Head Tracking with 3D Texture Map Model in Planning Meeting Analysis [*]

Yingen Xiong
Center for Human Computer Interaction
Virginia Polytechnic Institute and State University
621 McBryde Hall, MC0106
Blacksburg,VA 24061, USA
yxiong@cs.vt.edu

Francis Quek
Center for Human Computer Interaction
Virginia Polytechnic Institute and State University
618 McBryde Hall, MC0106
Blacksburg,VA 24061, USA
quek@cs.vt.edu

ABSTRACT

In order to realize automatic detection, recognition, annotation and understanding of video events associated with formal and informal meetings, we need to find first where the participants of interest are and extract their behavior in the meetings. This requires robust and efficient human body, head, face, and hand tracking algorithms in a clustered background. In this paper, we address the tracking of participants' heads and faces. This is important to estimate the gaze orientations of meeting participants. The feature-based algorithm does not work well because of the low resolution of meeting video. We propose a simple and efficient approach to track human head from the videos of formal and informal meetings. In this approach, we build a head 3D texture map model dynamically on the fly with the incoming image stream and use this model to obtain head orientations and positions. The proposed approach can be divided into four parts. The first part is face detection with color model. We build human skin color model with Gaussian distribution. With this model, we perform face detection and obtain face region for each frame of incoming image stream. The second part is face image registration. We propose a voting procedure to get rotation angle of the face in 2D image plane. With the rotation angles, we can rectify each frame in the same orientation. The next part is head modelling. We model the human head as an ellipsoid with 360^o wide image. We build the head 3D texture map model dynamically with the incoming image stream. The head model is in the head orientation (θ, φ) space. Finally, we use the 3D texture map model to perform head tracking and obtain head position and orientation for each frame of incoming image stream. In the mean time, we use the tracking results to modify the 3D texture map model, so that we can obtain better results.

Categories and Subject Descriptors

I.4.9 [**Image Processing and Computer Vision**]: Applications; H.5.m [**Information Interfaces and Presentation**]: Miscellaneous

General Terms

Algorithms, Measurement, Experimentation, Human Factors, Languages

Keywords

Multi-model meeting analysis, meeting events, video analysis, head tracking, face tracking, head modeling, head image registration, head 3D texture map model

1. INTRODUCTION

In video-based multi-modal meeting analysis, meeting events are recorded by cameras. Understanding of meeting behavior in videos must necessary combine the psycholinguistics of multi-modal human language, signal and language processing, and computer vision. Meetings are gatherings of humans for purpose of communication, collaboration, confrontation, etc. In our research for multi-modal analysis of planning meetings, we proposed a cross-disciplinary effort dedicated to the analysis of planning meetings spanning the entire research arc from data acquisition through coding and annotation of the multi-modal communication. In order to realize automatic detection, recognition, annotation and understanding of video events associated with formal and informal meetings, we need to find first where the objects of interest i.e. the participants, are and to extract the participants' behavior in multi-modal communication. This requires robust and efficient human body, head, face, and hand tracking algorithms in a clustered background.

Human gesture, speech, and gaze function as an integrated whole, no part of which may be fully understood in isolation [1]. Since they are not subservient to each other, but spring from a 'deeper' semantic source in the human mind, they provide a glimpse into the function of the mind in the production of human communication. In order to understand meeting events, we must understand human multi-modal communication behavior, study human gesture, speech, and gaze, and apply them to the analysis of multi-modal communication. The tracking of participants'

[*] This research has been supported by the Advanced Research and Development Activity ARDA VACEII grant 665661: From Video to Information: Cross-Model Analysis of Planning Meetings.

ICMI'05, October 04-06, 2005, Trento, Italy.

head and face is important to estimate the gaze orientations of meeting participants.

Head tracking is a well-researched area; various approaches have been developed based on head model [2, 3, 4, 5, 6, 7], facial features [8, 9, 10, 11], support vector machines [12] optical flow [4], and Kalman filtering [13, 14, 15]. However, trying to track human head in a low resolution meeting video is a real challenging task. Besides, the head tracking algorithms must allow great change in orientation. In this situation, the feature-based algorithm does not work well, because we can not track these features in each frame for the low resolution video. Appearance-based algorithms do not need to track features on the images but have low tracking accuracy. Model-based algorithms have higher tracking accuracy but usually have higher complexity. In our case for the analysis of meeting events, we hope get higher tracking accuracy from the low resolution meeting videos. In our work, we combine appearance-based and model-based techniques to develop a simple and efficient head tracking algorithm. This algorithm does not need to track features and allows great change in orientation. It has four steps including face region segmentation, face image rectification, head 3D texture map model developing, and head tracking with the 3D texture map model. In the first step, we develop a Gaussian skin color model with skin color samples from the face images and apply this model to segment the face region for each frame of the videos. In the second step, we propose a voting procedure to obtain the rotation angle for the current frame. With this rotation angle, we can rectify the face image to the same orientation of the previous frames. In the next step, we model the human head as an ellipsoid with a 360^o wide image. We project face images onto the ellipsoidal surface to build a 3D texture map model in (θ, φ) space. In the final step, we register each frame of the incoming image stream onto the 3D texture map model to obtain positions and orientations of the head, so that we can track the head in the videos.

Here is the organization of the paper. In section 2, we summarize our approach. The face detect with Gaussian skin color model is described in section 3 and the face image registration and rectification are given in section 4. Head 3D texture map modeling with incoming image stream and head tracking using the 3D texture map model are discussed in section 5 and 6, followed by a summary of the paper in section 7.

