Bounding Errors for Improved 3D Face Tracking in Visual Interfaces

Jochen Heinzmann and Alexander Zelinsky
Robotic Systems Lab
Research School of Information Sciences and Engineering
The Australian National University, Canberra ACT 0200
[jochen|alex]@syseng.anu.edu.au

Abstract

This paper describes recent results in a human-robot interaction project. The aim of the project is to develop human-friendly robots, in particular control systems and user interfaces for such systems. In this paper we present an error analysis of the visual face tracking system that has been developed as an interface appropriate for natural interaction between a robot and an operator. The system utilizes a monocular camera and a hardware vision system to identify features of a face and estimate the 3D pose of the head.

1 Introduction

If robot technology is to be introduced into the everyday human world, the technology must not only operate efficiently executing complex tasks such house cleaning or putting out the garbage, the technology must be safe and easy for people to use. This requires a human-machine interface that lets people interact with the robot in a natural way. People naturally express themselves through language, facial gestures and expressions. Speech recognition in controlled situations using a limited vocabulary with minimal background noise interference is now possible and could be included in human-robot interfaces. However, vision-based human-computer interfaces have only recently begun to attract considerable attention. With a visual interface a robot could recognize facial gestures such as "yes" or "no", as well being able to determine the user's gaze point i.e. where the person is looking. The ability to estimate a person's gaze point is most important for a human-friendly robot. For example a robot assisting the disabled may need to pick up items that attract the user's gaze. The face tracking system described in this paper will be connected to a Barrett WAM (Whole Arm Manipulator) robot arm. In a desktop configuration a person will be able to select a particular object on the table with his/her gaze and command the robot to pick it up and hand it over. Facial gestures will be used for error correction, such as a nod to confirm the selection of the correct object or shaking the head to indicate that the wrong object was selected. In this paper we report an accuracy evaluation of a real-time vision based interface that tracks a persons head with a monocular camera and estimates the 3D pose.

2 3D Pose Recovery

The recovery of the pose of objects is one of the fundamental algorithms in computer vision. In particular for rigid objects a variety of solutions has been proposed. Methods that only use as many object points as necessary to restrict the problem in a way that yields only one accurate solution are widely used. Over constraining the problem by including additional object points usually leads to computationally expensive least squares minimization problems. See [Koller *et al.*, 1993][Lowe, 1991] for iterative solutions and [Arun *et al.*, 1987][Horn, 1987] for closed form solutions. Since the spatial position of three object features describe the pose of a rigid object in space, a variety of algorithms have been developed using feature triplets. There are two major classes of solutions, those assuming affine projection and those assuming perspective projection of the object points. The perspective projection is the most accurate model however systematic errors are introduced by differences between the camera lens system and the assumption of a pinhole camera. [Haralick *et al.*, 1991] gives an overview of the algorithms assuming perspective projection.

The affine projection has various advantages, as the calculations are significantly simpler and therefore better suited for real-time applications. The focal length does not need to be known if only the rotational aspect of the pose is of interest. Instead of the four solutions produced by perspective projection solutions the affine projection has only two solutions for a given projection of a feature triplet. In many cases the use of the affine projection can be justified by the large distance between the object and the camera relative to the minor differences

in depth of the features. In such cases the difference between the solutions using the affine projection model and the perspective model are negligible. As a rule of the thumb the ratio between the distance of the object from the camera and the depth differences of the individual features should be at least 10:1 [Costall, 1993][Gee and Cipolla, 1994][Thompson and Mundy, 1987].

Algorithms proposed to solve the three-point to 3D-pose problem include Ullman's [Ullman, 1986], Huttenlocher and Ullman's [Huttenlocher and Ullman, 1990], Grimson, Huttenlocher and Alter's [Grimson *et al.*, 1992], Alter's [Alter, 92] and Cygnaski and Orr's [Cyganski and Orr, 1988]. Alter [Alter, 92] described a brief analysis of the numerical stability of his algorithm. Grimson et al [Grimson *et al.*, 1992] made an error analysis of the Huttenlocher and Ullman algorithm, which we also consider. Grimson et al derived bounds for the error in the final result bounded by an ϵ-circle around the precise position. The calculation of the error propagation through the algorithms is complex and only rough overestimates can be derived. Also, Grimson et al made no attempt to derive the systematic error introduced by assuming affine projection nor did they try to improve the accuracy of the algorithm.

