine matching points between consecutive frames. However, SIFT can be replaced by faster algorithms such as SURF to make the algorithm work in real time.

Table 1: MAE of different algorithms

Algorithm	Mean Absolute Error (deg)		
	Pitch	Yaw	Roll
Method in [23]	3.3	6.1	9.8
Method in [25]	3.8	3.2	1.4
Method in [4]	3.7	4.6	2.1
Method in [5]	3.92	4.04	6.71
Our Method	**2.5**	**3.8**	**3.6**

7. REFERENCES

[1] B. Ma, W. Zhang, S. Shan, X. Chen, and W. Gao, "Robust head pose estimation using LGBP," *Pattern Recognition*, vol. 2, pp. 512–515, 2006.

[2] T. Gritti, "Toward fully automated face pose estimation," in *Proceedings of the International workshop on Interactive multimedia for Consumer Electronics,2009.*

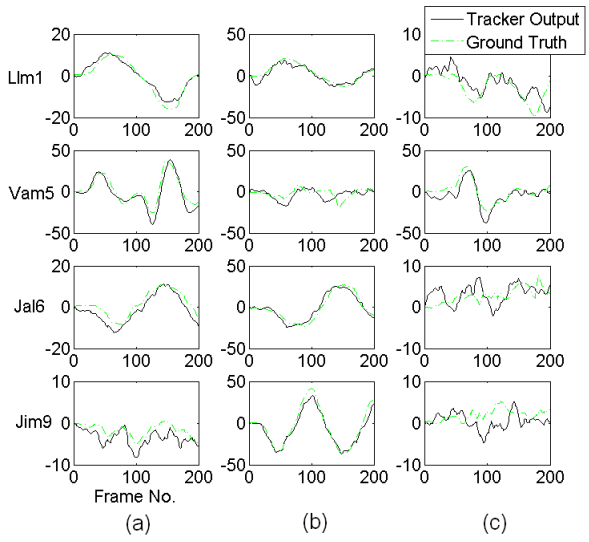


Figure 9: Rotation angles for four Boston University data sequences (a) Pitch, (b) Yaw, (c) Roll (deg)

[3] E. Murphy-Chutorian and M. Trivedi, "Head pose estimation in computer vision: A survey," *IEEE Transactions on PAMI*, pp. 607–626, 2008.

[4] J. Jang and T. Kanade, "Robust 3D head tracking by online feature registration," in *8th IEEE International Conference on Automatic Face and Gesture Recognition*, 2008.

[5] S. Choi and D. Kim, "Robust head tracking using 3D ellipsoidal head model in particle filter," *Pattern Recognition*, vol. 41, no. 9, pp. 2901–2915, 2008.

[6] S. Ohayon and E. Rivlin, "Robust 3d head tracking using camera pose estimation," in *International Conference on Pattern Recognition*, vol. 1, 2006.

[7] R. Ruddarraju, A. Haro, and I. Essa, "Fast multiple camera head pose tracking," *Proceedings, Vision Interface*, 2003.

[8] T. Brox, B. Rosenhahn, J. Gall, and D. Cremers, "Combined region and motion-based 3D tracking of rigid and articulated objects.," *IEEE Transactions on PAMI*, vol. 32, no. 3, p. 402, 2010.

[9] G. Aggarwal, A. Veeraraghavan, and R. Chellappa, "3d Facial pose tracking in Uncalibrated videos," *Pattern Recognition and Machine Intelligence*, pp. 515–520, 2005.

[10] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[11] D. DeMenthon and L. Davis, "Model-based object pose in 25 lines of code," *International Journal of Computer Vision*, vol. 15, no. 1, pp. 123–141, 1995.

[12] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[13] J. Ahlberg, "Candide-3–an updated parametrized face," *Report No. LiTH-ISY*, 2001.

[14] C. Barber, D. Dobkin, and H. Huhdanpaa, "The Quickhull algorithm for convex hulls," *ACM Transactions on Mathematical Software (TOMS)*, vol. 22, no. 4, pp. 469–483, 1996.

[15] M. De Berg, O. Cheong, M. Van Kreveld, and M. Overmars, *Computational geometry: Algorithms and applications*. Springer, 2008.

[16] H. Edelsbrunner, *Geometry and topology for mesh generation*. Cambridge Univ. Press, 2001.

[17] C. Bradley, "The Algebra of Geometry: Cartesian, Areal and Projective Co-ordinates," *Highperception Ltd., Bath*, 2007.

[18] F. Dornaika and J. Ahlberg, "Face and facial feature tracking using deformable models," *International Journal of Image and Graphics*, vol. 4, no. 3, p. 499, 2004.

[19] F. Dornaika and J. Ahlberg, "Fitting 3D face models for tracking and active appearance model training," *Image and Vision Computing*, vol. 24, no. 9, pp. 1010–1024, 2006.

[20] I. Matthews and S. Baker, "Active appearance models revisited," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 135–164, 2004.

[21] R. F. O. Jesorsky, K. Kirchberg, "Audio and Video based Person Authentication - AVBPA," *IEEE Transactions on PAMI*, 2001.

[22] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge University Press New York, NY, USA, 2003.

[23] M. La Cascia, S. Sclaroff, and V. Athitsos, "Fast, reliable head tracking under varying illumination: an approach based on registration of texture-mapped 3 D models," *IEEE Transactions on PAMI*, vol. 22, no. 4, pp. 322–336, 2000.

[24] A. Jepson, D. Fleet, and T. El-Maraghi, "Robust online appearance models for visual tracking," *IEEE Transactions on PAMI*, pp. 1296–1311, 2003.

[25] J. Xiao, T. Moriyama, T. Kanade, and J. Cohn, "Robust full-motion recovery of head by dynamic templates and re-registration techniques," *International Journal of Imaging Systems and Technology*, vol. 13, no. 1, pp. 85–94, 2003.