An Investigation of Model Bias in 3D Face Tracking

Douglas Fidaleo[1], Gérard Medioni[1], Pascal Fua[2] and Vincent Lepetit[2]

[1] Institute for Robotics and Intelligent Systems, University of Southern California
{dfidaleo|medioni}@usc.edu

[2] Computer Vision Laboratory, École Polytechnique Fédérale de Lausanne
{Pascal.Fua|Vincent.Lepetit}@epfl.ch

Abstract. 3D tracking of faces in video streams is a difficult problem that can be assisted with the use of a priori knowledge of the structure and appearance of the subject's face at predefined poses (keyframes). This paper provides an extensive analysis of a state-of-the-art keyframe-based tracker: quantitatively demonstrating the dependence of tracking performance on underlying mesh accuracy, number and coverage of reliably matched feature points, and initial keyframe alignment.

Tracking with a generic face mesh can introduce an erroneous bias that leads to degraded tracking performance when the subject's out-of-plane motion is far from the set of keyframes. To reduce this bias, we show how online refinement of a rough estimate of face geometry may be used to re-estimate the 3d keyframe features, thereby mitigating sensitivities to initial keyframe inaccuracies in pose and geometry. An in-depth analysis is performed on sequences of faces with synthesized rigid head motion. Subsequent trials on real video sequences demonstrate that tracking performance is more sensitive to initial model alignment and geometry errors when fewer feature points are matched and/or do not adequately span the face. The analysis suggests several indications for most effective 3D tracking of faces in real environments.

1   Introduction

3D tracking of faces in video streams is a difficult problem that can be assisted with the use of a priori knowledge of the structure and appearance of the subject's face at predefined poses. Tracking accuracy, however, is dependent (in part) upon the quality of this knowledge: ie, the underlying 3D accuracy and initial alignment of the tracking model in a selection of key image frames corresponding to the selected poses.

Unfortunately, for many tracking applications it is unreasonable to assume that a model of the tracked subject exists, or that sufficient views of the face are available a priori to optimally align the mesh. As shown in Figure 1, a single generic face is an unsatisfactory prior for all tracking subjects and single-view
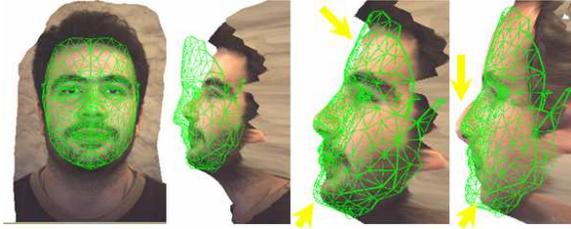
Fig. 1. (left) Improper registration of tracking mesh is not apparent from a single image. (right) Registration errors are dependent on subject's facial structure. With the first subject, errors are concentrated in forehead and chin area. The second subject has a more shallow chin and more pronounced nose bridge making these areas more difficult to align.

initialization can mask egregious registration errors. While a model of the subject may be created using global bundle adjustment as in [2], this is a lengthy offline process. Reasonable results at or near keyframes can be achieved with a relaxed 3D structure (ie, a generic face mesh) but as the subject deviates from the keyframe poses, tracking becomes sensitive to the initial pose alignment. Furthermore, even when accurate keyframe registration and geometry is available, 3D tracking from 2D features can be sensitive to the number and quality of matched features in each image.

The primary goal of this paper is to present a thorough experimental investigation of the tracking performance of a state-of-the-art 3D tracker applied to faces. We validate quantitatively the claims of tracking performance dependence on model accuracy by comparing performance with a variety of meshes on image sequences derived from real faces, but with synthetically generated motion whose parameters are precisely known. We show that it can be better to track with a much weaker prior such as an ellipsoid than to introduce a strong erroneous bias with a misaligned generic "face-like" mesh when optimal keyframe initialization is not possible. In both cases, the suboptimal mesh leads to degraded tracking results when the subject's pose is far from an in-plane translation of the keyframe when compared to an accurate 3D mesh. Additional factors contributing to tracking performance are also investigated, including the number of feature points accurately matched to the keyframe, the total face coverage of the points, and reprojection error.

We also demonstrate that by refining the geometry of the internal tracking model using initial estimates of camera pose, errors in both mesh geometry and alignment are reduced, and tracking performance is enhanced. Beginning with a rough estimate of face geometry we iteratively refine the model online using a simple stereo-based update approach and use the more accurate structure to re-estimate the 3d keyframe features.