For our face tracking application [Zelinsky and Heinzmann, 1998] we chose the algorithm proposed by Huttenlocher et al [Huttenlocher and Ullman, 1990] for the pose recovery since it only produces the two real solutions of the pose recovery problem, in contrast to the original formulation of Ullman [Ullman, 1986]. In the following section we analyze the systematic error as well as the sensitivity of the estimate to tracking errors. Also we derive an extension using the error analysis to decrease the systematic error. This extension was implemented and experimentally evaluated. Results are presented in this paper.

3 Systematic Error

Our pose estimation algorithm assumes an affine projection of the model points into the image plane. In reality, the features are projected into the image plane under perspective projection. The systematic error caused by this discrepancy is analyzed in this section. We also propose an improvement to the feature triplet pose recovery algorithm that is able to reduce the systematic rotational error of the pose estimate by 75% and the maximum translational error by 90%.

The well known formulas for pose recovery are stated for completeness, for a description of the underlying geometry refer to [Huttenlocher and Ullman, 1990][Alter, 92].

$$\mathbf{L} = \mathbf{N}_2 \cdot \mathbf{M}_2^{-1} \tag{1}$$

where $\mathbf{N}_2$ $\mathbb{R}^{2\times 2}$ contains the image positions $\mathbf{n}_2$ and $\mathbf{n}_3$ in it's columns and $\mathbf{M}_2$ $\in\mathbb{R}^{2\times 2}$ contains the x- and y-components of the model points in it's columns. In the next step a digression is made to consider how two orthogonal and equal length vectors $\mathbf{u}_{1,2}$ are transformed. Choosing $\mathbf{u}_{1,2}$ $\in\mathbb{R}^2$ to be (1,0) and (0,1), k_1 and k_2 become

$$k_1 = -\begin{bmatrix} l_{11} \\ l_{21} \end{bmatrix} \cdot \begin{bmatrix} l_{12} \\ l_{22} \end{bmatrix} \tag{2}$$

$$k_2 = \left\| \begin{bmatrix} l_{11} \\ l_{21} \end{bmatrix} \right\| \cdot \left\| \begin{bmatrix} l_{12} \\ l_{22} \end{bmatrix} \right\| \tag{3}$$

$$c_1 = \pm\sqrt{\frac{1}{2}\left(k_2 + \sqrt{k_2^2 + 4k_1^2}\right)} \tag{4}$$

$$c_2 = \begin{cases} \frac{k_1}{c_1} & c_1 \neq 0 \\ \pm\sqrt{-k_2} & c_1 = 0 \end{cases} \tag{5}$$

$$s = \sqrt{l_{11}^2 + l_{21}^2 + c_1^2} = \sqrt{l_{12}^2 + l_{22}^2 + c_2^2} \tag{6}$$

$$s\mathbf{R} = \begin{bmatrix} l_{11} & l_{12} & \frac{1}{s}(c_2 l_{21} - c_1 l_{22}) \\ l_{21} & l_{22} & \frac{1}{s}(c_1 l_{12} - c_2 l_{11}) \\ c_1 & c_2 & \frac{1}{s}(l_{11} l_{22} - l_{12} l_{21}) \end{bmatrix} \tag{7}$$

As a metric for the rotational error between two spatial orientations we use the angle in the equivalent angle-axis representation as defined by Craig [Craig, 1986]. We define the rotational distance metric $\Delta(\mathbf{P}_1, \mathbf{P}_2)$ as the angle in this representation:

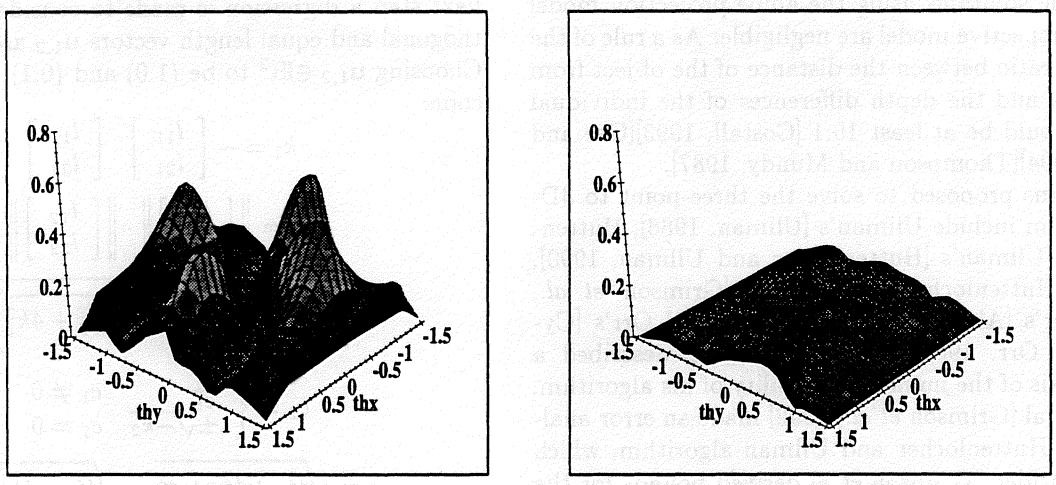
$$\Delta(\mathbf{P}_1, \mathbf{P}_2) = \arccos\left(\frac{\mathbf{P}'_{1,1} + \mathbf{P}'_{2,2} + \mathbf{P}'_{3,3} - 1}{2}\right) \tag{8}$$

where $\mathbf{P}' = \mathbf{P}_1^{-1} \cdot \mathbf{P}_2$ is the transformation from $\mathbf{P}_2$ to $\mathbf{P}_1$. The rotational difference between the true orientation $\mathbf{P}_1$ and the measured orientation $\mathbf{P}_2$ is plotted over the angles θ_x and θ_y of the rotations about the x- and y-axes in Figure 1a using the original Huttenlocher and Ullman algorithm. The calculations are based on the model and distances shown in Figure 3 which resembles the situation in our face tracking application. The angle θ_z of the rotation about the z-axis equals 0.

The rotational error varies strongly over the visible range of the triplet plane. The four big humps are generated by strong errors in the estimation of the third elements of the first two column vectors $\mathbf{p}_1 = \frac{1}{s}[l_{11}, l_{21}, c_1]^t$ and $\mathbf{p}_2 = \frac{1}{s}[l_{12}, l_{22}, c_2]^t$ of the rotation matrix $\mathbf{M}_2$, corresponding to the z-component of the new x- and y-axes.

If the image coordinates of the features are generated by an affine projection, the first two column vectors $\mathbf{p}_{1,2}$ have equal length. Thus, s scales both $\mathbf{p}_1$ and $\mathbf{p}_2$ to length 1. However, feeding the algorithm with feature coordinates generated by a perspective projection, the length of the first two column vectors is generally different, corresponding to two different depth estimates that can be derived from the two columns.

$$s_1 = \sqrt{l_{11}^2 + l_{21}^2 + c_1^2} \neq \sqrt{l_{12}^2 + l_{22}^2 + c_2^2} = s_2 \tag{9}$$



a) b)

Figure 1: The systematic rotational error in the pose recovery plotted over the rotational angles θ_x and θ_y about the x- and y-axes. Plot a) shows the error for the original algorithm, plot b) for the improved version.

The correct value $\tilde{s}$ lies between s_1 and s_2. Choosing $s = \max(s_1, s_2)$ as the normalization and depth estimation parameter in Equation 7 can reduce the maximum error in the depth estimate to about 10% of the original maximum error.

