Model-based Feature Refinement by Ellipsoidal Face Tracking

Sung-Uk Jung  
Human Identification Research Team, ETRI,  
Daejeon 305-350, Korea  
brcastle@etri.re.kr

Mark S. Nixon  
School of ECS, University of Southampton,  
SO17 1BJ, UK  
msn@ecs.soton.ac.uk

Abstract

We describe a new method to relieve common assumptions/restrictions in head tracking by using a model-based approach. This improves local feature matching which only considers the pattern around the extracted feature excluding the object shape, so that misalignment can occur. In this paper, to overcome constraints on motion we consider region- and distance-based feature refinement methods to validate the local features used when tracking the ellipsoidal object. We also present a direct mapping method to reconstruct 3D feature positions for tracking. The utility of the new method has been demonstrated for face pose estimation using the Boston face database.

1. Introduction

Feature extraction is a fundamental step for analyzing object’s movement in video. In visual surveillance images there might be large pose and size variations. Therefore, features should be invariant to image scale, rotation, and affine transformations. Further, the features detected in successive images should be robust to change in illumination and 3D viewpoint, and to noise.

There are many approaches for local feature extraction including the Scale-Invariant Feature Transform (SIFT) [1] and Speeded Up Robust Features (SURF) [2]. To determine the relationship between two images the first step can be to extract their corresponding points. Then, the second requires filtering the invalid features. Generally, local feature extraction consists of three procedures: detecting feature positions, calculating the near-region patterns by a descriptor and filtering/matching local features with the descriptor. The problem is that matching with only the feature descriptor can result in misalignment of feature matches since it considers the pattern around the extracted point and object shape remains unused.

In previous studies many approaches for head pose estimation have been developed using local feature and patterns. Liu et al. [3] used SIFT features to match the corresponding feature points. Using Epipolar geometry [4], the fundamental matrix was calculated to obtain the pose information. Hager and Belhumeur [5] generated a 2D plane model using a single camera and Lucas-Kanade tracking. Cascia et al. [6] generated a 3D cylinder model where 3D head motion was treated as a linear combination of motion templates and orthogonal illumination templates. Basu et al. [7] interpreted the optical flow in terms of the possible rigid motion and applied it to heads with a variety of shapes and hair styles, using 3D ellipsoidal model. Xiao et al. [8] used a 3D cylinder model to track the head. An iteratively Re-weighted Least Squares technique was adapted to fit the face to the model. Also, the templates were updated to reduce the effects of self-occlusion and
gradual lighting changes while tracking. Jang and Kanade [9] initialized the face using Bayesian Tangent Shape Model face alignment method. SIFT and normalized correlation methods were used to extract and match the feature points.

In this paper, we propose a feature refinement method to obtain the valid feature from a selection of features. This method estimates the head pose using a 3D ellipsoidal model and a non-linear optimization method. Based on the model we develop a direct mapping from 2D image coordinates to object coordinates.

2. Local feature extraction in pose and size variation

To illustrate the potential for feature misalignment we use the human face as a target object. We extracted the face features using the SIFT and SURF descriptors. The experimental samples are from the XM2VTS face database [10] and the Sheffield face database [11].

In figure 1 the green points show the extracted interest points and the lines display the matched points. In figure 1(a), there are many matching points in the high resolution images (692 × 548 pixels). In total, there are 157 matching points, including mismatches. In the right image of figure 1(a) the image is reduced by Bicubic image resizing (692 × 548 to 300 × 240 pixels). Then, the corresponding points are extracted. There are 1906 points in the high resolution image, 274 points in the low resolution image, and the number of matching points is 272. The result shows that over-fitting can take place during matching. Though this result depends on the threshold value within SIFT matching it demonstrates that there are difficulties in extracting corresponding points if the images have different size. In figure 1(b) the samples concerned pose and size variation showing misalignments with the SURF descriptor. Essentially, misalignments occur because SURF and SIFT are based on using a 2D image homography to determine the corresponding points. For a 3D shape like a face, occlusion or pattern distortion results from pose variation when the shape is projected into a 2D image.

From above samples, we can conclude that it is reasonable to refine the valid features. We choose to refine the SIFT features, in that more points are detected by SIFT (as in figures 1(a) and (b)).