The experiments on synthesized motion sequences extend directly to real sequences with the important caveat that due to variable image quality and resolution, the number of accurately matched features can be low. Further in-

vestigation on real sequences shows that these effects must be minimized not only for accurate but also stable tracking. The investigation concludes with a set of indications for effective 3D tracking of faces.

We have chosen to use the real-time tracker by [2] for our investigation due to the reported high quality performance, both in speed and accuracy.

2   Previous Work

In most rigid object tracking approaches the pose estimate at a given time is dependent on the estimate at the previous frame. Dubbed *recursive* tracking in [2], the concatenation of motion estimates causes error to be aggregated and can result in considerable tracking drift after several frames.

If the class of tracked objects is restricted (such as, to faces) a priori knowledge of the object properties can be leveraged to improve tracking accuracy and resolve pose ambiguities. 3D model-based tracking introduces this knowledge in the form of the structure, pose, and, in some cases, surface texture of the object. The 3D model is used to regularize feature motion in [6][8][5] [7][11][12].

To eliminate drift, keyframe approaches perform tracking by detection, utilizing information obtained offline such as the known pose of the head in specific frames (keyframes) of the tracking sequence. Input images are matched to existing keyframes and provide accurate pose estimates at or near key poses. Such approaches suffer from tracking jitter and can require several keyframes for robust tracking. In an uncontrolled environment, it may not be possible to accurately establish multiple keyframes.

A critical issue in all 3D model based approaches, is the accurate estimation of the tracking model. In keyframe approaches, accurate pose is also required at keyframes. Indeed, [2] performs optimal pose and model estimation at keyframes using global bundle adjustment. This preprocessing is lengthy and is acceptable for offline tracking, or in situations where the subject to be tracked is known and can be enrolled in the system prior to the tracking phase. However, such effort is impractical for more general "ad-hoc" tracking situations such as surveillance.

View synthesis approaches for rapid model registration can be used to render the appearance of the tracking model at different poses as in [4]. A best-fit search among these views reveals the correct registration parameters. This method performs well when lighting conditions are consistent between the rendered face and the face image. However, like most appearance based approaches is likely to be sensitive to drastic lighting changes and cosmetic changes on the face such as facial hair and makeup.

Most model based trackers assume a rough estimate of face shape such as an ellipsoid in [9][6] and a cylindrical model in [5]. In each of these approaches the initial inaccurate tracking mesh remains static throughout the tracking sequence, introducing considerable error.

In the model-based bundle adjustment work by Shan et.al. [3] a generic face model is allowed to deform to account for both facial deformations and

rigid transformation. The number of optimization parameters is reduced by constraining the model points to lie on the surface of a mesh defined by a linear combination of face-metrics. For further performance, the dependence on the 3D model parameters is eliminated using a transfer function that estimates 3d as a projection onto the model surface. Subsequent optimization is performed only over camera parameters and model coefficients. Because the deformed model is constrained to be a linear combination of existing models, model error will be present if the subject's face can not be modeled as such (ie, does not lie in the convex hull of the basis shapes). Though significantly faster than classical bundle adjustment formulations, performance is not realtime. The tracker used in this paper uses a similar approach but ignores model deformation to perform rigid face tracking.

The work most similar to our update approach is [1] where a complex head model is fit to a sequence of face images. After recovering accurate head pose from bundle adjustment on sets of image triplets, stereo matching is performed on image pairs and a generic face mesh is fit to the recovered 3D. In lieu of local bundle adjustment with fixed internal camera parameters Jebara et. al. recursively estimate camera geometry (focal length), mesh structure, and pose [12] within an extended Kalman filter framework [10].

In [11] potentially erroneous feature point matches are eliminated by focusing on a set of optimally trackable feature points where optimality is a function of the determinant of the Hessian at a given feature location and the corresponding surface normal of the point projected onto the model surface.

In contrast to [12] and [11] we separate model update from the internal optimization scheme of the tracker. Mesh vertices are updated using estimates of head pose acquired with the current 3D model. Tracking improves after reinitialization with the updated model. Though the update approach is tested with a specific tracker, maintaining the update outside of the internal tracking mechanism enables augmentation of any existing model based tracker.

3   Rigid 3D Tracking Overview

The starting point for our investigation is the tracker by Fua et. al. that combines a recursive and keyframe based approach to minimize tracking drift and jitter, and reduce the number of keyframes required for stable tracking. This section presents a brief overview of the tracking approach, but the reader is deferred to the original paper [2] for details.