Simply scaling the unused column $\mathbf{p}_j$ to length 1 generates a valid rotation matrix. Obtaining a more accurate scaling value s gives rise to a more accurate estimate of the z-component of $\mathbf{p}_j$. This allows us to reduce the systematic rotational error by 75%. Since the first two elements of that vector and the scaling are already determined, only the third element of $\mathbf{p}_j$, c_j, can be recomputed such that $\mathbf{p}_j$ has length 1. This recalculation changes the orientation of the vector, thus, $\mathbf{p}_i$ and $\mathbf{p}_j$ are no longer orthogonal. The third element of $\mathbf{p}_i$ has to be recomputed such that $\mathbf{p}_i$ and $\mathbf{p}_j$ are orthogonal, and then $\mathbf{p}_i$ can be scaled to length 1. The third column $\mathbf{p}_3$ of the rotational matrix $\mathbf{R}$ is generated according to the original algorithm to complete the matrix.

The extensions to the original algorithm are outlined below:

1. Calculate $\mathbf{L}$, k_1, k_2, c_1 and c_2 according to the formulas 1-5.
2. Calculate the length of both columns $s_1 = \sqrt{l_{11}^2 + l_{21}^2 + c_1^2}$ and $s_2 = \sqrt{l_{12}^2 + l_{22}^2 + c_2^2}$
3. Let i, j be indices such that $s_i \geq s_j$. Then the scaling factor can be determined from the length of the longer vector $s = s_i$.
4. Recalculate c_j to make column $\frac{1}{s}\mathbf{p}_j$ to be length 1 preserving the sign of c_j.
5. Recalculate c_i so that $\mathbf{p}_i \perp \mathbf{p}_j(R)$ and rescale $\mathbf{p}_i$ so that $\frac{1}{s}\mathbf{p}_i$ is of length 1;
6. Calculate the third column or $\mathbf{R}$ using Formula 7.

The resulting rotational error of the rotation matrix obtained from the improved algorithm is shown in Figure 1b. The extensions to the algorithm reduce the rotational error to about 25% of the error obtained from the original algorithm.

4 Tracking Error Sensitivity

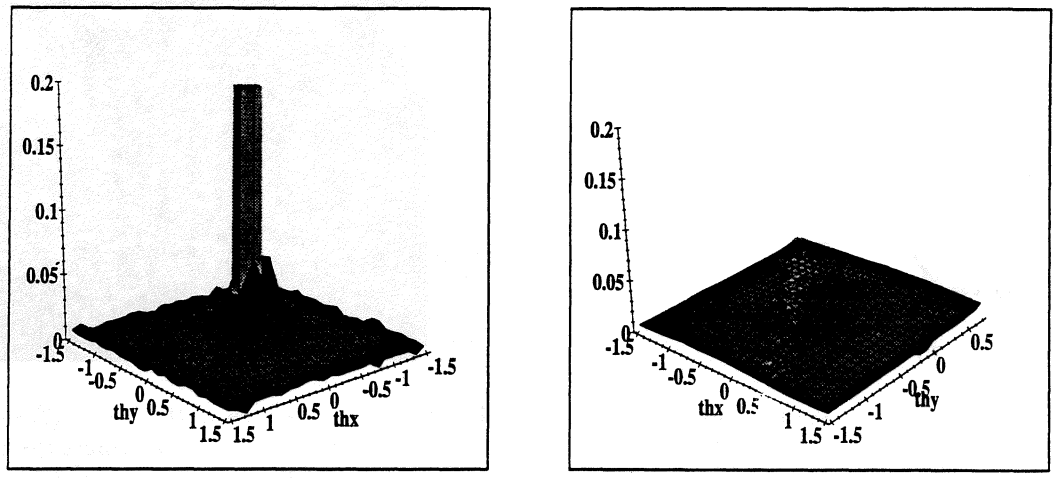
In control theory sensitivity is a widely used term that describes the change in a parameter a as result of changes in a parameter b. The sensitivity function describes the fractional change in a, due to b, divided by the fractional change in a [Anand and Zmood, 1995]. We define the sensitivity ς_r in respect to the rotational part of the pose estimation to tracking errors for a particular pose $\mathbf{P}$ as the 2-norm of the vector of the differential quotients of the orientation distance function Δ.