2. SUMMARY OF OUR APPROACH

Figure 1 shows the whole approach for the head tracking with 3D texture map model. The approach can be divided by four parts including face detection, face image registration, head 3D texture map model developing, and head tracking with the 3D texture map model. The first part is face detection for the incoming image stream. In this part, we build skin color model with Gaussian distribution. With this Gaussian skin color model, we segment face region for each frame of the video, so we can obtain face image stream. The second part is the head image registration. In this part, we apply harris corner detector to detect corner points in face region. With these corner points, we register the current face image with the previous frame of the video, so that we can get the rotation angle between these two face images. Through the registration, we can obtain the same orientation face images. The next part is building head 3D texture map model. In this part, we model the human head as an ellipsoid with 360° wide image. We project and warp the face images obtained from face detection onto the ellipsoidal surface. We expand the ellipsoidal surface on (θ, φ) plane and we can compute a texture map for each face. By registering all texture maps of the face image stream onto the $(\theta,\ \varphi)$ plane, we can obtain head 3D texture map model. The final part is the head tracking with the head 3D texture model. In this part we compute a texture map for each face image and register this texture map onto the 3D texture map model. We can obtain the head orientation of the current frame. Combining with the 2D face detection process and camera model, we can also obtain the head position in each frame of the incoming image stream. In the meantime, we also use the tracking results to update our 3D texture map model. During the tracking process, we build the head 3D texture map model dynamically on the fly using incoming image stream and the model will be updated continuously. With more accurate model, we can obtain higher accurate results.

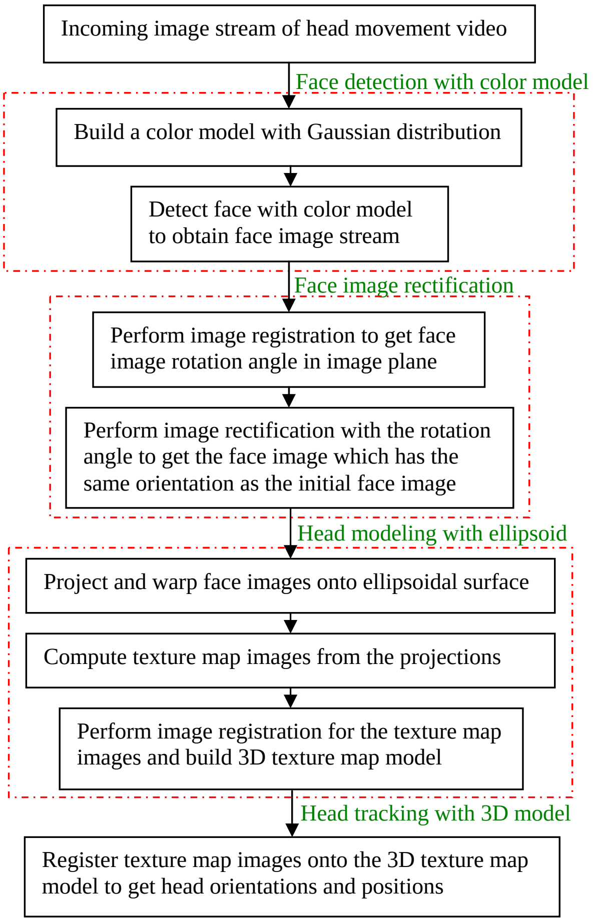

Incoming image stream of head movement video
Face detection with color model
Build a color model with Gaussian distribution
Detect face with color model to obtain face image stream
Face image rectification
Perform image registration to get face image rotation angle in image plane
Perform image rectification with the rotation angle to get the face image which has the same orientation as the initial face image
Head modeling with ellipsoid
Project and warp face images onto ellipsoidal surface
Compute texture map images from the projections
Perform image registration for the texture map images and build 3D texture map model
Head tracking with 3D model
Register texture map images onto the 3D texture map model to get head orientations and positions

Figure 1: Approach for Head Tracking with 3D Texture Map Model

3. FACE DETECTION

In this section, we investigate face in two dimensional images. We build a skin color model with Gaussian distribution and segment face region for each frame of incoming image stream. The outcome of this process is the face image stream which can be used to build head 3D texture map model.

To segment human face region from background, we apply well-developed skin color model theory established by Yang and other researchers [16, 17, 7, 18]. A survey on skin color detection techniques is available in [19].

The skin color model theory is based on the facts that (a) human skin color are clustered in the color space; (b) the skin color differences among people of difference in races, sexes, and ages can be reduced by intensity normalization; (c) under certain lighting conditions, a skin color distribution can be characterized by a multivariate Gaussian distribution in the normalized color space. Therefore, we can model human face with different color appearances in the normalized color space. By computing the probability of a pixel in skin color Gaussian distribution we can employ the maximum likelihood threshold to estimate the face region.

We build the face skin color model in RGB space. Usually the RGB space in original color image includes luminance component, which makes it difficult to characterize skin color because lighting effects change the appearance of the skin. In order to eliminate lighting effects, the original color images are converted to chromatic color images. The formula is

$$\begin{cases} r = \frac{R}{R+G+B} \\ b = \frac{B}{R+G+B} \\ g = \frac{G}{R+G+B} \end{cases} \tag{1}$$

In above, as $r + b + g = 1$, there are only two independent components, so we omit the third component. For each pixel, we have a color vector $\mathrm{x} = (r\ b)^T$. The two dimensional Gaussian distribution model is expressed as N(m,C) with

$$\begin{cases} \mathrm{m} = E\{\mathrm{x}\} \\ \mathrm{C} = E\{(\mathrm{x\text{-}m})(\mathrm{x\text{-}m})^T\} \end{cases} \tag{2}$$

where,

m is the mean vector;

C is the covariance matrix.