3. Model-based feature filtering

The 3D ellipsoidal model is defined by following: a point on a 3D object $P_o$ can be represented by $[X_o, Y_o, Z_o]$ where $X_o=r_x \sin \alpha \sin \beta$, $Y_o=r_x \cos \alpha$, and $Z_o=r_z \sin \alpha \cos \beta$. $r$ is the radius along each axis. The angular resolu-

![Image](attachment:image.png)

![Graph](attachment:graph.png)

...tion is one degree, so that the total number of model components is 360 × 360.

Once the initialization process is complete, which requires manual specification of the rotation and translation of the model in the first frame, the invalid features can be removed. Figure 2(a) shows the extracted SIFT features. The green crosses depict the SIFT points which are matched to the following frame. The corresponding points selected by the SIFT descriptor are refined by a region-based and a distance-based measure.

First, the SIFT matching points within ± 50° from the center point are considered. The center point is the intersection between the initial translation vector and the 3D model. Figure 2(c) displays the region. The yellow region is the fitted model, the red region shows the region within ± 50° from the center point ($\alpha = 90°$ and $\beta = 0$ in the model definition) and the white cross in the model is the center point. As shown in the figure, the red region can cover the facial components (eyes, nose and mouth) which have rich information on head pose.

The second is to filter the invalid matching points based on the distances between each pair of potential corresponding points. The distances can show the distribution of movements. In a histogram of the distances, the misaligned points can be detected and removed. In an alternative representation, the distribution of the histogram can be modeled as Gaussian ($X \sim N(\mu, \sigma^2)$). The SIFT points are taken between $\mu-2\sigma$ and $\mu+2\sigma$, covering 68.2% of the Gaussian distribution. Figure 2(d) shows the distance distribution from each pair of the matched SIFT points. The final valid features are shown in figure 2(b).

After filtering, the 3D positions of the SIFT points
in the object coordinates are reconstructed by direct mapping.

As shown in figure 3, the 3D point on the ellipsoidal model is mapped to the 2D point in the image via rotation, translation, and projection. In figure 3, a model point \( P_o \) is mapped to the point \( P_e \) by changing from the object coordinates to the camera coordinates. Then, it is mapped to the point \( P_i \) in the image plane by the camera matrix. Conversely, if the point \( P_i \) is known, the point \( P_o \) (in object coordinates) can be found directly. Therefore, assuming that the point \( P_i \) is an extracted SIFT point, the 3D position in the object coordinate can be obtained directly.

Practically, the ellipsoidal model \( E_o \) and \( E_e \) has 360 \( \times \) 360 (spherical) points. For the projected ellipsoidal model \( E_{2D} \) only 180 \( \times \) 360 points are valid due to invisible parts. An extracted SIFT point could be one from the model \( E_{2D} \), or located between points. Therefore, to reconstruct an exact 3D position in the model \( E_o \) we approximately linearize the position using the distance ratio between the extracted SIFT point and around nine points on the model.

Let \( X_o \) and \( P_o \) be the extracted SIFT points and the points in the model \( E_{2D} \). \( X_o' \) and \( P_o' \) are the corresponding points in the model \( E_o \). Let the distance between \( X_o \) and \( P_o \) be \( d_n \)

\[
    r_k = (1/d_n) \sum_{l=1}^{n} 1/d_k \quad \text{then} \quad \sum_{k=1}^{n} r_k = 1 \tag{1}
\]

The 3D position, \( X_o' \), can be calculated by

\[
    X_o' = \sum_{k=1}^{n} r_k \cdot P_o' \tag{2}
\]

From the above relationship the 3D position is reconstructed and this 3D position will be used to calculate the motion vector.

4. Motion vector estimation

The Levenberg-Marquadt algorithm is a non-linear optimization algorithm which can provide the numerical solution to minimize an objective function.

