3D Pose Estimation for Planes

Changhai Xu
CS, UT Austin
changhai@cs.utexas.edu

Benjamin Kuipers
CSE, U of Michigan
kuipers@umich.edu

Aniket Murarka
CS, UT Austin
aniket@cs.utexas.edu

Abstract

This paper presents a method to robustly track planes and estimate their 3D poses in a video. A weighted incremental normal estimation method for planes (WINEP) is presented using Bayesian inference. This estimation method guarantees an optimal solution based on all the observations up to the current time, and the computational cost at each time step does not increase with the growing number of past frames. The tracking algorithm integrates boundary information with point feature tracking, which avoids accumulating errors due to intensity changes, image noise, and inaccurate parameter estimation. The tracking algorithm deals with low-textured as well as highly-textured planes. The tracked boundary locations provide the input data for 3D plane pose estimation.

Experiments show that our hybrid tracking method using both point and line features is better than using only point features, and our pose estimation algorithm is more robust and accurate than the conventional homography decomposition method, especially under circumstances of noisy observations and low number of input features.

1. Introduction

Planar surfaces abound in man-made environments, such as surfaces on buildings, walls, boxes, and many other manufactured objects. Tracking these planar surfaces and estimating their poses play an important role for an intelligent agent building models of its environment. Robust plane pose estimation algorithms can help build 3D models of objects that are composed of (approximately) planar surfaces. In this paper we propose a method to robustly track planar surfaces and estimate their poses.

We refer to a planar surface enclosed by a 2D boundary as a *component*, and assume the component boundary is a chain of line segments, i.e., a polygon. The goal of this paper is to robustly estimate the 3D poses for components in an image sequence through tracking.

Point feature tracking, such as the KLT method [19], is widely used to track moving objects. However, it suffers from the feature drift problem in a long sequence of images, and will accumulate errors due to intensity changes, image noise and inaccurate parameter estimation. Instead, boundary features are much more robust when the lighting condition changes or significant noise exists. To avoid error accumulation, we build a hybrid tracker by incorporating boundary features into point feature tracking.

One of the highlights of our tracking method is that we exploit the KLT algorithm to maintain only temporary correspondence between each two adjacent frames based on point features, while the permanent correspondence across all the frames is maintained by boundary features. The temporary correspondence from the KLT algorithm is only used to predict the boundary location from one frame to the next. Then, within the local areas of the predicted boundary, a correction step is taken by selecting the best matched line segment detected using the Hough transform [6]. Our tracking method works for low-textured as well as highly-textured components.

The tracker records a history of the component boundary locations, which are provided as input data for estimating 3D component poses. A key step in component pose estimation is estimating the component's normal. Traditional solutions include the analytical method through decomposing a homography matrix, and the nonlinear optimization (bundle adjustment) method. However, these methods suffer from problems such as being too sensitive to noise, expensive computation, or local maxima. In addition, the nonlinear optimization method requires a good starting point and there is no guarantee that it will converge, which makes it hard to use in robotic applications.

We present a probabilistic method, WINEP (Weighted Incremental Normal Estimation for Planes), which provides an optimal estimation of component normals based on tracked features from all past frames. The method gives a robust estimation result and also has a low computational cost per frame that is independent of the number of observed frames due to the recursive nature of the probabilistic formulation. This estimation method works more robustly and accurately than the conventional homography decomposition method when there are only a few input features and when the input features are noisy.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 and 4 present our tracking and pose estimation methods in detail respectively. Experimental results are presented in Section 5. Section 6 discusses future work and concludes.

2. Related Work

Various kinds of representations can be used in object tracking. Comaniciu *et al.* [5] proposed a kernel-based tracking algorithm where an object is represented by an ellipsoidal region in the image and the mean-shift tracker maximizes the appearance similarity. Isard and Blake [13] presented a particle filter based tracking algorithm where object shape is represented by B-splines. Tran and Davis presented a robust object tracking method using regional affine invariant features [20].