We manually collect over 20 human faces deliberated from different races, age, sex in our lab video database. They are used to compute a generalized skin color Gaussian distribution model. The skin color mean vector and covariance matrix are

$$\mathrm{m} = \begin{pmatrix} 105.08 \\ 141.08 \end{pmatrix}, \mathrm{C} = \begin{pmatrix} 137.96 & 125.34 \\ 125.34 & 251.15 \end{pmatrix}.$$

The model is shown in figure 2 with r plotted against b.

The skin color model above is a generalized from multiple subjects under different lighting conditions. In actual tracking implementation, we sample several image frames from an individual video and generate a subject-oriented skin color model. If necessary, we might even update the skin color frame by frame for a tracking video sequence. But the experiment shows that the skin color model value (mean and covariance) do not change significantly and the update does not gain much accuracy in computing the skin color probability later on.

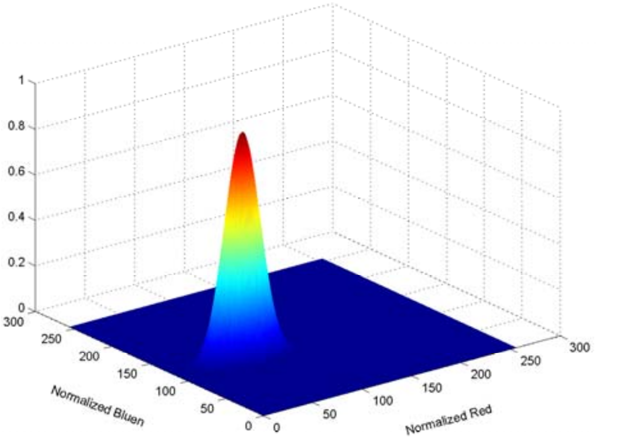
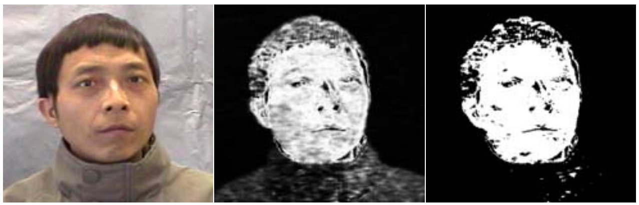


Figure 2: Human Face Skin Color Gaussian Distribution Model



(a) Face (b) Probability (c) Region

Figure 3: Face, Skin Color Probability, Skin Color Region vs Non-Skin Color Region

Once the skin color model is built, we can use it to compute the probability of each pixel in an image being skin color or not. The probability of a pixel with color x is

$$P(\mathrm{x}) = \exp\left\{-\frac{1}{2}(\mathrm{x\text{-}m})^T\mathrm{C}^{-1}(\mathrm{x\text{-}m})\right\}. \tag{3}$$

Given a probability threshold, we can threshold non-skin pixels in the image, and find the face candidate regions. As skin colors do vary between each individual subject, we shall find best threshold value for different subject under different application (background, illumination, etc.)

In implementation, we scale the skin color probability of every pixel in an image from [0, 1] to [0, 255], thus we can create a probability gray scale image. Furthermore, by thresholding, we create a black/white binary image to represent non-skin color regions and skin color regions. One example is shown in figure 3.

4. FACE IMAGE REGISTRATION

As described in section 2, we model the human head as an ellipsoid with a 360^o wide image. We project face images onto the ellipsoidal surface to build this model. Before we do the projection, we need to know the head rotate angle between current and previous frames in 2D image plane (We require that the subject face to the camera in initial frame). With this rotate angle, we can rectify the current frame, so that we have same head orientation for these two frames in 2D image plane. With these rectified face images, we build head 3D model.

