3D Model Based Head Pose Tracking by Using Weighted Depth and Brightness Constraints

Guoyuan Liang, Hongbin Zha, Hong Liu
National Lab on Machine Perception
Peking University, People's Republic of China
{lianggy, zha, liuhong@cis.pku.edu.cn}

Abstract

This paper proposes a robust method of tracking human head poses from a sequence of monocular images. First we estimate the head pose parameters in the first frame by an affine correspondence based method developed in our lab. Then both the linear brightness and depth constraint equations derived from the small interframe rigid motion assumption are used to implement the fast tracking of the head poses. We also take advantage of geometry information of the features on the face surface to weight the brightness and depth constraint equations to get more accurate results. Finally, in order to diminish the effects of gradual illumination changes and occlusions, we estimate the reliability of the features frame by frame and dynamically update the reliable feature set. Experiments show the proposed method can robustly track the head poses especially for the types of motions which make obvious depth variation.

1. Introduction

Many researchers have proposed different approaches on face pose tracking in the past decade. Most of them can be classified into two categories: face property-based methods [3,9] and model-based methods [4,5,10,11]. In essence, the pose estimation problem based on 2D-3D feature correspondences is a nonlinear one and quite difficult to find a closed form solution. Some researchers [1,7] proposed direct and linear methods for the pose estimation. Harville et al [2] take advantage of the vedio-depth information provided by a stereo camera system and presented a linear method combining brightness and depth constraints to estimate the head poses.

In this paper, we present a robust approach using a 3D head model and weighted depth and brightness constraint equations to estimate head poses. There are three main improvements which contribute to the robustness in our approach. First, we use an affine correspondence based method which does not rely on the initial pose of the real face to estimate initial pose parameters as the starting point of pose tracking. Second, we take advantage of geometry information of the 3D face model to calculate the weights for each feature for scaling the brightness and depth constraints. Third, we evaluate the reliability of each feature and update the reliable feature set frame by frame. This technique can effectively diminish the error accumulations caused by gradual changes of illumination or self-occlusion. The performance of our approach was tested on several model head image sequences (the ground truth for motion parameters are known) and some real head image sequences. Experiments show our method can reliably and robustly track head poses in a wide range of head motion.

Fig.1 shows the block diagram of our system. We have available 3D head models which are generated with a *FastScan* laser scanner. The head images are captured by a *Mintron 64G-1K* camera which is calibrated beforehand by using a self-calibration method [6].

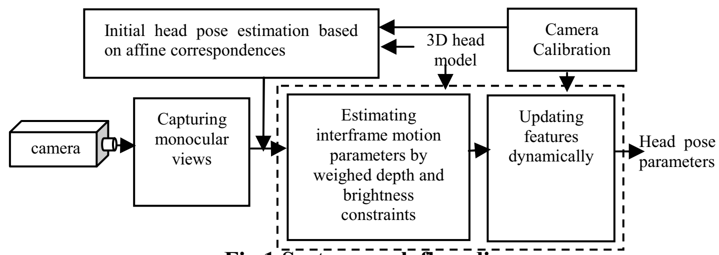

Fig.1 System work flow diagram

2. Motion Parameter Estimation

We have developed an affine correspondence based method [5] to estimate head poses in a single image. It is based on the observation that some features on the face (e.g. eyes and mouth corners) satisfy the condition of affine transformation. The method does not rely on the initial pose of the first frame in the image sequence. After the initial pose estimation, we generate a certain number of features on the face in the input image using the well-known KLT criteria [8] and backproject them to the 3D head model. KLT method ensures that the selected features are located in the region with rich texture in the face and can be tracked reliably. Based on rigid and small motion assumption between consecutive frames, we can simplify the nonlinear pose estimation problem to a linear constraint problem which can be solved reliably by some linear optimization methods.