Distinctive local point features have been widely used for tracking, such as by the KLT method [19] or the SIFT matching method [16, 10]. While point features have many successful applications, maintaining feature tracks over many frames may be quite difficult [20, 23], especially when the input images are noisy. The KLT method is efficient, but it may suffer from the feature drift problem over a long sequence of images [23, 2, 9].

More robust tracking can be obtained by integrating other features with point features. Gall *et al.* [9] exploited region matching to avoid point feature drift. Pressigout and Marchand [18] and Vacchetti *et al.* [21] incorporated point and edge features for tracking by minimizing the reprojection errors. Similar to [18, 21] our method integrates boundary features and point features, but it differs in that only boundary features are used for permanent correspondence across the images and point features only maintain temporary correspondence. This allows us to achieve good tracking performance since the boundary features in general are more robust to image noise.

Estimation of the normals of planar surfaces plays an important role in our work. Two images of the same 3D plane are related by a homography matrix. The homography matrix can be calculated given at least four pairs of matching points/lines [11, 17]. Then the plane pose, rotation and translation between the two camera spaces can be obtained by decomposing the homography matrix [17, 4, 15]. This method is fast, but it is sensitive to noise and may give more than one physically possible solution. Zelnik-Manor and Irani [22] derived constraints for multiple planes across multiple views to improve homography estimation. A nonlinear optimization method using multiple images [12, 17] can be used to improve the homography decomposition method, but it has high computational cost because a relatively large number of parameters need to be estimated at the same time. Improvements were made in [7] to achieve real-time performance for a few dozen frames. The nonlinear optimization method also requires good initializations which are hard to guarantee in practice. We have developed a new probabilistic method for plane normal estimation to overcome these problems.

3. Component Tracking

The boundary for a component is detected by searching for contours that are closed and composed of a sequence of line segments, within the moving region detected by background subtraction. Once the boundary is detected, a tracker is assigned to the component, and tracks the component automatically over time.

A tracker can be very generically defined as a symbolic pointer into the sensory stream that maintains the correspondence between a higher-level, symbolically represented concept and the ever-changing image in the sensory stream [14]. Specifically, in our system a tracker maintains the correspondence between the symbol "component" and its time-varying boundary locations in the image sequence. The history of the boundary locations will provide input data for component pose estimation in Section 4.

Sparse point feature tracking methods such as the KLT method [19] and the SIFT matching method [16] have been widely used to track moving objects. We use the KLT method in our work due to its efficiency and effectiveness in videos where only small motion takes place between consecutive frames.

One important difficulty with point features is that they may drift slowly away from the correct positions from frame to frame, especially in low-resolution images, due to intensity changes, image noise and parameter estimation errors. In general the drift between adjacent frames is very small, but the accumulated error across a long sequence of frames can be very problematic. To solve this problem, we integrate boundary information into our tracking algorithm.

3.1. Point Feature Tracking

We track KLT point features only between each two adjacent frames instead of across multiple frames due to the possible feature drift problem. That is, features are redetected at each time step and different sets of features are used between different adjacent frames.

Salient point features are first detected inside the component boundary at time $t-1$. Then the features are tracked by the KLT tracker at time t. The RANSAC algorithm [8] is used to remove incorrectly matched features. In our experiments, the KLT tracker works very robustly since it only has to maintain the feature correspondence between two consecutive frames. This feature correspondence will be used to predict the component boundary location at time t, based on the already-known boundary location at time $t-1$.