4.1 Procedure for Face Image Registration

In this section, we propose a voting procedure for the face

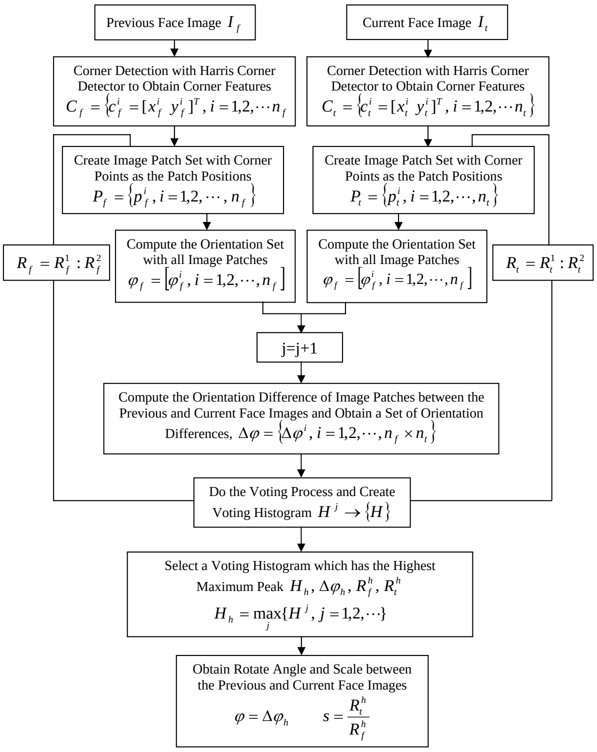

Previous Face Image I_f
Current Face Image I_t
Corner Detection with Harris Corner Detector to Obtain Corner Features $C_f = \{c_f^i = [x_f^i \; y_f^i]^T, i = 1,2,\cdots n_f$
Corner Detection with Harris Corner Detector to Obtain Corner Features $C_t = \{c_t^i = [x_t^i \; y_t^i]^T, i = 1,2,\cdots n_t\}$
Create Image Patch Set with Corner Points as the Patch Positions $P_f = \{p_f^i, i = 1,2,\cdots, n_f\}$
Create Image Patch Set with Corner Points as the Patch Positions $P_t = \{p_t^i, i = 1,2,\cdots, n_t\}$
$R_f = R_f^1 : R_f^2$
Compute the Orientation Set with all Image Patches $\varphi_f = [\varphi_f^i, i = 1,2,\cdots, n_f]$
Compute the Orientation Set with all Image Patches $\varphi_f = [\varphi_f^i, i = 1,2,\cdots, n_f]$
$R_t = R_t^1 : R_t^2$
j=j+1
Compute the Orientation Difference of Image Patches between the Previous and Current Face Images and Obtain a Set of Orientation Differences, $\Delta\varphi = \{\Delta\varphi^i, i = 1,2,\cdots, n_f \times n_t\}$
Do the Voting Process and Create Voting Histogram $H^j \rightarrow \{H\}$
Select a Voting Histogram which has the Highest Maximum Peak $H_h, \Delta\varphi_h, R_f^h, R_t^h$
$H_h = \max_j \{H^j, j = 1,2,\cdots\}$
Obtain Rotate Angle and Scale between the Previous and Current Face Images
$\varphi = \Delta\varphi_h \qquad s = \frac{R_t^h}{R_f^h}$

Figure 4: 2D Face Image Registration

image registration to obtain head rotation angle in 2D image plane. Figure 4 shows the whole process of procedure. we detect the corner points in both images with Harris corner detector. Using these corner points as positions, create image patches. In order to deal with rotation situation, we use a circle as the shape of image patches. By change the radius, we can change the size of image patches, so that we can deal with the scaling situation. For each corner point we can create a patch, so we can obtain an image patch set for previous face image and another set for current face image. By computing orientation for each image patch with eigenvector approach, we can obtain an orientation set for previous face image patches and another orientation set for current face image patches. With these two orientation sets, we can compute the orientation difference for each image patch between previous and current face images. We use these orientation differences to vote and create a voting histogram. The orientation difference corresponding to the maximum peak of the histogram is the rotate angle between previous and current face images. To deal with the scaling problem, we need to change the size of image patch by changing the radius, so that the corresponding patches on previous and current face images can cover the same scene. Use different sizes of image patches to create different voting histograms. With the voting histogram which has the highest maximum peak, we can determine the value of the scaling between previous and current face images.

4.2 Orientation of an Image Patch

For a given image patch $p(i,j)(i = 1, 2, ..., m)$, the covariance matrix is defined as

$$COV_p = E\{(X - m_x)(X - m_x)^T\} \tag{4}$$

where,

$X = \begin{pmatrix} i \\ j \end{pmatrix}$ is the position of a pixel;

$m_x = \begin{pmatrix} m_{xi} \\ m_{xj} \end{pmatrix}$ is the centroid of the image patch $p(i,j)$, the first order moment,

$$\begin{cases} m_{xi} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} jp(i,j)}{\sum_{i=1}^{n}\sum_{j=1}^{m} p(i,j)} \\ m_{xj} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} ip(i,j)}{\sum_{i=1}^{n}\sum_{j=1}^{m} p(i,j)} \end{cases} \tag{5}$$

From equation 4, we have

$$COV_p = \frac{1}{N_2 - 1}\begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} \tag{6}$$

where,

$$\begin{cases} C_{11} = \sum_{i=1}^{n}\sum_{j=1}^{m} p(i,j)(i - m_{xi})^2 \\ C_{12} = \sum_{i=1}^{n}\sum_{j=1}^{m} p(i,j)(i - m_{xi})(j - m_{xj}) \\ C_{21} = \sum_{i=1}^{n}\sum_{j=1}^{m} p(i,j)(j - m_{xj})(i - m_{xi}) \\ C_{22} = \sum_{i=1}^{n}\sum_{j=1}^{m} p(i,j)(j - m_{xj})^2 \end{cases} \tag{7}$$

The eigenvalues can be found by solving

$$|COV_p - \lambda I| = 0 \tag{8}$$

Equation 8 will give us two eigenvalues. Suppose λ_1 is the largest eigenvalue and λ_2 is the smallest eigenvalue. The normalized eigenvectors V_1 and V_2 that correspond to the eigenvalues λ_1 and λ_2 are of course orthogonal. The eigenvalues satisfy the relations

$$\begin{cases} V_1^T COV_p V_1 = \lambda_1 \\ V_2^T COV_p V_2 = \lambda_2 \end{cases} \tag{9}$$

The direction of eigenvector V_1 is defined as the orientation of image patch $p(i,j)$.

$$\begin{aligned} tan2\varphi_1 &= \frac{2C_{12}}{C_{11} - C_{12}} \\ &= \frac{2\sum_{i=1}^{n}\sum_{j=1}^{m} p(i,j)(i - m_{xi})(j - m_{xj})}{\sum_{i=1}^{n}\sum_{j=1}^{m} p(i,j)(i - m_{xi})^2 - \sum_{i=1}^{n}\sum_{j=1}^{m} p(i,j)(j - m_{xj})^2} \end{aligned} \tag{10}$$

or

$$\begin{cases} \varphi_1 = arctan\left(\frac{\lambda_1 - \sum_{i=1}^{n}\sum_{j=1}^{m} p(i,j)(i - m_{xi})^2}{\sum_{i=1}^{n}\sum_{j=1}^{m} p(i,j)(i - m_{xi})(j - m_{xj})}\right) \\ \varphi_2 = arctan\left(\frac{\sum_{i=1}^{n}\sum_{j=1}^{m} p(i,j)(i - m_{xi})(j - m_{xj})}{\lambda_2 - \sum_{i=1}^{n}\sum_{j=1}^{m} p(i,j)(i - m_{xi})^2}\right) \end{cases} \tag{11}$$

where, φ_1 is the direction of eigenvector V_1 and φ_2 is the direction of eigenvector V_2. Figure 5 shows the orientation of an image patch.