A keyframe in [2] consists of a set of 2d feature locations detected on the face with a Harris corner detector and their 3D positions estimated by back-projecting onto a registered 3D tracking model. The keyframe accuracy is dependent then on both the model alignment in the keyframe image, as well as the geometric structure of the tracking mesh. Especially when the face is far from the closest keyframe, there may be several newly detected feature points not present in any keyframe that are useful to determine inter-frame motion.

These points are matched to patches in the previous frame and combined with keyframe points for pose estimation.

The current head pose estimate (or closest keyframe pose) serves as the starting point for a local bundle adjustment. Classical bundle adjustment is typically a time consuming process, even when a reasonable estimate of camera and 3D parameters is provided. However, by constraining the 3D points to lie on the surface of the tracking model, the method is modified to run in real-time without substantial sacrifice in accuracy. When an accurate 3D model of the tracked object is used, reported accuracy approaches that of commercial batch processing bundle adjustment packages requiring several minutes per frame.

Unfortunately, a perfect 3D model of the tracked subject is rarely available to the tracker a priori. As we will show next, tracking performance can degrade drastically when a generic face model is used due to errors in initial alignment. Experiments on real video sequences also exhibit problems due to limited feature point coverage on face images. These issues are somewhat more significant as they are less predictable and can result from an inherent lack of sufficient information in the image.

We first describe the data used in the synthesized and real video experiments and present results and analysis of experiments demonstrating the dependence of tracking accuracy on mesh accuracy and alignment. The mesh update method is detailed and improved tracking results are shown using the updated models. This is followed by an investigation of performance on real image sequences.

4   Test Data

4.1   Synthesized Motion

A set of experiments is performed on sequences of rotating 3D faces. To generate the sequences, textured 3D models of four subjects are acquired using the Face-Vision200 modeling system [14]. For each model, two independent sequences of images are rendered. The first consists of pure rotation about the horizontal (X) axis, and the second, rotation about the vertical (Y) axis. In both cases, the sequences begin with the subject facing the camera and proceed to -15 degrees, then to 15 degrees, and return to neutral in increments of 1 degree. A total of 60 frames is acquired for each sequence. Image dimensions are 484x362.

4.2   Real Video

Two real video sequences are tested for consistency with the synthetic trials. In both cases a subject is instructed to rotate his head from right to left mimicking the synthetic sequences. Ground truth rotation is acquired using commercial bundle adjustment software [15].
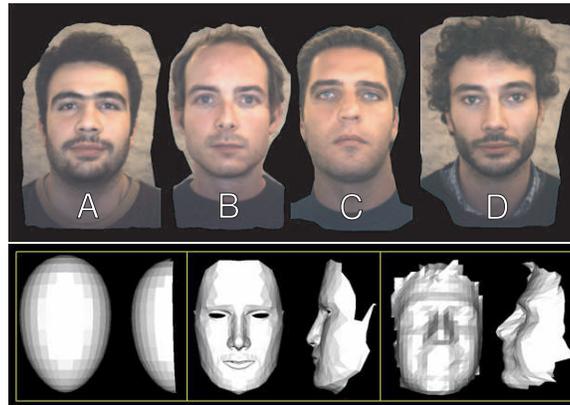
Fig. 2. (top) Four test models. (bottom) Ellipsoid, generic face, and example true mask (for subject A) used for tracking.

5 Investigation of tracking model bias

The tracker utilizes two primary sources to estimate camera pose: prior and observed information. The model prior information is embedded in the keyframes and is defined by the tracking mesh, its initial pose, the 2D feature points detected on the face, and their 3D positions estimated by back-projecting to the registered mesh. Observed data consists of 2D feature points detected in non-keyframe images that are matched to the pre-defined keyframe features. Indeed these are fundamental information sources in many 2D-feature-based 3D-trackers, hence the analysis extends beyond the particular choice of tracker in this investigation.

While errors in both the prior and observed data can contribute to tracking inaccuracies, the effects of the latter are negligible in the controlled synthetic sequences. We therefore focus our attention on tracking bias induced by inaccuracies in the model prior and defer the analysis of observed information to the discussion of real sequences later in the paper.