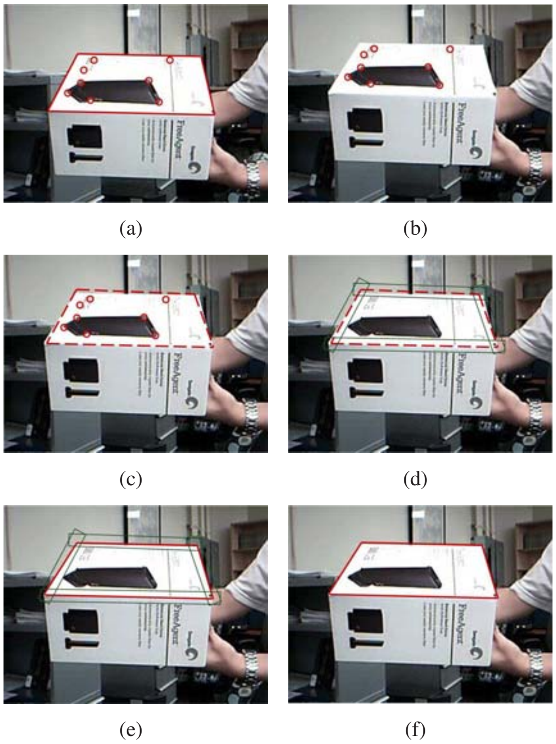

Figure 1. Tracking steps. The first figure is based on the image at $t-1$, and all the others are based on the image at t. (a) The red solid contour is the component boundary s^c_{t-1} in frame $t-1$, and point features are detected inside the boundary shown as red circles. (b) The point features are tracked in frame t using the KLT tracker. (c) The boundary in frame t is predicted as $\hat{s}^c_t$ (shown as a dashed contour) by a transformation of s^c_{t-1}, where the transformation is obtained from feature correspondence between these two adjacent frames. (d) Local interest regions (shown as green rectangles) are formed around each line segment on $\hat{s}^c_t$. (e) Within each local interest region, lines are detected using Hough transform and the best matched line is selected (shown as a red solid line). (f) The final boundary s^c_t in frame t is computed.

3.2. Boundary Prediction

The detected features at time $t-1$ and the tracked features at time t are related by a planar homography transformation H_{at}. Given at least four pairs of matched point features, H_{at} is calculated through the Direct Linear Transformation (DLT) method [12, 17, 1].

Let the component boundary at time $t-1$ be s^c_{t-1}, then we have

$$\hat{s}^c_t = H_{at} s^c_{t-1} \tag{1}$$

where $\hat{s}^c_t$ is the predicted boundary at time t.

3.3. Boundary Correction

In order to avoid accumulated error from point feature tracking, we update the predicted boundary to fit the observed data by matching line segments in their neighborhood areas. Compared to point features, line features are much more robust to intensity changes and image noise. Also line features tend to give more accurate position estimation than point features.

Around each line segment on the predicted boundary $\hat{s}^c_t$, a local interest region is formed in the image at time t. The interest region is a rectangle with a user-defined height and width, where the rectangle's centroid is the line segment's centroid, and two laterals of the rectangle are parallel to the line segment. Within this rectangle, candidate line segments are detected using the Hough transform [6] after the Canny edge detection process [3], and the best matched line segment is used to correct the predicted line segment.

From at least four pairs of matched line features, another homography matrix H_{bt} is obtained [11], and the component boundary at time t is finally updated as

$$s^c_t = H_{bt} \hat{s}^c_t \tag{2}$$

or alternatively

$$s^c_t = H_{bt} H_{at} s^c_{t-1} \tag{3}$$

The implementation of the hybrid tracker is demonstrated in Fig. 1.

3.4. Discussion

In our hybrid tracker, point features maintain only temporary correspondence between each two adjacent frames, while line features maintain the permanent correspondence across all the frames for the tracked component. Since line features are more robust to image noise, this tracking scheme can work well with low quality videos.

The Hough transform is applied only within the local interest regions of the predicted boundary. In general the KLT tracking is fairly accurate between adjacent frames, so the interest regions are typically small such that the computational cost for boundary correction is low.

This hybrid tracking scheme doesn't require the component to be highly-textured. For low-textured components it can also work well, because the estimation error in H_{at} will not be propagated due to the boundary correction step. We tested this case in our experiments where only 10 features were used.

Although untextured components are not our focus in this paper, the tracking scheme can potentially deal with them by only using the boundary correction step. In this case, the interest regions must be made larger.

4. Component Pose Estimation

Now that a component has been robustly tracked over frames, we want to estimate its pose in each frame. To do this we present a probabilistic method WINEP (Weighted

Incremental Normal Estimation for Planes), which provides an optimal solution based on all observations up to the current frame, and the computational cost at each time step does not increase with the growing number of frames.