4.3 Image Patch Shapes and Positions

In this registration procedure, how to choose image patches is very important. Although we do not need to know which patch in the current face image corresponds to which patch in previous face image, we do need the corresponding patches to cover the same scene, so the positions and shapes of the image patches are very important.

In order to cover same scene for the corresponding patches, we need to find some features in images to determine the positions of the patches. Image corners are good features for image patch positions. We apply Harris corner detection algorithm [20] to extract these corner points. As shown in

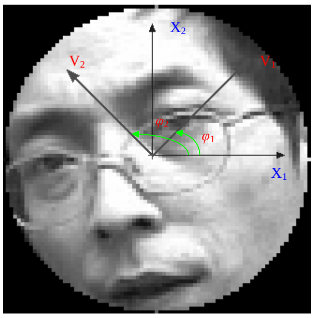
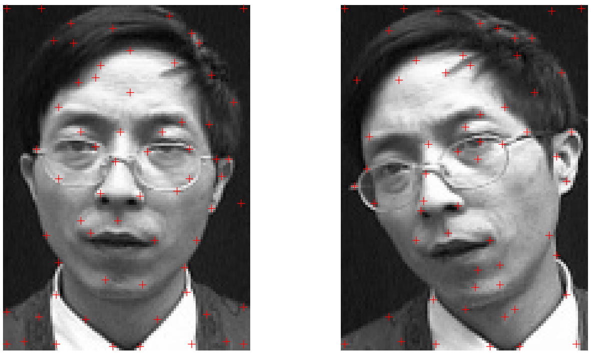


Figure 5: Image Patch Orientation

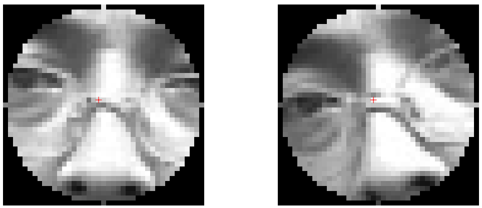


(a) Previous face Image (b) Current Face Image

Figure 6: Image Patch Positions

figure 6, although the current image has rotation related to the initial image, the corner feature has not changed in its own image. If we choose a proper shape for the image patch, it can cover same scene.

As mentioned above, image patch shapes are crucial for this registration approach. Because the current image has rotation related to the previous image, the rectangle shape can not be used. To handle this rotation, we choose a circle as the shape of image patches. As shown in figure 7, although there is rotation between the current and previous images, the image patches cover same scene if the patches are in the right positions.



(a) Previous Image Patch (b) Current Image Patch

Figure 7: Image Patch Shapes

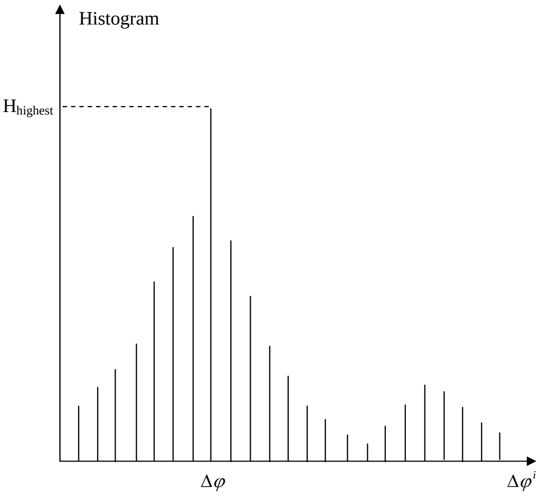


Figure 8: Voting Histogram

4.4 Voting Histogram for Head Rotation and Scaling in 2D Image Plane

Suppose the rotation angle between current image I_t and previous image I_f is φ . p_t is a patch on current image I_t and p_f is a patch on previous image I_f . $\varphi_t$is the orientation of image patch p_t and $\varphi_f$is the orientation of image patch p_f . The orientation difference between these two patches is

$$\Delta\varphi = |\varphi_t - \varphi_f| \tag{12}$$

If p_t and p_f are corresponding image patches, we will have

$$\Delta\varphi = |\varphi| \tag{13}$$

On the basis of above, we can obtain the rotation angle of the current face image related to the previous face image by voting process. First, we compute the orientation differences of all image patches between current face image I_t and previous face image I_f, so we obtain a number of $\Delta\varphi^i(i = 1, 2, ..., n_t \times n_f)$. Second, we create the histogram for the $\Delta\varphi^i(i = 1, 2, ..., n_t \times n_f)$. Finally, we choose the $\Delta\varphi$ corresponding to the maximum peak of the histogram as the rotation angle between current and initial images shown in figure 8.