2.1 Brightness and Depth Constraint Equations

It's well-known that the brightness constraint is based on the similarity of the intensity of features between adjacent frames. Harville et al [2] expand the idea to the depth field and derive the linear depth constraint. The head motion parameters and the variation of brightness and depth can be related by the perspective projection model. Let us denote the coordinates of a 3D feature as $\mathbf{X}=(X,Y,Z)^T$ and its 2D counterpart as $\mathbf{x}=(x,y)^T$, f the focal length of camera. Then the linear brightness constraint can be expressed by

$$-I_t=\frac{1}{Z}\begin{pmatrix} f\,I_x & f\,I_y & -(x\,I_x+y\,I_y)\end{pmatrix}\mathbf{Q}\,\mathbf{\Phi}, \tag{1}$$

where I_x, I_y and I_t are the image intensity gradients with respect to x, y and t, $\mathbf{\Phi}$ is the motion parameter vector, and $\mathbf{Q}$ is a matrix relative to the coordinates of a 3D feature.

Proceedings of the Third International Conference on Image and Graphics (ICIG'04)
0-7695-2244-0/04 $20.00 © 2004 IEEE

Using the same idea, we can derive the linear depth constraint equation in a similar form to (1):

$$-Z_t = \frac{1}{Z}\begin{pmatrix} f Z_x & f Z_y & -(Z + xZ_x + yZ_y) \end{pmatrix} \mathbf{Q}\,\mathbf{\Phi}, \quad (2)$$

where Z_x, Z_y and Z_t are the depth gradients with respect to x, y and t. Provided the feature number N, we can stack the brightness and depth constraints for all features in matrix form. Calculating the pose parameters then equals to solving the following optimization problem

$$\min \left\| \mathbf{W}(\mathbf{b} - \mathbf{H}\,\mathbf{\Omega}) \right\|^2. \quad (3)$$

Here $\mathbf{W}$ is an N×N diagonal weight matrix to describe the contribution of each feature. Provided N>6, the pose parameter vector $\mathbf{\Phi}$ can be solved by a weighted linear least square method. We will discuss how to determine the weights in the next section.

Generally speaking, the intensity is not very stable and easy to be affected by the illumination changes, self-occlusion and other subtle factors. The depth of a feature, on the contrary, is stable enough to provide reliable constraint for the pose estimation. Therefore, we can expect that the combination of the brightness and depth constraints produces more accurate results.

2.2 Determination of Feature Weights

The existence of image noises, illumination changes and self-occlusion will cause non-uniform intensity distribution in the image and should be represented quantitatively by the weights.

The first aspect to be considered is that the features located on the side of face surface should have small weights because they are easier to straddle the boundary or to be occluded. Fig.2 shows a 3D feature $\mathbf{X}$ and its 2D projection $\mathbf{x}$. O is the focus of camera and θ is the angle between the surface normal at $\mathbf{X}$ and the line connecting O and $\mathbf{X}$.

We define the expression for calculating the *weight for side* as

$$\omega_s = 1 - \min(|\theta|, \pi/2) \cdot 2/\pi \quad . \quad (4)$$

When $\theta \geq \pi/2$, ω_s equals to 0, which means $\mathbf{X}$ is invisible. When $\theta < \pi/2$, the weight varies with respect to θ linearly with smaller θ yielding larger weight.

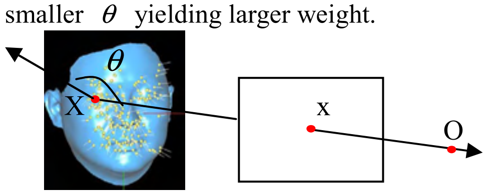

Fig.2 Determination of the weight by the angle θ between the surface normal and the direction from the feature to the focus.

Actually, the Taylor expansion used in the derivation of constraint equations assumes linear approximation to the variation of intensity/depth in the temporal and spatial field. When the interval between consecutive frames is small enough, the linearity in temporal field can be assured, but the linearity in spatial field depends on the surface curvature of the face. Smaller curvature part on the face surface means better linearity and should contribute more to the estimation of head poses. Therefore we calculate the *weight for curvature* by

$$\omega_c = 1 - \frac{1}{\min(C_f, T)} \cdot T \ , \quad (5)$$

where C_f is the Gauss curvature at the feature and T is a predefined threshold. When $C_f \geq T$, $\omega_c = 0$, which means the curvature is too large to satisfy the linear assumption. When $C_f < T$, smaller C_f yields larger ω_c, as Fig.3 shows.