5.1 Investigation 1: Mesh Accuracy

To demonstrate the connection between tracking and model accuracy, tracking results are compared for four different tracking meshes: an ellipsoid, a generic face mask, an updated mesh, and an accurate ("true") 3D model of the subject. The ellipsoid is a weak prior, making no assumptions regarding the location of features on the face such as the eyes, nose, and mouth. The generic face mesh makes stronger assumptions on these features, but other than the manual fitting process (which involves a nonuniform scaling of the mesh) does not account for the true structure of the subject's face. The updated mesh is a refined version of the ellipsoid and makes equally strong assumptions as the generic mask, but
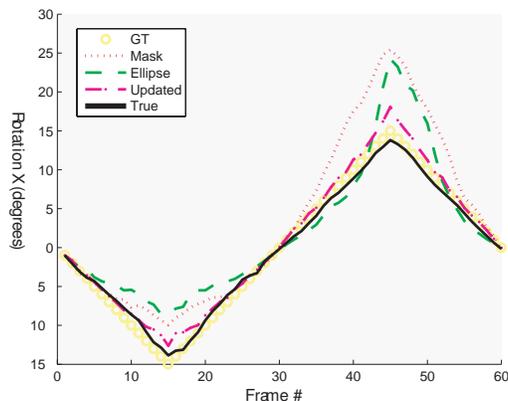
Fig. 3. Recovered X rotation in degrees (vertical axis) versus frame number (horizontal axis) from the tracker for each of the test meshes. Results are from a synthetic sequence with pure rotation about the horizontal axis. Ground truth (GT) shown for comparison.

derives these assumptions from observed data (discussed in Section 5.3). The true mesh for each subject is derived from the same model used to generate the sequence. The texture is not used, but the geometry is identical, eliminating errors due to geometry inaccuracies. To balance the comparison, each mesh is designed or edited to cover only the face portion of the model as shown.

Figure 3 shows the X component of the recovered rotation compared to ground truth on a representative sequence. Aggregate error for all four subjects is shown in the chart in Figure 4. The average sum of square differences (SSD) is computed with respect to the known ground truth for each degree of freedom.

The largest error consistently occurs with the generic face, and least error with the true mesh. It is evident (and expected) that performance of the tracker improves significantly with the true model geometry. An interesting observation, however, is that the ellipsoidal mesh actually performs better than the face mask in most cases.

An explanation for this is that the mask imposes a stronger (but erroneous) prior on the tracker. Prominent features such as the nose and chin are difficult to align properly using only an aspect change, and in some cases it may not be possible at all given different proportions of human faces. These discrepancies are not significant at small rotations, but become more prominent as the out-of-plane motion increases.

Indeed the example in Figure 3 exhibits tracking performance that is similar for both the ellipsoid and mask within 3-5 degrees of the keyframe. However when more of the face profile is exposed, chin and forehead alignment becomes an issue with the tracker attempting to compensate for the misalignment. Results from the updated mesh are discussed in Section 5.3.
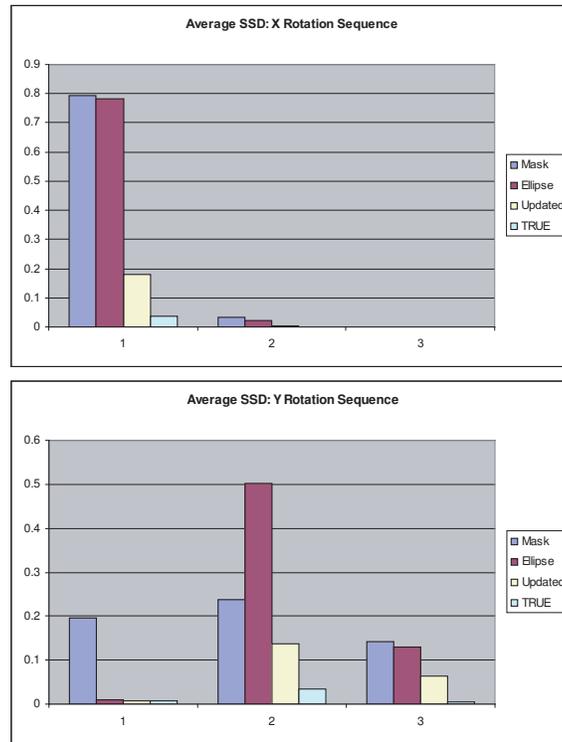
Fig. 4. Average performance over all test subjects on synthetic sequences. Largest error consistently occurs with the generic face. The three groups along the horizontal axis correspond to average rotational tracking errors in X,Y, and Z respectively. Each of the four bars in each group reflects average tracking accuracy (SSD) for one of the four tracking meshes shown in the legend. (top) X-axis rotation (bottom) Y-axis rotation. Units are in degrees.