If the scale between the current and previous face images is not one, we can obtain the value of the scaling through a serial of voting processes. By changing the size of the image patches and computing the voting histograms, we can obtain a series of voting histograms shown in figure 9. Choose the one which has the highest maximum peak in voting histograms. Suppose the image patch size on the current image is $A_{ti}(i = 1, 2, , n_1)$ and the image patch size on the reference image is $A_{fj}(j = 1, 2, , n_2)$. For each pair of image patch sizes A_{ti} and A_{fj}, we can obtain a voting histogram $H^k(k = 1, 2, , n_1 \times n_2)$ and corresponding orientation difference $\Delta\varphi^k(k = 1, 2, ..., n_1 \times n_2)$.

Choose the one which has the highest maximum peak $H_{highest}$

$$H_{highest} = \max_k\{H^k, k = 1, 2, ..., n_1 \times n_2\} \tag{14}$$

Let A_t and A_f denote the patch sizes on current and initial images corresponding to the histogram $H_{highest}$ which has the highest maximum peak. The value of the scaling between the current and previous images can be computed as

$$s = \sqrt{A_t/A_f}. \tag{15}$$

We also use this to estimate head position changes in Z direction.

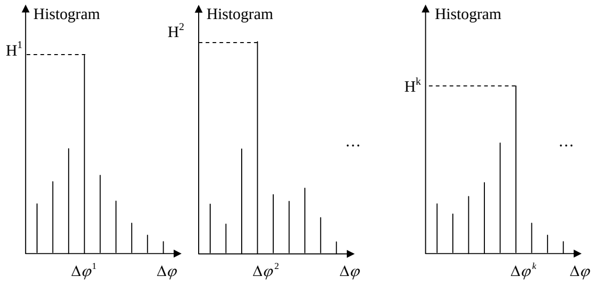


Figure 9: A series of Voting Histograms corresponding to difference patch sizes

In the mean time, the orientation difference $\Delta\varphi$ corresponding to the histogram $H_{highest}$ is the rotate angle φ between the current and previous images. Figure 10 shows an example of rotation angle between current and previous images.

5. HEAD MODELING WITH ELLIPSOID

5.1 Face Image Projection

Figure 11 shows the process of a face image project onto an ellipsoidal surface. We assume that the head is an ellipsoid with a 360° wide image, or more precisely, a video showing facial expression changes, texture mapped onto the ellipsoidal surface. Only an 180° wide slice of this texture is visible in any particular frame; this corresponds with the visible portion of the face in each video image.

Shown as figure 11 (a), suppose we have a point q on ellipsoidal surface. Its orientation is determined by (θ, φ). The corresponding point on image plane I_p is $q^{'}$. For point $q(x, y, z)$ on ellipsoidal surface, we have

$$\begin{cases} y = Rcos\varphi \\ x = Rsin\varphi cos\theta \\ z = Rsin\varphi sin\theta \end{cases} \tag{16}$$

On the other hand, for the big ellipse, we have

$$\begin{cases} y = r_a cos\varphi \\ x = r_b sin\varphi \end{cases} \tag{17}$$

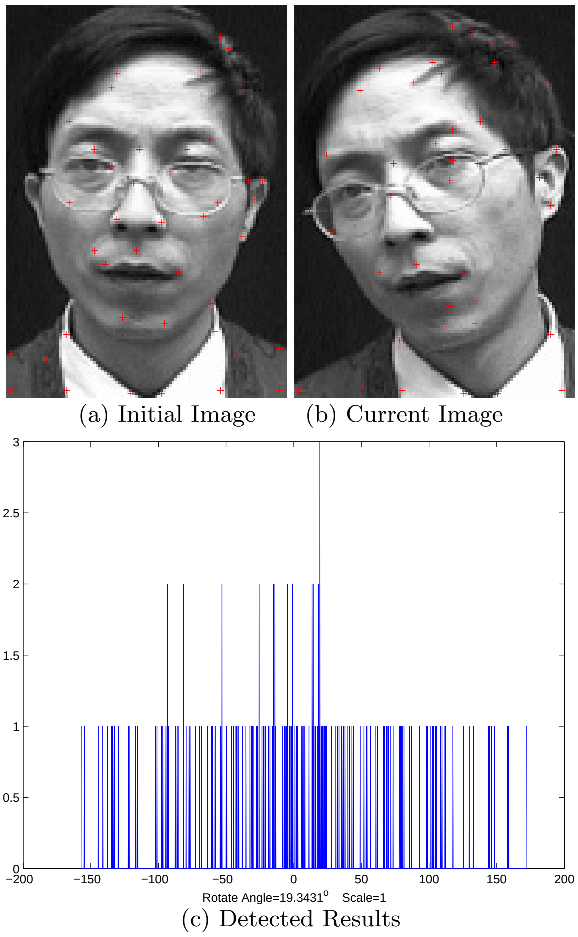
and

$$R = x^2 + y^2 \tag{18}$$

So the point q on ellipsoidal surface can be determined by $(\theta, \varphi, r_a, r_b)$

As shown in figure 11 (b), if we consider about 3D object to 2D image, the projection of arc cd on ellipsoidal surface is the line cd in image plane I_p. Now we need to do the reverse process. We project pixels on line cd, $q^{'}$, onto ellipsoidal surface, q. Then we stretch arc cd into a line c_1d_1.

So given a point $q^{'}$ on face image, we can project it onto the ellipsoidal surface and get point q. After stretch the arc into a line, we get point q_1. By this way, for each point in face image, we can obtain a corresponding point on the stretched ellipsoidal surface and its orientation is determined by θ and φ. So we can transfer face image on 2D image plane into texture map on (θ, φ) plane. Figure 12 shows an example of texture map of a face image.