Taking both aspects into account, we get the total weight for each feature as

$$\omega = C\omega_s\omega_c \ , \quad (6)$$

where C is a scalar constant.

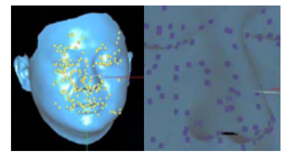

Fig.3 Features on the surface of a 3D model with different curvatures.

2.3 Inerframe Motion Estimation

In eq. (2), the depth gradients with respect to x and y is very easy to calculate if we have a 3D head model. So the problem is how to compute the depth gradient with respect to t. Some researchers use the video-rate depth information [2] to calculate the depth gradient with respect to t and combine the linear brightness and depth constraint eq.(1) and (2) to build a single linear system. Because the video-rate depth information is not available in our system, we use the brightness and depth constraints in a different way. The main steps are described as follows:

1) Assume the set of all features generated after the initial pose estimation is $P = \{p_i\}_{i=1}^{N}$. Form feature group set $G = \{G_j\}_{j=1}^{M}$, with each group $G_j = \{p_{gi}\}_{i=1}^{n}, n < N$ and $p_{gi} \in P$, p_{gi} being randomly selected from P.

2) Use the brightness constraint eq.(1) to estimate the head pose parameters Φ_j for each feature group G_j. The set of all pose parameter groups is $\Phi = \{\Phi_j\}_{j=1}^{M}$.

3) For each pose parameter group $\Phi_j \in \Phi$, calculate the median of the errors between the 2D features and the 2D projections of all 3D features. Then sort Φ in ascending order according to the median errors and select first M' groups to form a new set $\Phi' = \{\Phi_j\}_{j=1}^{M'}$, $M' < M$.

4) Calculate the depth at time t+1 for each pose parameter group in Φ'. Then use the technique to be described in section 4 to determine reliable pose parameter groups and calculate the mean of depths for each feature in P.

5) Use the depth mean to calculate the depth gradient with respect to time t. Then compute the final pose parameters by the depth constraint eq.(2).

We iterate above steps to realize the temporal head pose tracking.

3. Dynamic Feature Updating

Because of the inevitable presence of errors while tracking, some features may not be reliable any more. Although we have put smaller weights to the features which may be unreliable, a better way is to discard them when the projection errors are too large to be accepted. We update reliable feature set used in the calculation of pose parameters frame by frame. The main steps are explained as follows:

1) At each frame (except the first frame), after the estimation of pose parameters, calculate the errors between 2D features and 2D projections of all 3D features and build an error set $E=\{\varepsilon_i\}_{i=1}^{N}$, which is sorted in ascending order.

2) Give an initial guess (for example, 0.9) for the ratio r of reliable tracked features. Build a set of reliable features R which contains the first $N_r = N \times r$ elements in E and a set of unreliable features U which contains $N - N_r$ elements.

3) Compute the mean m and standard deviation σ of the elements in R.

4) Search in E for elements which satisfy the condition $|\varepsilon_i - m| > 2.5\sigma$, assume the number is N_u.

i) if $N_r + N_u = N$, then go to V, current R is the reliable feature set.

ii) if $N_u > N - N_r$, then $N_r = N_r - 1$, move the last element of R to the first position of U, then go to step 3.

iii) if $N_u < N - N_r$, then $N_r = N_r + 1$, move the first element of U to the last position of R, then go to step 3.

5) We remove unreliable features in U and use the KLT criteria [8] to generate the same number of features in the face to complement the reliable feature set R.

After the updating operation, we use the new feature set to estimate the next interframe motion parameters.