$$\varsigma_r(\mathbf{P}) = \left\| \begin{bmatrix} \varsigma_3 \\ \varsigma_4 \\ \varsigma_5 \\ \varsigma_6 \end{bmatrix} \right\|_2 \tag{10}$$

$$\varsigma_i = \lim_{\epsilon \to 0} \frac{1}{\epsilon} \Delta(p(\mathbf{N}(\mathbf{P})), p(\mathbf{N}(\mathbf{P}) + \epsilon \mathbf{v_i})) \tag{11}$$

where $\mathbf{N}(\mathbf{P})$ is the vector of projected image positions, $p(\mathbf{N}(\mathbf{P}))$ is the recovered pose from this projection and $\mathbf{v_i}$ is the error vector for the ith element.

The rotational sensitivity ς_r of the model triangle over $[\theta_x, \theta_y]$ is plotted in Figure 2a. The spike in the center reaches up to infinity since the partial derivatives



a) b)

Figure 2: The sensitivity of the estimated pose over the rotations about the x- and y-axes. Plot a) shows the sensitivity for a single triangle and plot b) for the combination of two triangles

of the projected position of the features in respect to θ_x and $theta_y$ equals 0 at this point. The plot gives a good impression where the pose estimate will be robust and where noise with large standard deviation will occur. Figure 2b shows the reduced sensitivity which can be achieved by using multiple feature triplets.

5 Using Multiple Triplets

The pose recovery of a feature triplet under affine projection has a twofold ambiguity. Two poses $\mathbf{P}_1$ and $\mathbf{P}_2$ generate the same projection which correspond to a reflection about a plane parallel to the image plane. In our discussion of the systematic errors we assumed that we can always determine the correct solution $\hat{\mathbf{P}}$ for the pose. The ambiguity can be resolved by using 2 feature triplets in non-parallel planes as indicated in Figure 4. Both triplets generate two solutions, but only one solution is consistent with both triplets. This solution corresponds to the correct pose $\hat{\mathbf{P}}$ of the object. However, when using two triplet planes even without noise the two correct solutions in general do not perfectly match.

A straight forward solution to the problem is to weight the two almost matching solutions according to their sensitivity. The equivalent angle-axis formulation provides a mechanism to directly interpolate between two spatial orientations by rotating only a fraction of the angle around the equivalent axis K.

$$\mathbf{P} = EAR\left(\hat{\mathbf{P}}_1, \hat{\mathbf{P}}_2, \frac{\varsigma_r(\hat{\mathbf{P}}_2) \cdot \Delta(\mathbf{P}_1, \mathbf{P}_2)}{\varsigma_r(\hat{\mathbf{P}}_1) + \varsigma_r(\hat{\mathbf{P}}_2)}\right) \cdot \hat{\mathbf{P}}_1 \quad (12)$$

where $EAR(\hat{\mathbf{P}}_1, \hat{\mathbf{P}}_2, \alpha)$ is the rotational matrix around the equivalent axis $\mathbf{k}$ of $\hat{\mathbf{P}}_1$ and $\hat{\mathbf{P}}_2$ with angle α and $\varsigma_r(\mathbf{P})$ is the rotational sensitivity of $\mathbf{P}$.

However, by doing so the available information is not used in an optimal way. The sensitivity is a good measure of the overall error that can be expected from a particular pose estimate. But different components of the pose estimates have different individual sensitivities. Considering Figure 3, it is intuitively obvious that rotations about the z-axis can be detected very well in the frontal view, whereas rotations about the x- and y-axes prove to be difficult. By defining individual sensitivities for each of the rotational components allows us to synthesize a better spatial orientation $\mathbf{P}$ from multiple measurements which gives a lower overall sensitivity.