(a) Initial Image (b) Current Image

(c) Detected Results

Figure 10: Head 2D rotate angle

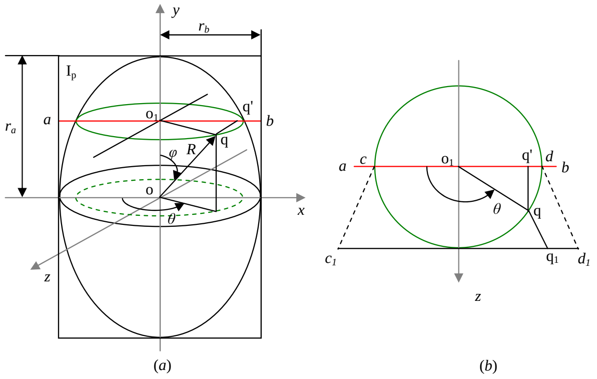
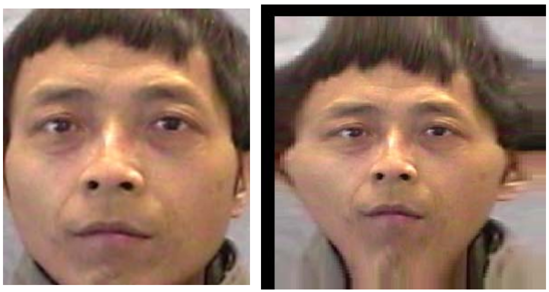
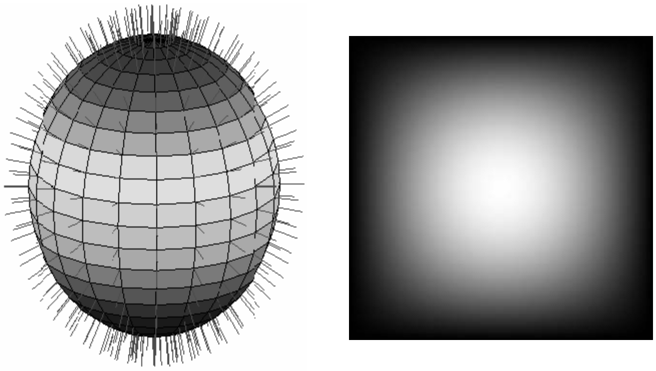


Figure 11: Face Image Projection onto An Ellipsoidal Surface



(a) Face Image (b) Texture Map

Figure 12: An Example of Texture Map of a Face Image



(a) (b)

Figure 13: The Normal of Ellipsoidal Surface and Confidence Map

5.2 Confidence Maps

As the input image is inverse projected onto ellipsoid surface, not all pixels have equal confidence. Here, we use normal of the surface to measure the confidence.

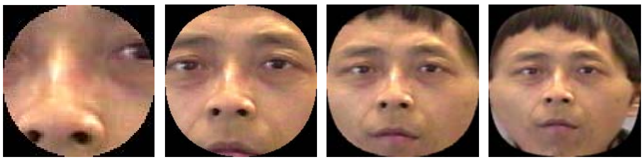
For the surface expressed by $F(x, y, z) = 0$, a normal at a point (x, y, z) on the surface is given by the gradient $\nabla F(x, y, z)$. Usually, we normalize it

$$N = \frac{\nabla F(x, y, z)}{\|\nabla F(x, y, z)\|}. \tag{19}$$

Figure 13 (a) shows the normal of an ellipsoidal surface.

Now, we can define our confidence for the projection of an 2D image onto an ellipsoidal surface. If the direction of normal at a point is the same as the direction of projection axis (camera's optical axis), the confidence is maximum. If the angle between the direction of normal and the direction of projection axis is 90^o, then the confidence is zero.

As we described in section 5.1, we transfer face image on 2D image plane into texture map on (θ, φ) plane. So our confidence map for the ellipsoidal surface needs also to be in (θ, φ) plane. Figure 13 (b) shows the confidence map in (θ, φ) plane. With different level confidence, we can get different sizes of texture map image. Figure 14 shows texture maps which the confidences are greater than or equal $90\%, 70\%, 50\%$,and 30% separately.



(a) (b) (c) (d)

Figure 14: Texture Maps in Different Confidence Levels

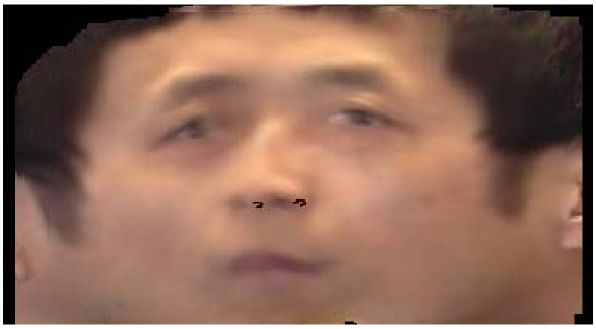


Figure 15: Head 3D Texture Map Model

5.3 Head 3D Map Model

For the incoming image stream, we perform face detection using Gaussian skin color model. We obtain a face image stream. We project each frame of the face image stream onto an ellipsoidal surface and compute a texture map in (θ, φ) plane. With a confidence map, we can select the texture map area which corresponding to the threshold of the confidence. We register each texture map of the face image stream to the texture map of initial frame (we assumed that the subject faces to camera at the beginning). At last we obtain head 3D texture map model in (θ, φ) plane. Figure 15 shows a head 3D texture map model built by this approach.