5.2   Investigation 2: Model Registration

Referring back to Figure 1, a mesh that appears properly aligned in a frontal image may actually be grossly misaligned as is apparent in the profile view. This misalignment establishes incorrect a priori information. While the effects of the model bias may be negligible near the original keyframe, as tracking proceeds, the tracker will attempt to resolve the new feature information with the incorrect keyframe information by minimizing reprojection error. As keyframe information is "trusted" to be correct, the result is biased toward an incorrect conclusion. This section provides empirical evidence for this phenomenon with test sequences of intentionally misaligned meshes.

   The keyframe alignments of the previous section are perturbed by rotating 5 degrees about the horizontal axis. Figure 5 shows the results of tracking with the misaligned meshes. Overall performance decreases for each of the meshes.
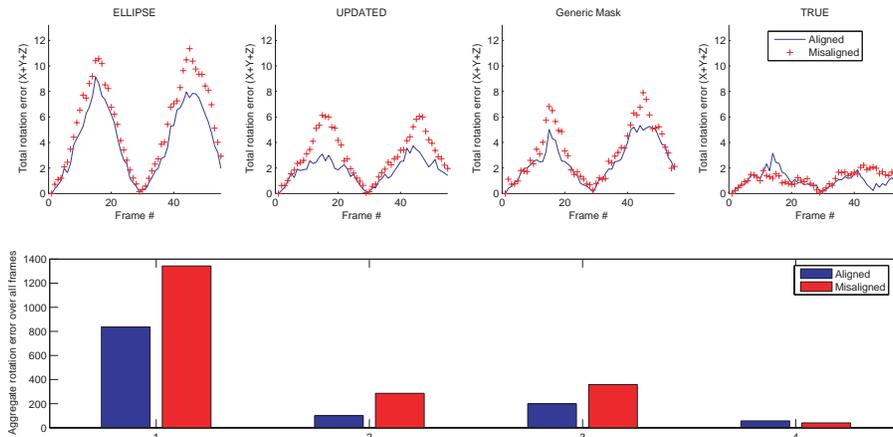
Fig. 5. Results from misalignment experiments. (left) Comparison of tracking error for each image with aligned and misaligned meshes rotated 5 degrees about the horizontal axis. (right) Aggregate error over all frames of sequence.

In the case of the TRUE mesh, there is a marginal difference in performance. It is expected that due to the fact that faces are relatively smooth continuous surfaces, small deviations in alignment for perfect geometry will embed smaller errors in the prior. Though not tested, larger errors in alignment should induce similar magnitude errors for all face-like meshes.

5.3   Investigation 3: Online Model Refinement

The results in the previous sections demonstrate that despite perfect 3D information, tracking performance can degrade significantly when the model is misregistered in the keyframes. Errors in the geometry of the tracking mesh introduce similar errors. Both of these error sources can be minimized by updating the geometry of an initial tracking model online. Beginning with a rough estimate of the face geometry and we iteratively refine the model and use this more accurate structure to re-estimate the 3D keyframe features thereby reducing the erroneous bias imposed by the misaligned mesh.

Any starting mesh is a candidate for update however an ellipsoid is chosen for its qualitative approximation of face shape without introducing strong assumptions on feature location.

Update method The 3D locations of the vertices of the tracking mesh are updated as follows:

The tracker is initialized with a 3D mesh with roughly the same proportions as the subject's face. As shown in the previous section, using a more complicated generic face model does not necessarily improve initial tracking accuracy (and in some cases can hinder it). Rather than risk introducing a strong erroneous bias
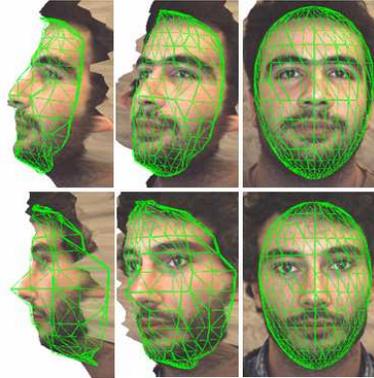
Fig. 6. Updated tracking meshes at different poses. The updated structure conforms well to the subject's face.

with a misaligned generic face mesh, we use an ellipsoidal mesh as it assumes nothing about face orientation or location of features. Furthermore, in our current experiments tracking with the ellipsoid provides good pose estimates within a few degrees of the initial keyframe. This baseline is sufficient for incremental improvement of the sparse tracking model.