4. Experimental Results

We designed two experiments to examine the effectiveness of our approach. The first one aims to test the accuracy. We put a model head on a turntable which can rotate with known angles and simulate rotations of the face around X-, Y-, and Z-axes respectively, as Fig.4 shows.

Three image sequences are used in the first experiment. They were captured frame by frame, which ensures the rotation angles between adjacent frames are known. The first sequence begins with the front position towards the camera. Then the head makes -70°, $+70^\circ$, -70° and $+70^\circ$ rotations (1 degree per frame, total 280 frames) around the Y-axis and return to the starting position. The second and third sequences also begin with the front position and rotate around X- and Z-axes, respectively. The rotation angle ranges from -45° to $+45^\circ$ (1 degree per frame, total 180 frames).

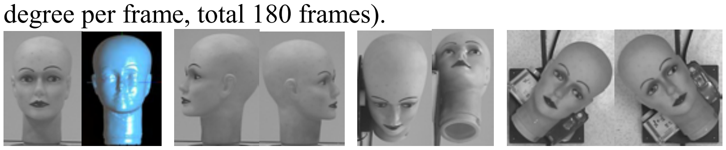

Fig.4 Model head image sequences. Left: the intensity image of the model head and its 3D data. Middle-left: two extrema and the initial position for the first sequence. The Middle-right: two extrema and the initial position for the second sequence. Right: two extrema and the initial position for the third sequence.

We demonstrate four sets of results for the following methods: 1) The Brightness Constraint (BC) based method only, 2) Both BC and Depth Constraint(DC) based method, 3) Weighted BC and DC and 4) Weighted BC, DC and Dynamic Feature Updating (DFU) based method, as Fig.5 shows. The estimated results only using BC are quite inaccurate for all three kinds of rotations because it only uses the intensity information in 2D images and is easily affected by noises, illumination changes and occlusions. The method combining BC and DC can remarkably improve the performance except for the rotations around Z-axis because there is no obvious depth variation there. Weighted BC and DC can reduce errors especially when the face reaches to the extrema of rotations where the occlusions and unreliable feature tracking may occur. The most notable effect of DFU is diminishing the accumulative errors for the long-term tracking.

We also recorded three real face image sequences (each has 190 frames, about 8 seconds) include rotation about X- and Y-axes and a freewill rotation. Figure 6 shows some selected frames from the image sequences and the estimated results. We can see obvious errors in the fourth image. It is a good proof that DFU can effectively diminish the accumulative errors.

5. Conclusions

We have presented a model based method of head pose tracking from a monocular image sequence. We weight the depth and brightness constraints and use a dynamic feature updating technique to achieve long-term and reliable pose tracking. A generic head model can be used in future work, In addition, an affine motion assumption in 2D images also can be considered to make better constraint condition.

Acknowledgement

This work was sponsored in part by the National Science Foundation of China under grant no. 60175025.

References

[1] O. Faugeras, *Three-Dimensional Computer Vision,* MIT Press, Cambridge, MA.

[2] M. Harville, A. Rahimi, T. Darrell, G. Gordon and J.Woodfill, “3D Pose Tracking with Linear Depth and Brightmess Constraints”, *Proc. IEEE 7th Int. Conf. on Computer Vision,* Vol.1, pp.206-213, 1999.

[3] T. Hogg, D. Rees and H. Talhami, “Three-dimensional pose from two-dimensional images: a novel approach using synergetic networks”, *Proc. IEEE Int. Conf. on Neural Networks,* Vol.2, pp.1140-1144, 1995.

[4] V. Lepetit, J. Pilet and P. Fua, “Point matching as a classification problem for fast and robust object pose estimation”, *Proc. the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol.2, pp.244-250, 2004

[5] G. Liang, H. Zha and H. Liu, “Affine Correspondence Based Head Pose Estimation for a Sequence of Images by Using a 3D Model”, *Proc. of IEEE 6th Int. Conf. on Automatic Face and Gesture Recognition*，pp.632-637, 2004

[6] S. Ma and Z. Zhang, *Computer Vision*, Science Press, pp. 97-98, 1998.(in Chinese)

[7] L. Quan and Z. Lan, “Linear N⩾4-point pose determination”, *Proc. IEEE 6th Int. Conf. on Computer Vision,* pp 778-783, 1998.