4.1. Background

Since a component is a planar surface, the coordinates of points between two poses of the component are related by a 2D (planar) homography H, and we have $H = R + TN^T/d$, where R and T are the rotation and translation matrices relating the two poses, d and N are the distance and unit normal of the plane in the reference camera space.

From at least four pairs of matching points/lines, H can be determined up to a scaling factor using the Direct Linear Transform method [12, 17, 11]. The computed H can then be decomposed to give two sets of solutions for $\{N, R, T/d\}$ (Algorithm 5.2, [17]).

While this method is fast, it's sensitive to noise, especially when the number of input features is small and the observed features are noisy. Also the selection of the correct solution from the two candidate solutions can be difficult without prior knowledge.

One way to overcome the problem of noise and unstable pose estimates is to minimize an error measure over several images at the same time. This nonlinear optimization method can improve on the results of the decomposition method if the estimates of the rotations, translations, and normal are good to begin with. But it is hard to converge with poor initializations and the algorithm sometimes converges to the wrong minima. Even with good initializations, there is no guarantee for convergence. Furthermore, the running time of the minimization process can be very expensive if the number of features and frames is high. These issues make the method hard to use in robotic applications.

4.2. Geometric Constraints

The world space is chosen to be aligned with the camera space. We define a component space, where the x-axis is arbitrarily chosen on the component, the z-axis is along the direction of the component normal, and the origin can be any arbitrary point on the component. Note that every point belonging to the component will have zero value on the z-axis in this component space.

The frame sequence is numbered as $1, \ldots, t$. We also denote a certain frame as frame 0 which is also called the *reference frame*. Note that frames 1 through t are consecutive frames, but frames 0 and 1 are not necessarily so. At any time t, the component normal is denoted as N_t, and its distance to the origin of the camera space as d_t.

Thus the 3D component at $t = 0$ can be represented as

$$N_0 P = d_0 \tag{4}$$

for any 3D point $P = (P_x, P_y, P_z)^T$ on the component. Let $p = (p_u, p_v)^T$ be the calibrated image coordinates corresponding to P. Using a perspective camera model, we can easily calculate P from Eq. 4 given N_0, d_0 and p. Thus a component space can be built, and the corresponding point of P is denoted as $P^c = (P^c_x, P^c_y, 0)^T$ in the component space.

At time t, let the translation and rotation from the component space to the camera space be T_t and $R_t = (R_{1t}\ R_{2t}\ R_{3t})$, where $R_{kt} (k = 1, 2, 3)$ are the column vectors in R_t. The point P^c on the component and the its image coordinates $p_t = (p_{ut}, p_{vt})^T$ are related by

$$\begin{aligned} \lambda^P \begin{pmatrix} p_{ut} \\ p_{vt} \\ 1 \end{pmatrix} &= (R_{1t}\ R_{2t}\ R_{3t}\ T_t) \begin{pmatrix} P^c_x \\ P^c_y \\ 0 \\ 1 \end{pmatrix} \\ &= (R_{1t}\ R_{2t}\ T_t) \begin{pmatrix} P^c_x \\ P^c_y \\ 1 \end{pmatrix} = H_t \begin{pmatrix} P^c_x \\ P^c_y \\ 1 \end{pmatrix} \end{aligned} \tag{5}$$

where λ^P is the point depth in the camera space, and H_t is a homography matrix that maps points from the component plane to the image plane.

The transformation H_t can be determined up to a scaling factor based on four or more pairs of matching points/lines. Since R_t is a rotation matrix, it satisfies $\|R_{1t}\| = \|R_{2t}\| = 1$ and $R_{1t} \perp R_{2t}$. Equivalently we have the following constraints,

$$\|H_{1t}\| - \|H_{2t}\| = 0 \tag{6}$$

$$H^T_{1t} H_{2t} = 0 \tag{7}$$

where H_{1t} and H_{2t} are the first two column vectors in H_t.