We define the metrics Δ_x, Δ_y and Δ_z for the rotations about the three axes as the absolute difference between their respective rotations in the spatial orientations $\mathbf{P}_1$ and $\mathbf{P}_2$.

$$\Delta_x(\mathbf{P}_1, \mathbf{P}_2) = |Rot_x(p(\mathbf{P}_1)) - Rot_x(p(\mathbf{P}_2))| \quad (13)$$

Δ_y and Δ_z are defined in a similar way.

Similarly to the metrics, we define the sensitivity functions ς_x, ς_y and ς_z for the individual components of the spatial transformation.

The individual components of the spatial transformation $\mathbf{P}$ describing the overall pose of the object can now be easily derived. Each rotation $Rot_{x,y,z}(\mathbf{P})$ about the three axes is the weighted average of the two respective rotations in $\mathbf{P}_1$ and $\mathbf{P}_2$ with the sensitivity as weights. Equation 14 gives the rotation around the x-axis, the formulas for the y- and z-axes can be similarly derived.

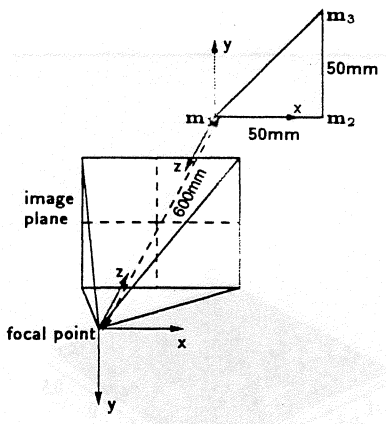


Figure 3: The geometry of the model triplet used for the accuracy and sensitivity evaluations.

$$Rot_x(\mathbf{P}) = \frac{\frac{Rot_x(\mathbf{P}_1)}{\varsigma_x(\mathbf{P}_1)} + \frac{Rot_x(\mathbf{P}_2)}{\varsigma_x(\mathbf{P}_2)}}{\frac{1}{\varsigma_x(\mathbf{P}_1)} + \frac{1}{\varsigma_x(\mathbf{P}_2)}} \quad (14)$$

The resulting sensitivity for the overall pose estimate using the original model triplet and an additional triplet of the same shape but rotated 45° around the y-axis. This situation is similar to our face tracking application we discuss in the next section. Figure 2b shows the sensitivity for estimate derived from the combined triplets. The spike in the center of the graph in Figure 2a has been eliminated by the combination of two triplets. The two remaining smaller humps represent the spikes of the individual triplets. When combining two triplets the angle between their spatial normals should be sufficiently large so that their individual sensitivity spikes are separated in the object orientation space since the sensitivity of the combined pose is at least as high as the lowest individual sensitivity.

6 Experimental Results

The visual feature tracking and pose estimation was evaluated using the MARITA-system (MAnnequin Robot for Investigation of Tracking Accuracy). MARITA consists of a mannequin head mounted onto a cable driven high performance pan-tilt-device. The two degrees of freedom can rotate the head around it's vertical axis as well as tilt the head for- and backward. Rotations about the z-axis (facial normal) are not possible, however, the error in estimating this fixed angle can still be measured. Also the transformation from the X-Y euler angles of the mechanism to the Z-Y-X euler angles used for the internal pose representation causes small rotations about the z-axis. MARITA allows us to position the mannequin head to an accuracy better than 0.1° and test the accuracy of the pose estimate of the face tracker under realistic conditions.

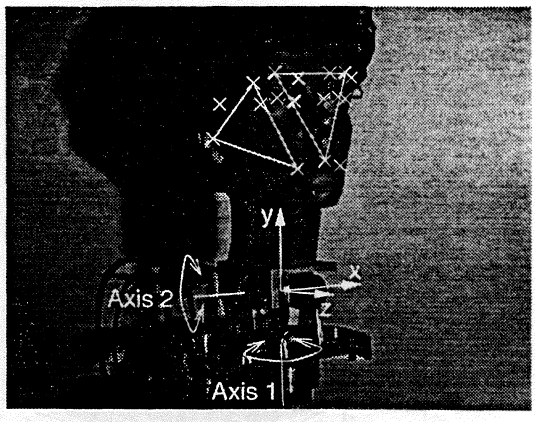


Figure 4: Outside view of the MARITA system with the rotational joints and head coordinate system. The white crosses indicate the feature positions, the triangles the triplet planes.

