Robust 3D Head Tracking Using Camera Pose Estimation

Shay Ohayon and Ehud Rivlin
Computer Science Department
Israel Institute of Technology
Haifa, Israel 32000

Abstract

In this paper we present a robust method to recover 3D position and orientation (pose) of a moving head using a single stationary camera. Head pose is recovered via a camera pose estimation formulation. 3D feature points (artificial or natural occurring) are acquired from the head prior to tracking and used as a model. Pose is estimated by solving a robust version of "Perspective n Point" problem. The proposed algorithm can handle self occlusions, outliers and recover from tracking failures. Results were validated by simultaneous tracking using our system and an accurate magnetic field 3D measuring device. Our contribution is a system that is not restricted to track only human heads, and is accurate enough to be used as a measuring device. To demonstrate the applicability of our method, three types of heads (human, barn owl, chameleon) were tracked in a series of biological experiments.

1. Introduction

An accurate estimation of head position and orientation (pose) in 3D is important in many applications. Knowledge about gaze direction can be used in human-computer interfaces, video compression and face recognition systems. Furthermore, possible applications are not limited solely to computer science domains. Our motivation is driven by many biological experiments, in which a human observer needs to monitor an experiment and operate accordingly. Such experiments may be fully automated if information about the animal's gaze was available. Reporting all 6 degrees of freedom (DOF) of the head is important since it enables analysis of movements in the head intrinsic coordinate system, revealing preferred directions of movement.

In this paper, we present a robust approach for tracking a head in video sequences. While previous approaches give only a rough estimate to head pose, and use complex representations such as a triangular mesh, a textured cylinder [2, 1] or superquadric [7], we represent the head as a sparse set of 3D points. This set is acquired prior to tracking and used to model head geometry. During tracking, a camera pose estimation problem is solved using the known model points and the available visible features. Since the camera is static, the recovered camera pose relative to the 3D reference points is equivalent to head pose. The point representation enables an accurate recovery of head pose and the tracking of a head with an arbitrary geometry (not necessarily human), as will be shown subsequently.

The problem of external camera pose estimation relative to a 3D set of reference points was addressed by many researchers [5, 4, 11]. Most of the proposed methods assume that the 3D position of several points, relative to a fixed reference frame, is given. The "Perspective n Point" (PnP) problem [11] is an exception, since it uses relative distances between 3D features to estimate pose. This attribute renders the method especially suitable for head tracking since relative distances between 3D points on the head are invariant under rigid body transformation. Furthermore, face deformation can change the 3D spatial configuration of features only by a limited amount.

System Overview: Prior to tracking, a set of 3D points is acquired from the head. This set will be referred as the acquired model. The actual tracking sequence is acquired using a single stationary camera. 2D image features are extracted from this sequence according to their spatial characteristics and predicted locations. The 2D features correspond to 3D points on the moving head. The distances from camera center to the 3D head points are recovered by solving a robust version of the PnP problem. The 3D head points represent the recovered model. Head pose is determined by finding the best rigid body transformation which maps the points from the acquired model to the recovered model (absolute orientation problem). Lost 2D image features are recovered by projecting their 3D position onto the image using the recovered head pose. In the following sections, a detailed description of each module of our system will be given.

Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)
0-7695-2521-0/06 $20.00 © 2006 IEEE

2 Model Acquisition

While previous approaches use complex surface representations to model head geometry (often, difficult and time consuming to acquire), our representation is sparse and easy to obtain. Only a few 3D points (typically $N \leq 20$) are used to model the head. The 3D points are acquired using a single camera which views the head from different angles. Multiple views are used to recover the 3D position of model points relative to a checkerboard pattern [5]. The acquired points are then transformed to a head centered coordinate frame, such that $[0, 0, 0]$ represents the head center. Points can also be acquired using a stereo system or any other 3D measuring device. We will denote the acquired model M, represented in a head centered coordinate frame, as the set of 3D points $P_M^u = [X_M^u, Y_M^u, Z_M^u]$, $1 \leq u \leq N$.