The ellipsoid mesh is manually aligned with the face in the first frame by applying a translation and nonuniform scaling to the mesh. A single keyframe is generated using this initial registration consisting of the projection matrix P_0, model vertices X_i, and their projections $x_i = \Phi(P_0, X_i)$. A set of "update features" is generated by sampling a $7x7$ window at each x_i.

The tracker provides a new P_t for each image I_t. When a suitable baseline is achieved (3-5 degrees) using the initial tracking model, the update features are matched by correlation in I_t. Using camera estimates P_0 and P_t, straightforward stereo reconstruction [13] is performed at matched features and the new 3D location of model vertices is updated.

The original keyframe mesh is substituted with the updated mesh and a new keyframe is generated. In our current experiments a single update pass is performed. However, the improved tracking results allow multiple passes to be performed to increase the model and tracking accuracy.

Mesh update results We use the method in the previous section to generate updated versions of the ellipsoid for each of the subjects. The synthetic sequences of section 4.1 are re-tracked using the updated models as described. Figure 6 shows the tracking mesh after a single update for two models at initialization and an intermediate stage of tracking. The profile view is generated manually to show the accuracy of the alignment. After a single update, the mesh captures the overall shape and prominent features of the subjects, obviating the need for precise alignment.
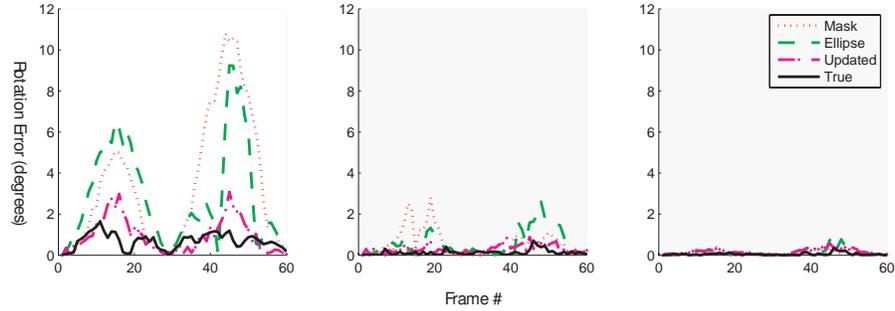
Fig. 7. Absolute tracking error in X, Y, and Z-axis rotation relative to ground truth with synthetic "X-Rotation" sequence. Comparison of results with four tracking meshes.
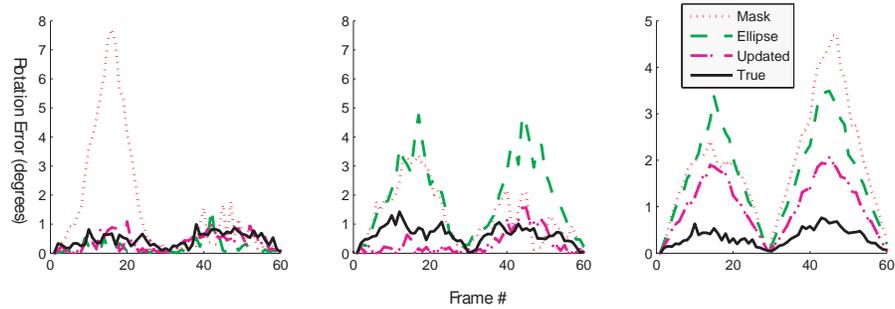

Fig. 8. Absolute tracking error in X, Y, and Z-axis rotation relative to ground truth on with synthetic "Y-Rotation" sequence. Comparison of results with four tracking meshes.

Figures 7 and 8 show tracking results for the two sequences of subject A (X and Y rotation respectively). The top row shows the recovered head rotation separated into X, Y, and Z components.

The average results over all four subjects are summarized on the chart presented earlier in Figure 4. The tracking performance with the updated meshes is considerably better than the ellipse or generic mask for all tracked parameters.