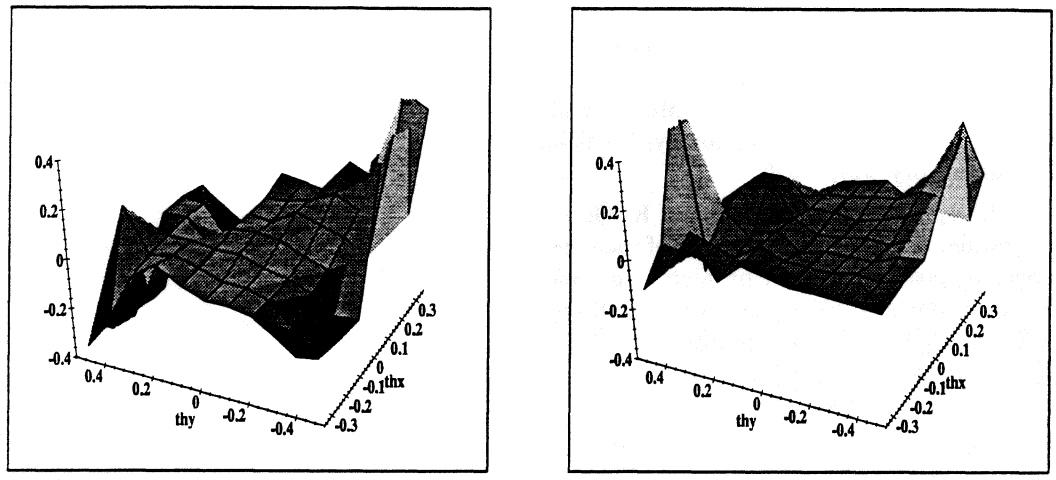
6.1 Features

To track the pose of the mannequin head a set of 19 features was selected manually, including 6 features per eye area, 3 features in the mouth area and 2 features in each ear area. The white crosses in Figure 4 denote the features used for tracking. Not all 19 features are used for pose estimation. Instead, only six features located at the corners of the triangles in Figure 4 are used to estimate the 3D pose. The other features are providing more clues for visual feature tracking which makes the overall tracking more stable and robust to noise and partial occlusions. With this selection of features and model triangles the spatial orientation of the head can be measured by at least two triplets for a wide range of configurations.

6.2 Tracking results

To measure the accuracy parameters of our face tracker we connect it directly to the controller of MARITA. A supervisory program moves the head into different position and then a sample of the pose estimators performance is taken over a period of 500 frames. As parameters for the performance the bias (mean error) between the correct pose and the measured pose as well as the standard deviation of this pose are measured. These parameters closely relate to the systematic error and the sensitivity. The points for evaluation are spaced 0.1rad apart, ranging from -0.5 to 0.5rad in y-rotation and -0.3 to 0.3rad in x-rotation. Overall 77 positions are evaluated.

Figure 5a and b show the mean absolute bias of the measured rotations about the y- and z-axes. The bias in most areas is close to 0 which indicates good performance of the pose estimator. On the edges of the plot where the head was rotated around the y-axis about 30°, the non-normalized template tracking failed and abrupt changes



a) b)

Figure 5: Performance measurements of the spatial orientation estimate: The bias of the measurements of the rotation a) about the y-axis (Rot_y) and b) about the z-axis (Rot_z).

in the magnitude of the error occur. The mean absolute biases were 0.06699rad (3.8°) in 75% of the area for the y-rotation and 0.04075rad (2.3°) in 74% for the rotation around the z-axis. The estimate of the rotation around the x-axis contains the highest error of the three axes because it is highly dependent on correct tracking of the features around the ear. Those features proved to be difficult to track due to the lack of unique texture. The mean absolute bias in the area where the features were tracked successfully (68%) is 0.10479rad (6.0°) for the x-axis. Overall, the mean absolute bias of the equivalent axis angle distance was 0.14098rad (8.1°) on 68% of the area.