3 Head Pose Estimation

In this section, we describe the process of estimating the head pose. The 3D model points P_M^u, which are represented in a head centered coordinate frame, are searched in the image according to their 2D spatial appearance (chroma or any other cue). Once the 2D image features (p_u) are matched to 3D model points, the distances from camera center to the 3D head points are recovered by solving the a robust version of the PnP problem. Finally, the coordinates of the recovered 3D head points $P_R^u = [X_R^u, Y_R^u, Z_R^u]$ are matched to the 3D model points P_M^u and the best rigid body transformation, which maps between P_M^u and P_R^u is found by solving the absolute orientation problem. This yields the rotation R and translation T that represents the head pose in the current frame.

3.1 PnP and Recovered 3D Head Points

The distances from camera center to the 3D head points are recovered by solving a robust version of the PnP problem, which uses the 2D image features and the known 3D acquired model. In the following, we review the problem and present our robust solution to it.

The relative Euclidean distance between two model feature points u and v is given by

$$d_{u,v} = \left\| P_M^u - P_M^v \right\|. \tag{1}$$

Camera center is denoted by C. According to the cosine theorem

$$\begin{aligned} &\left\| P_R^u - P_R^v \right\|^2 = \\ &\left\| P_R^u \right\|^2 + \left\| P_R^v \right\|^2 - 2 \left\| P_R^u \right\| \left\| P_R^v \right\| \cos \angle P_R^u C P_R^v. \end{aligned} \tag{2}$$

Left hand side of equation 2, can be expressed in terms of the acquired model

$$\left\| P_R^u - P_R^v \right\| = \left\| P_M^u - P_M^v \right\| = d_{u,v}. \tag{3}$$

We assume that camera internal parameters are known and represented as a 3x3 matrix K [8]. The direction vector, from camera center C to feature u is given by:

$$N(u) \triangleq K^{-1} \left[p_u, 1 \right]^T. \tag{4}$$

Therefore, the angle $\angle P_R^u C P_R^v$ can be found from the two direction vectors $N(u)$ and $N(v)$:

$$\cos \angle P_R^u C P_R^v = \frac{N(u)\, N(v)}{\left\| N(u) \right\| \left\| N(v) \right\|}. \tag{5}$$

When three features (u, v, w) are available, the cosine theorem (equation 2) can be formulated for pairs $(u, v), (u, w), (v, w)$. Three quadratic equations in three unknowns $(\left\| P_R^u \right\|, \left\| P_R^v \right\|, \left\| P_R^w \right\|)$ are obtained. Using Sylvester resultant ([8]), the set of three equations are reduced to a single forth degree polynomial: $g(x) = \sum_{i=0}^{4} a_i x^i$, such that $g(x_u) = 0$ and $x_u \equiv \left\| P_R^u \right\|^2$. The root x_u of the computed polynomial corresponds to the squared distance from camera center to the 3D point ($\left\| P_R^u \right\|^2$). However, this polynomial can have up to four different real roots. Thus, three features are not enough to obtain a unique solution. Additional information is needed, such as a fourth point to obtain an unambiguous solution. When $N \geq 4$ un-occluded points are available, then $\lambda \equiv \binom{N-1}{2}$ forth degree polynomials can be obtained. The unique solution is the common root of all these polynomials. When exact measurements are not available due to noise, it is difficult to determine the common root since noisy polynomial coefficients result in a drift of the roots position.

Quan and Lan proposed in [11] a linear solution which stacks the λ forth degree polynomial coefficients into a single matrix. Their solution was obtained using the SVD of the coefficient matrix. There are two drawbacks of this approach. It is not stable when outliers are present and a positive solution is not guaranteed. Inspired by [10], we use robust M-estimator ρ, such as Huber, to overcome the above difficulties. A robust estimation of x_u is obtained by minimizing

$$x_u = \arg\min \sum_{j=1}^{\lambda} \rho \left(\sum_{i=0}^{4} a_i^j x^i \right). \tag{6}$$

This problem is non convex and has multiple local minima. Therefore, an iterative method (Gauss-Newton) is applied with Quan and Lan's [11] linear solution as initial guess.

Once x_u is obtained, the 3D coordinate of P_R^u is:

$$P_R^u = \frac{N(u)}{\left\| N(u) \right\|} \sqrt{x_u} = \frac{N(u)}{\left\| N(u) \right\|} \left\| P_R^u \right\|. \tag{7}$$

This process is repeated for each point $1 \leq u \leq N$, until all visible points of the recovered model are obtained.