6. HEAD TRACKING WITH 3D TEXTURE MAP MODEL

Similar as the process of building 3D texture map model, the head tracking process also needs to do these steps: detect face region for each frame of incoming image stream; project each face image onto an ellipsoidal surface and compute texture map. By registering the texture map of incoming face image stream onto the head 3D texture map model, we obtain head position and orientation for each frame. In this way, we can perform head tracking with the 3D texture model. In the mean time, we also use the tracking results to modify the head 3D texture map model.

7. CONCLUSION

In video-based multi-modal analysis of formal and informal meetings, understanding of video events associated with the meetings needs to understand human communication behavior. In order to realize automatic detection, recognition, annotation of these video events, we need robust and efficient human body, head, face and hand tracking algorithms in a clustered background. In this paper, we proposed a simple and efficient head tracking algorithm. It is developed by combining appearance-based and model-based approaches. The algorithm is suitable for the case of low resolution meeting video and allows great change in orientation. In this algorithm, we model the human head as an ellipsoid with

360^o wide images. The face is an image patch on the ellipsoidal surface. We build a head 3D texture map model dynamically on the fly by projecting the face images onto the ellipsoidal surface. Because the head model is in (θ, φ) space, we register each frame of the incoming image stream onto the 3D texture map model to obtain head positions and orientations. In the whole process, the face region of each frame is segmented using a Gaussian skin color model developed using skin color samples and Gaussian distribution. We also proposed a voting procedure to obtain head rotation angle in 2D image plane. We can rectify face images in same orientation using the angle. The head tracking results are very important to estimate the gaze orientations of meeting participants, which is very important for the multi-modal analysis of video events of meetings.

8. REFERENCES

[1] D. McNeill, "Growth points, catchments, and contexts," *Cognitive Studies: Bulletin of the Japanese Cognitive Science Society*, vol. 7, no. 1, pp. 22–36, 2000.

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

[3] T. Darrell and et al, "Integrated person tracking using stereo, color, and pattern detection," *International Journal of Computer Vision*, vol. 37, pp. 175–185, 2000.

[4] D. DeCarlo and D. Metaxas, "Optical flow constraints on deformable models with applications to face tracking," *International Journal of Computer Vision*, vol. 38, pp. 99–127, 2000.

[5] K.S. Huang and M. Trivedi, "Video arrays for real-time tracking of persons, head and face in an intelligent room," *Machine Vision and Application*, vol. Omni-directional Vision, Special Issue, 2001.

[6] Stephen McKenna and Shaogang Gong, "Tracking faces," in *Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition (FG'96)*, Killington, Vermont USA, 10 1996, pp. 271–276.

[7] J. Yang and A. Waibel, "A real-time face tracker," in *Proceedings of the Third Workshop on Applications of Workshop on Computer Vision (WACV'96)*, Sarasota, Florida, 1996.

[8] J. Ahlberg, "Real-time facial feature tracking using an active model with fast image warping," in *International Workshop on Very Low Bit-rate Video Coding*, Athens, Greece, 2001.

[9] Michael J. Black and Yaser Yacoob, "Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion," in *the Fifth IEEE International Conference on Computer Vision (ICCV '95)*, MIT, Cambridge, MA, USA, 6 1995, pp. 374–381.

[10] Y. Huang, T.S. Huang, and H. Niemann, "Segmentation-based object tracking using image warping and kalman filtering," in *IEEE Signal Processing Society 2002 International Conference on Image Processing*, Rochester, New York, 2002.

[11] L. Reveret and I. Essa, "Visual coding and tracking of speech related facial motion," in *IEEE International Workshop on Cues in Communications*, Kauai, Hawaii, 2001.

[12] F. Smeraldi, N. Capdevielle, and J. Bigun, "Face authentication by retinotopic sampling of the gabor decomposition and support vector machines," in *2nd International Conference on Audio and Video Based Biometric Person Authentication (AVBPA'99)*, Washington DC, 1999.

[13] Ali Azarbayejani and Alex P. Pentland, "Recursive estimation of motion, structure, and focal length," *PAMI*, vol. 17, no. 6, pp. 562–575, 6 1995.

[14] J. Sturm and et al, "Real time tracking and modeling of faces: An ekf-based analysis by synthesis approach," in *IEEE International Workshop on Modeling People*, Corfu, Greece, 1999.

[15] T. Jebara and A. Pentlan, "Parameterized structure of motion for 3d adaptive feedback tracking of faces," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, 1997.

[16] J-C. Terrillon, M. David, and S. Akamatsu, "Automatic detection of human faces in natural scene images by use of a skin color model and of invariant moments," in *IEEE Proceedings of the Third International Conference on Automatic Face and Gesture Recognition*, April 14-16 1998, pp. 112–117.

[17] Benjamin D. Zarit, Boaz J. Super, and Francis K. H. Quek, "Comparison of five color models in skin pixel classification," in *IEEE Proceedings of International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, September 26-27 1999, pp. 58–63.

[18] J. Yang, W. Lu, , and A. Waibel, "Skin-color modeling and adaptation," in *Proceedings of the Third Asian Conference on Computer Vision (ACCV98)*, Hong Kong, 1998, pp. 687–694.

[19] V. Vezhnevets, V. Sazonov, and A. Andreeva, "A survey on pixel-based skin color detection techniques," in *Proc. Graphicon-2003*, Moscow, Russia, September 2003, pp. 85–92.

[20] C. J. Harris and M. Stephens, "A combined corner and edge detector," in *In Proc. 4th Alvey Vision Conf.*, Hong Kong, 1988, p. 147C151.