[8] J. Shi and C. Tomasi, “Good Features to Track”, *Proc. IEEE. Int. Conf. on Computer Vision and Pattern Recognition*, pp. 593 –600, 1994.

[9] Y. Wei, L. Fradet and T. Tan, “Head pose estimation using Gabor eigenspace modeling”, *Proc. Int. Conf. on Image Processing*, Vol. 1, pp. 22-25, 2002.

[10] J. Xiao, T. Kanade, and J.F. Cohn “Robust Full-Motion Recovery of Head by Dynamic Templates and Re-registration Techniques”, *Proc. IEEE. Intl. Conf. on Automatic Face and Gesture Recognition*, pp. 593-600, 2002.

[11] J. Yao and W. Cham, “Efficient Model-based Linear Head Motion Recovery from Movies”, *Proc. the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol.2,pp.414-421, 2004

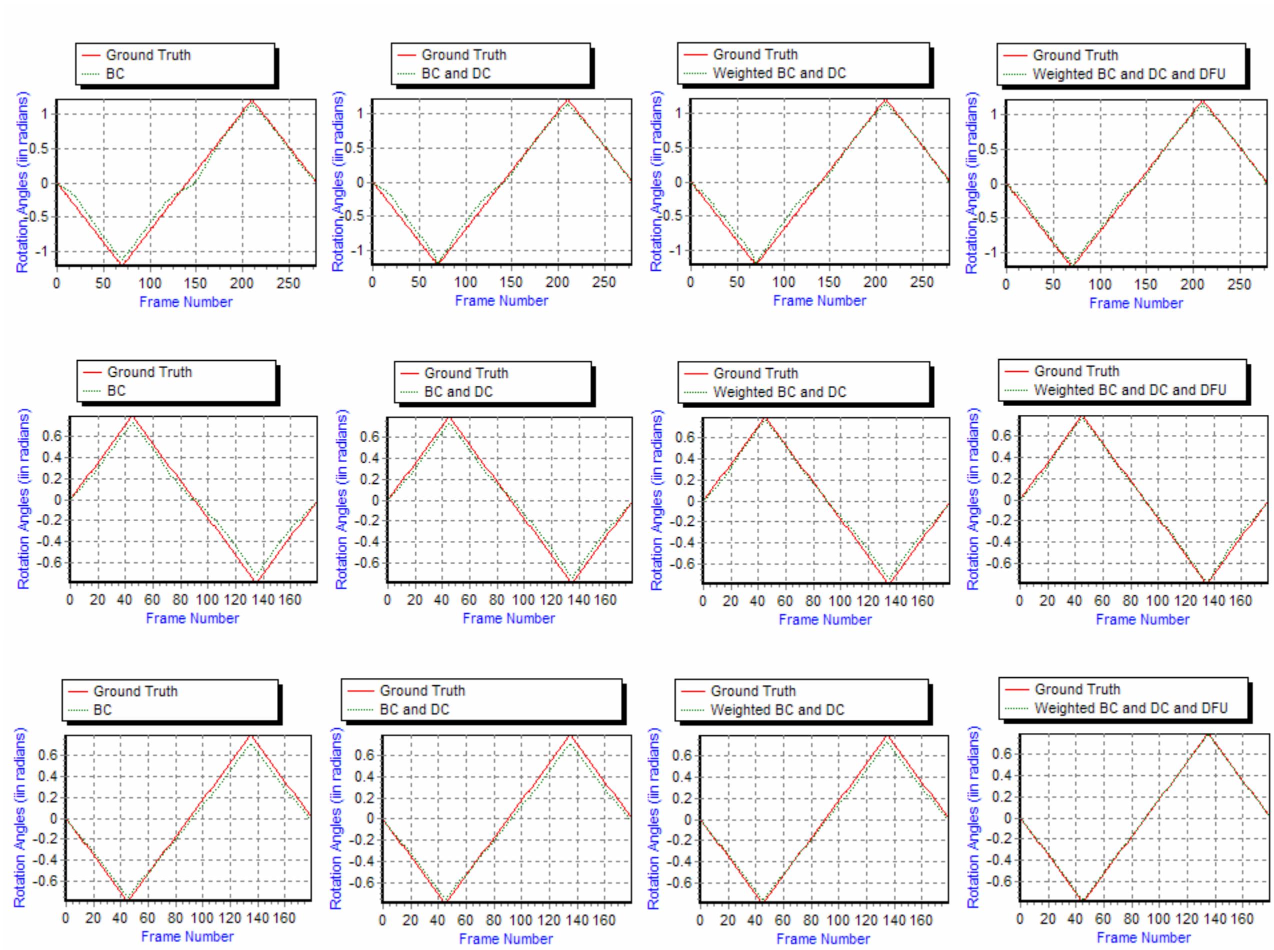