3.2 Absolute Orientation and Head Pose

Head pose is represented as the rigid body transformation which rotates (R) and translates (T) the acquired model, to align with the recovered model. The two models are connected via the following equation:

$$P_R^u = RP_M^u + T. \qquad (8)$$

R and T are found by solving this absolute orientation problem [3]. A linear solution can be obtain if R is represented using quaternions. Nevertheless, a more accurate solution can be obtained by projecting the 3D recovered model to the image and minimizing the re-projection error:

$$\min_{R,T} \sum_u \rho\, [p_u - \Phi_{R,T}\,(P_M^u)], \qquad (9)$$

where $\phi_{R,T}$ is the perspective projection operator and ρ is a robust M-Estimator. The linear solution is used as an initial guess to the optimization equation 9.

4 Implementation details

The pose estimation algorithm described in the previous section uses low level routines to search the 2D image features. Features are hand marked in the first frame, and are automatically tracked from thereon. The system can find the pose even with 3 points, since we can select the most probably solution among the four possible ones, by comparing it to the previous found pose. When all features are lost, the user have to initialize the tracking module. However, when natural occurring features are used, there is no need, since each feature has a large properties vector which identifies it almost uniquely.

To reduce the search space of possible features position, the poses from previous frames are extrapolated and used to obtain a predicted locations of 2D image features in the current image [8]. The predicted 2D locations are obtained by projecting the 3D model points using the extrapolated pose. Therefore, even occluded features can be recovered once they become visible, since their 3D position is known. When a lost feature is detected near the predicted position, it is considered a candidate. Only candidates which reduce the re-projection error (equation 9) are further tracked.

In our experiments we used either artificial markers or natural occurring features. Artificial markers were detected according to their hue, saturation and shape properties. Natural occurring features were found by searching for a significant local maxima in scale-space. Such features can be matched robustly since additional local information from the neighborhood of each feature is saved. The SIFT descriptors [6] were used for detection and matching.

When artificial features with same spatial appearance are used, ambiguities in correspondence may be encountered when two tracked features get too close to each other. The two features ”merge” into a single spatial position due to perspective distortion. Our algorithm identifies these merge events and ignores the problematic features when pose is solved. Then, ambiguities are solved by choosing the correspondence combination which minimizes the re-projection error. We typically encountered no more than 4 merge events at the same time.

5 Experimental Results

Three types of heads were successfully tracked in a series of experiments. Video sequences resolution was 720x576 pixels, but was reduced to improve running time. Head pose was obtained at a frame rate of approximately 3-4 fps with a Pentium 1.4Ghz. However, when merge events were detected, performance dropped dramatically since ambiguities had to be resolved.

Due to space limitation, we will only give a brief description of the experiments. The interested reader can refer to [8] and [9] for more details. The head of a barn owl was tracked to investigate the strange head movements owl make prior to prey capture. These pre-attack head movements were successfully tracked and analyzed with our system in a recent biological study [9]. High pose accuracy was obtained with the usage of blue circular artificial markers. These could easily be detected using their hue,saturation and shape properties. More than 30 video sequences with an average length of 27 ± 16 s, were successfully analyzed. Although many of the sequences contained missing features due to self occlusions, our system successfully managed to track the head (Fig 2, top). Features positions are marked in yellow dot. Notice that their 2D position is known even if they are occluded.

In another series of experiments conducted by out colleague Ofir Avni, both eyes and head of a chameleon were tracked (additional module was written to determine the pupil’s position). The goal was to measure correlations between independent eye movements, relative to the head pose. Several artificial rectangular markers were attached to the head and were used as markers. Several minute long sequence was successfully analyzed with our system (Fig. 2, middle). The direction of the eye is drawn as a yellow cone in the left most picture.

To demonstrate that head pose can be tracked using natural occurring features we tracked a human head without any attached markers. More than 100 SIFT descriptors were acquired and stored as a model, along with their 3D relative distance. However, only 5-15 could be reliably matched during the actual tracked sequence (Fig. 2, bottom). The detected SIFT descriptors are marked in yellow dots. The recovered pose was less accurate than the one obtained in the two experiments described above. It is more difficult

Figure 1. Comparison with miniBIRD.