Though the reduction of negative model bias with the ellipsoid is desirable, the mesh itself is not optimal. It is a coarse regular tessellation that does not take into account expected locations of features on the face. If important features (such as the nose bridge or chin boundary) do not happen to fall under the ellipsoid vertices, the update process cannot adequately capture the complete face structure. The sparsity of the ellipse template also increases the average error of the updated mesh. This problem may be remedied by either a uniformly dense tessellation, a non-uniform tessellation accounting for the expected location of
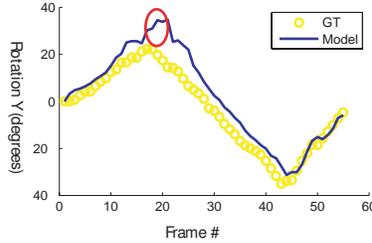
Fig. 9. A real sequence is tracked with the true mesh for the subject. Excellent tracking performance is expected, however the tracker gets stuck in a local minimum at the red circle due to poor feature point coverage.

important features, or an alternative update approach. The generic face mask is better with respect to tessellation, however it also makes strong assumptions on feature locations, preventing adequate alignment without a nonlinear scaling of the geometry (which requires identification of feature locations such as the eyes, mouth, and nose).

6 Real Video Sequences

The synthetic experiments support the claims that mesh accuracy and keyframe registration play an important role in accurate 3D tracking. When tracking faces in real video sequences, however, we must contend with lower quality input data that may affect the tracker in unpredictable ways. We therefore focus the remainder of the paper on the discrepancies between the expected results (as predicted by the synthetic experiments) and the results observed on real sequences, in order to identify sensitivities in 3D face tracking.

The most surprising case shown in Figure 9 will be the focus of our analysis. This is a clear cut case where the subject is being tracked with the true geometry of his face and should be expected to perform considerably better than the other meshes (as was the case with the synthetic trials). However, it turns out that the tracking accuracy is worse than all but the ellipse. Tracking progresses well up to a point where it appears that the mesh gets locked into an incorrect pose configuration.

The discrepancy between real and synthetic sequences can be explained by the number of accurately matched keyframe feature points and the face coverage they provide.

The number of feature points detected in the high error frames is significantly lower than the best case tracking results. More importantly, the correctly matched keyframe points are clustered on the portion of the face closest to the camera providing poor face coverage and creating pose ambiguity. The tracker minimizes the keyframe point reprojection error, but settles on a local minimum

corresponding to a poor tracking estimate. The tracker remains stuck in this local minimum for subsequent frames until more feature points are matched.

Comparing these results to the sequence tracked with the generic mesh, we observe another surprising phenomenon: in this case, the generic mesh performs *better* and doesn't get stuck in the local minimum. It turns out that feature point matching is dependent upon the local surface normal of the tracking mesh at the backprojected feature location. Therefore, given the same input image and 2D keyframe features, it is possible for a different number of points to be matched. Indeed, this is the cause of the discrepancy: While the set of keyframe points matched in the true and generic cases is different throughout the sequence, at the divergence point a single critical feature point is lost while tracking the true mesh. The loss of this point leaves a feature set that covers only a small portion of the face, inducing a less favorable error surface.

6.1 Reprojection error

In all cases, the tracking performance improves with model accuracy and alignment. A reasonable assumption, therefore, is that overall tracking performance is directly related to feature point reprojection error and a plot of reprojection error over time would be highly correlated with a similar plot of tracking error. Though large tracking errors induce large reprojection errors, the converse is not true: low reprojection error does not necessarily indicate low tracking error. This is due to the fact that as the tracker discards low confidence feature points, it is possible to settle into a minimum configuration where the reprojection error for detected keyframe points is low, but the tracking error is high.

7 Indications

The preceding analysis on controlled, synthesized motion sequences demonstrated a strong dependency between tracking accuracy and mesh geometry and alignment. Trials on real video uncovered a sensitivity to feature point number and coverage. We therefore conclude with a list of issues that should be considered when using and evaluating 3D model based trackers.

MESH COVERAGE: For a detected feature point to be registered as a keyframe point, it must back project onto the mesh at the initialization phase. Tracking meshes with smaller face coverage may miss important potential keyframe points on the outer boundary of the face. Therefore a tracking mask should be maximized to cover as much face area as possible.

IMAGE QUALITY: Despite the fact that the pixel area occupied by the face in the real sequences is larger than the synthetic cases by roughly 30%, on average 5 times fewer feature points are matched on each frame. Care should therefore be taken to either maximize image quality or tune feature detection parameters accordingly.