7 Conclusion

We have presented a 3D pose estimation system from monocular image sequences based on the three feature triangulation algorithm by Huttenlochen and Ullman. To address the problem of systematic errors caused by measurements from perspective projections a improved algorithm has been proposed which reduces the systematic error in the examples by 75%. The problem of high sensitivity of certain components of the spatial transformation to tracking error has been overcome by using a sensitivity model and multiple triangles in different special planes to derive a robust estimate. The algorithm has been implemented in our face tracking application for performance evaluation. Over a wide range of spatial orientations the derived estimates are stable and provide reasonable accuracy for a monocular system. For large rotations the performance of the system is mainly limited by the ability of the template based feature tracking system to reliably localize the features. More recent vision hardware makes brightness normalized correlation functions available which overcomes this problem.

References

[Alter, 92] Tao Daniel Alter. 3d pose from three corresponding points under weak-perspective projection. Technical Report AIM-1378, MIT, AI Lab, July 92.

[Anand and Zmood, 1995] D K Anand and R B Zmood. *Introduction to Control Systems.* Butterworth-Heinemann Ltd, Oxford, 3 edition, 1995.

[Arun *et al.*, 1987] K. S. Arun, T. S. Huang, and S. D. Blostein. Least squares fitting of two 3d point sets. *IEEE Trans Pattern Analysis and Machine Intelligence*, 9(5):698–700, September 1987.

[Costall, 1993] A. Costall. A cubist manifesto for visual science. *Image and Vision Computing*, 11(6):334–341, 1993.

[Craig, 1986] John J. Craig. *Introduction to Robotics.* Addison Wesley, 2 edition, 1986.

[Cyganski and Orr, 1988] D. Cyganski and J. A. Orr. Object Recognition and Orientation Determination by Tensor Methods. *Advances in Computer Vision and Image Processing*, 3:101–144, 1988.

[Gee and Cipolla, 1994] A.H. Gee and R. Cipolla. Non-intrusive gaze tracking for human-computer interaction. In *Proceedings of the International Conference on Mechatronics and Machine Vision in Prac-*

tice, pages 112–117, Toowoomba, Australia, September 1994.

[Grimson *et al.*, 1992] W. Eric Grimson, Daniel P. Huttenlocher, and T. D. Alter. Recognizing 3D ojbects of 2D images: An error analysis. Technical Memo AIM-1362, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, July 1992.

[Haralick *et al.*, 1991] R. M. Haralick, C. Lee, K. Ottenberg, and M. Nolle. Analysis and solutions of the three point perspective pose estimation problem. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition ICPR*, pages 592–598, Los Alamitos, CA, USA, 1991. IEEE Computer Society Press.

[Horn, 1987] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternion. *Journal Optical Soc. Am.*, 4(4):629–642, 1987.

[Huttenlocher and Ullman, 1990] D. P. Huttenlocher and S. Ullman. Recognizing solid objects by alignment with an image. *International Journal of Computer Vision*, 5(2):195–212, November 1990.

[Koller *et al.*, 1993] D. Koller, K. Danilidis, and H.-H Nagel. Model-based object tracking in monocular image sequences of road traffic scenes. *International Journal of Computer Vision*, 10(3):257–281, 1993.

[Lowe, 1991] D.G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):441–450, 1991.

[Thompson and Mundy, 1987] D. W. Thompson and J. L. Mundy. Three-dimensional model matching from an unconstrained viewpoint. In *Proc. IEEE Robotics and Automation*, pages 208–220, April 1987.

[Ullman, 1986] S. Ullman. An approach to object recognition: Aligning pictoral descriptions. AI Memo 931, MIT, 1986.

[Zelinsky and Heinzmann, 1998] Alexander Zelinsky and Jochen Heinzmann. A novel visual interface for human-robot communication. *Advanced Robotics*, 11(8):827–852, 1998.