We define the objective function to extract the motion information as

\[
    \arg \min_{\Delta t, \Delta r} \sum_{t=1}^{n} |p_{2d} - K(RP_{3d} + T)| \tag{3}
\]

where \( p_{2d} \) are the SIFT points in next frame, \( P_{3d} \) is the reconstructed 3D points from the 2D SIFT points in the current frame. The rotation matrix \( R \) contains \( \Delta \theta_x, \Delta \theta_y, \Delta \theta_z \) and translation matrix \( T \) contains \( \Delta x, \Delta y, \Delta z \).

We assume the camera matrix \( K \) is given. Basically, Equation 3 finds the rotation and translation matrices which minimize the distance between matched points in 2D space. The term, \( K(RP_{3d} + T) \) in Equation 3 converts the point from the coordinate 3D space into 2D image space. The motion information is extracted to adapt this objective function to Levenberg-Marquadt algorithm. In this way, the rotation and translation matrix can be updated using the previous motion information for each image frame.

5. Error correction

An error correction module is necessary since the model used is approximate (the assumption that the head shape is ellipsoidal) and an accumulated tracking / initialization error can occur. First, assuming that the difference between the potential motion vector calculated previously and the real motion vector is quite small, optical flow can be used to correct the motion vector.

Under small motion variance and no illumination change the velocity relationship between 2D and 3D motion can be described as following equation [12].

\[
    \frac{1}{Z} \left[ \begin{array}{c} f I_x - f I_y (x I_x + y I_y) \\ I_x' \end{array} \right] R \left[ \begin{array}{c} (1) - [P_o]_x \\ (1) - [P_o]_y \\ (1) - [P_o]_z \end{array} \right] \begin{bmatrix} \Delta t \\ \Delta \theta \end{bmatrix} = -I_t \tag{4}
\]

where \( I_x, I_y \), and \( I_t \) are image intensity gradients with respect to \( x, y, \) and \( t \) respectively. \( P_t \) is a point in object coordinates. \( R, \Delta t, \) and \( \Delta \theta \) are the 3D rotation matrix and instantaneous translation and rotation, respectively. \( \left[ (1) - [P_o]_x \right], [P_o]_y, [P_o]_z \) denotes a skew-symmetric matrix. \( I \) is the identity matrix and \( f \) is the focal length.

To deploy Equation 4 the rotation matrix \( R \) is changed into \( R_o \) which is the updated rotation matrix. Thus, the instantaneous rotation vector can be small. Therefore, even though there is huge variation between motion vectors Equation 4 can be used to correct the potential motion vector. A linear equation can be made by adapting the Equation 4 to all visible pixels. Then, a least squares solution is used to calculate the translation and rotation vector (\( \Delta t, \Delta \theta \)).

To verify the performance of face tracking we used the Boston face database [6] which has the 45 image.
sequences plus the ground truth of 3D head pose. The experimental procedure is as follows. First, corresponding SIFT points were extracted and filtered by the refinement methods. 3D positions of the selected points in the first frame are calculated by given a motion vector. Then, the potential motion vector is calculated using non-linear optimization between the corresponding 3D SIFT points. Finally, the corrected motion vector is obtained by optical flow.

The average errors across the whole database in each direction are shown in Table I where we compared our result with the previous methods using the same database. The difference with the comparator results can be viewed as illusory in the sense that the tracking results can depend on initialization. All methods show good performance results (less than 4° error in each direction). However, the comparator results are perhaps compromised in accuracy by the desire to achieve video-rate processing since these methods assume that there is small variance of head pose between frames. Unlike other methods, our new method can be more robust to translation since our method is based on local feature matching first, and then analyzing the pattern of face image. Figure 4 shows a sample of tracking result.

The blue (starred) line is the ground truth and the red (crossed) line is a test result. In this case, the yaw direction is varying from +20° to -40°.

6. Discussion and future work

The basic features used for face recognition can be refined using an object model. We can estimate the motion vector of the ellipsoidal object using the filtered SIFT features. The method differs from previous methods in two senses. First, we find 3D information from 2D information only using a single camera. Another is that we could remove the assumption in 3D position calculation that there should be a small variation between images. The approaches have been demonstrated with the Boston database showing that the tracking error was less than 4°. As such, this result can be adapted to the case of single view images with large change in ellipsoidal, cylinder and even plane object. In the future, we aim to solve some constraints such that the initialization is not automatic and that the object should be known.

References