FEATURE POINTS AND LOCAL MINIMA: Absence or inclusion of a single feature point can cause a dramatic change in the estimated pose. If the tracker gets stuck in a local minimum in the reprojection error surface, the pose may remain skewed until a sufficient number of reliable feature points are matched again. These local minima can be avoided or detected by analyzing the proportion of the face covered by the detected feature points.

MODEL REFINEMENT: Tracking accuracy is greatly influenced by mesh geometry and registration errors. If an accurate 3D model of the tracked subject is not available a priori, refinement of the structure online can mitigate both error sources simultaneously.

NON-LOCAL BUNDLE ADJUSTMENT: The experiments in this paper were performed with a single registered keyframe. Given an adequate number and coverage of feature points, it is sufficient to consider only the key and previous frame in the optimization. However, as we have seen, it is possible to get stuck in a local minimum when coverage is poor. Considering additional frames, though increasing the computational burden, is likely to help avoid local minima. This suggests a bundle adjustment framework with a variable size window of frames, dependent on the expected quality of the data (for example, based on feature the current number or coverage of feature points).

8 Conclusions

Using an existing model-based tracker, we have demonstrated the dependence of tracking accuracy on the accuracy of the underlying model geometry and registration. We have shown that a simple stereo based approach to mesh update significantly improves tracking performance. A single update of the model is performed using the narrow baseline camera pose recovered by the tracker.

Updating the mesh eliminates the need for multiple view rotational alignment of the mesh, as the resulting model automatically conforms to the subject's features. Aspect and translation alignment is still needed at initial ellipsoid placement, but this is a much simpler process and can be performed, for example, using the head bounding box information.

The discrepancy between the synthetic and real sequence results are attributed to the sensitivity of the tracker to initial pose alignment and lack of sufficient feature points matched to the keyframes on real sequences. When feature points do not span the entire face region, the pose optimization can get stuck in local mimima on the reprojection error surface corresponding to high pose error. We have provided a set of recommendations based on the investigations that we hope will assist in the development, implementation, and use of 3D tracking methodologies.

9 Acknowledgments

References

1. P. Fua, "Using model-driven bundle-adjustment to model heads from raw video sequences," *In Proceedings of the 7th International Conference on Computer Vision*, pages 4653, Corfu, Greece, Sept. 1999.
2. L. Vacchetti, V. Lepetit, P. Fua, "Stable Real-Time 3D Tracking Using Online and Offline Information," *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(10): 1385-1391 (2004).
3. Y. Shan, Z. Liu, and Z. Zhang, "Model-Based Bundle Adjustment with Application to Face Modeling," *International Conference on Computer Vision*, Vancouver, Canada, July 2001.
4. V. Lepetit, L. Vacchetti, D. Thalmann, and P. Fua, "Fully Automated and Stable Registration for Augmented Reality Applications," *International Symposium on Mixed and Augmented Reality*, Tokyo, Japan, September 2003.
5. M. 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*, 22(4), April 2000.
6. S. Basu, I. Essa, and A. Pentland, "Motion regularization for model-based head tracking," *International Conference on Pattern Recognition*, 1996.
7. D. DeCarlo and D. Metaxas, "The Integration of Optical Flow and Deformable Models with Applications to Human Face Shape and Motion Estimation," *Computer Vision and Pattern Recognition*, 1996.
8. Schodl, A., A. Haro, and I. Essa, "Head Tracking using a Textured Polygonal Model," *In Proceedings of Perceptual User Interfaces Workshop (held in Conjunction with ACM UIST 1998)*, San Francisco, CA., November 1998.
9. A. Azarbayejani, T. Starner, B. Horowitz, and A. Pentland, "Visually controlled graphics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6), 1993.
10. A. Azarbayejani and A. Pentland, "Recursive Estimation of Motion, Structure, and Focal Length," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(6), 1995.
11. J. Strom, T. Jebara, S. Basu, and A. Pentland. "Real time Tracking and Modeling of Faces: An EKF-based Analysis by Synthesis Approach," *Proceedings of the Modeling People Workshop at ICCV'99*, 1999.
12. T. Jebara and A. Pentland, "Parameterized Structure from Motion for 3D Adaptive Feedback Tracking of Faces" *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'97)*, 1997 .
13. R. Hartley and A. Zisserman, "Multiple View Geometry in Computer Vision," Cambridge University Press, Cambridge, UK, 2000.
14. Geometrix, (http://www.geometrix.com).
15. EoS Systems Inc., (http://www.photomodeler.com).