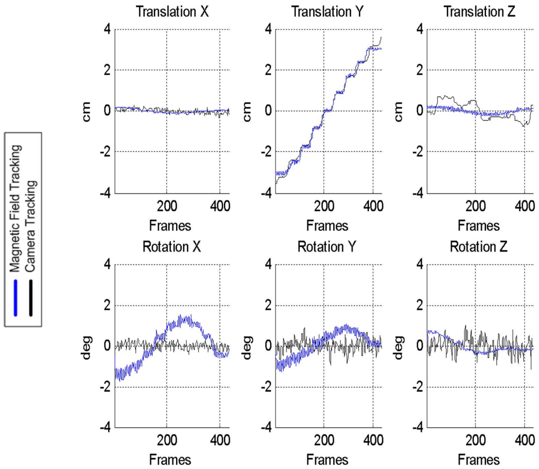
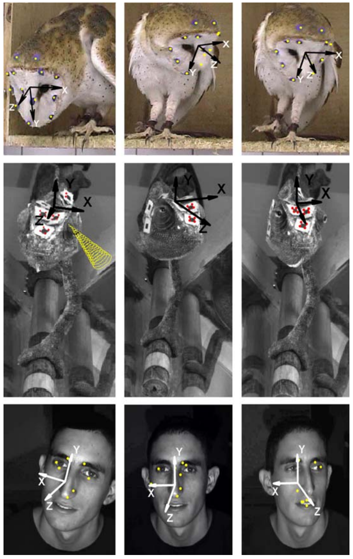


Figure 2. Tracking results.



to asses the exact accuracy, since no ground truth measurements were available (see next section).

6 Pose Accuracy Analysis

Accuracy was analyzed by simulations. However such simulations can be influenced by many factors, such as the shape of the tracked object, angle of view, etc. Therefore, the tracking results of a real calibrating object were compared to ground truth measurements. 18 Images with the calibrating object at various poses were analyzed. Results indicated that our system has a standard deviation of 2.3mm in position and $0.47°$ in orientation. Further validation of our results was obtained by a comparison with one of the commercially available 3D magnetic field measuring devices (miniBIRD - Ascension Technologies). This highly accurate (1.8mm in position, $0.5°$ orientation) system has a small wired sensor which measures the changes in the magnetic field induced by a near-by transmitter. Simultaneous recording from our system and miniBIRD showed comparable performance (Fig. 1).

Acknowledgments: The authors would like to thank Ofir Avni for supplying the results from his experiments with the chameleons.

References

[1] L. Brown. 3d head tracking using motion adaptive texture-mapping. In *IEEE Conference in Computer Vision and Pattern Recognition*, pages 998–1003, 2001.

[2] M. L. Cascia, J. Isidoro, and S. Sclaroff. Head tracking via robust registration in texture map images. In *IEEE Conference in Computer Vision and Pattern Recognition*, pages 508–514, 1998.

[3] O. Faugeras and M. Hebert. The representation, recognition, and locating of 3-d objects. *The International Journal of Robotics Research*, 5(3):27–52, 1986.

[4] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[5] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2000.

[6] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Jounral of Computer Vision*, 60(2):91–110, 2004.

[7] M. Malciu and F. J. Prêteux. A robust model-based approach for 3d head tracking in video sequences. In *Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 169–175, 2000.

[8] S. Ohayon and E. Rivlin. Robust 3d head tracking using camera pose estimation. Technical Report CS-2006-12, Computer Science Department, Technion, 2006.

[9] S. Ohayon, R. F. van der Willigen, H. Wagner, I. Katsman, and E. Rivlin. On the barn owl's visual pre-attack behavior: I. structure of head movements and motion patterns. *Journal of Comparative Physiology A*, 192(7), 2006.

[10] J. Park, B. Jiang, and U. Neumann. Vision-based pose computation: Robust and accurate augmented reality tracking. In *IWAR99*, page 3, 1999.

[11] L. Quan and Z.-D. Lan. Linear n-point camera pose determination. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 21, pages 774–780, 1999.