Fig. 5 Comparison of ground truth (solid line) with the computed rotation angles (in radians) around Y-axis (first row), X-axis (second row), and Z-axis (third row) separately. At each row, Left: BC only. Middle-left: BC and DC. Middle-right: Weighted BC and DC. Right: Weighted BC, DC and DFU. (BC: Brightness Constraint. DC: Depth Constraint. DFU: Dynamic Feature Updating.)

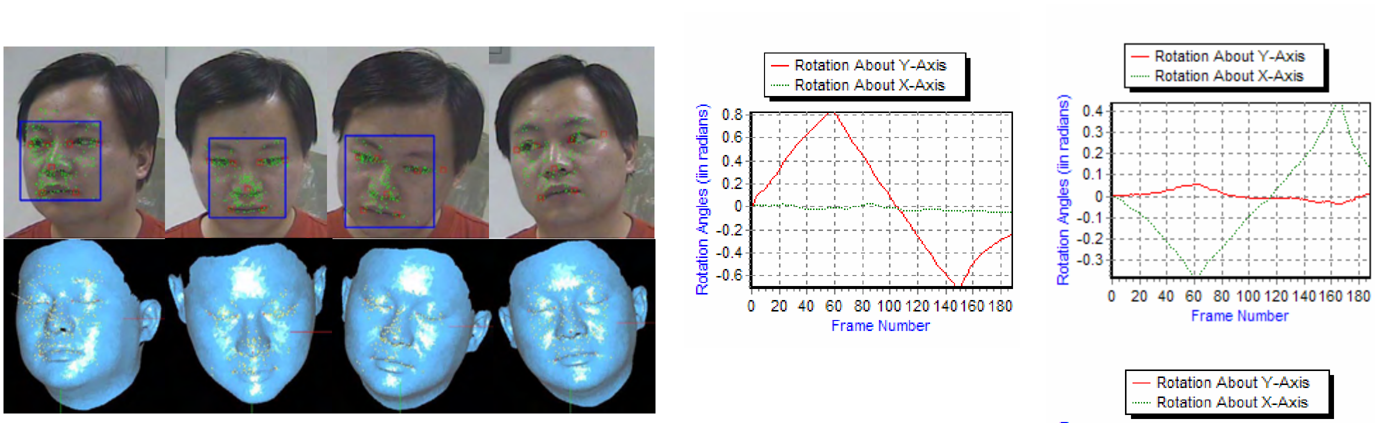

Fig.6 Left upper row: Frame 90 in sequence 1, frame 145 in sequence 2, frame 187 in sequence 3, frame 122 in sequence 3. Left bottom row: Corresponding pose estimated results of the images in first row. The three images from the left are estimated by our method, the fourth by BC and DC. Right: Computed rotation angles for three image sequences with rotation around Y-axis, X-axis and the free will rotation (from left to right).