4.3. Bayesian Formulation for Normal Estimation

To estimate the component pose in an image sequence, a key step is to estimate its normal N_0 and its distance to the origin d_0 in the reference frame. In the case of a single component, without loss of generality, d_0 can be set to any positive constant. We represent N_0 in a spherical coordinates as

$$N_0 = (sin\theta^N_0 cos\phi^N_0, sin\theta^N_0 sin\phi^N_0, cos\theta^N_0)^T \tag{8}$$

where $\theta^N_0 \in [0, \pi/2]$ and $\phi^N_0 \in [0, 2\pi)$ are the normal parameters.

Our goal is to estimate the probability density function $Pr(\theta^N_0, \phi^N_0 | z_{0:t})$, where z are the observations associated with the component.

By applying Bayes' theorem, we have

$$\begin{aligned} & Pr(\theta^N_0, \phi^N_0 | z_{0:t}) \\ \propto\ & Pr(\theta^N_0, \phi^N_0 | z_0) Pr(z_{1:t} | \theta^N_0, \phi^N_0, z_0) \end{aligned} \tag{9}$$

As discussed in Section 4.2 each z_k $(1 \leq k \leq t)$ has to meet certain geometric constraints given $\{\theta_0^N, \phi_0^N, z_0\}$, so we can estimate the normal parameters under the independent observation assumption that z_t is independent of $z_{1:(t-1)}$. Then Eq. 9 rewrites as

$$\begin{aligned} & Pr(\theta_0^N, \phi_0^N | z_{0:t}) \\ \propto \quad & Pr(\theta_0^N, \phi_0^N | z_0) \prod_{k=1}^{t} Pr(z_k | \theta_0^N, \phi_0^N, z_0) \end{aligned} \quad (10)$$

which enables us to obtain a recursive formulation as

$$\begin{aligned} & Pr(\theta_0^N, \phi_0^N | z_{0:t}) \\ \propto \quad & Pr(z_t | \theta_0^N, \phi_0^N, z_0) Pr(\theta_0^N, \phi_0^N | z_{0:(t-1)}) \end{aligned} \quad (11)$$

Based on Eq. 11 the problem is reduced to choosing the likelihood function $Pr(z_t | \theta_0^N, \phi_0^N, z_0)$ $(t > 0)$ and the prior function $Pr(\theta_0^N, \phi_0^N | z_0)$.

Computing the Optimal Solution. The entire parameter space for $\{\theta_0^N, \phi_0^N\}$ is uniformly discretized, and the globally optimal solution for N_0 is determined by choosing θ_0^N and ϕ_0^N that maximize $Pr(\theta_0^N, \phi_0^N | z_{0:t})$ in Eq. 11. The discretiztion resolution in our experiments is $\pi/64$. The parameter states whose posterior probability is lower than 10 percent of the highest probability are discarded at each time step.

4.4. Prior Information

When only a small number of frames are available and only little motion exists, the prior information on the component boundary will take the lead in determining the optimal pose, because motion information is still very ambiguous at this stage. The effect of prior information will gradually be phased out with increasing number of frames and increasing amount of motion.

If there is not much prior information available, we can simply set $Pr(\theta_0^N, \phi_0^N | z_0)$ to be a uniform function. Otherwise we can set the prior to be

$$Pr(\theta_0^N, \phi_0^N | z_0) = eval(s^c) \quad (12)$$

where s^c is the 2D component boundary shape in the component space and can be obtained from $\{\theta_0^N, \phi_0^N, z_0\}$, and $eval$ is a function that returns a score based on how well s^c satisfies certain prior geometric knowledge, such as centrosymmetric, rectangular and circular.

4.5. Likelihood Function

Given θ_0^N,ϕ_0^N,z_0 and z_t we calculate the homography matrix H_t in Eq. 5. H_t has to satisfy the geometric constraints shown in Eq. 6 and Eq. 7. So we can design the likelihood function $Pr(z_t | \theta_0^N, \phi_0^N, z_0)$ based on these constraints. Intuitively, the better the constraints are satisfied, the higher the probability that is assigned to the likelihood function.

So we design $Pr(z_t | \theta_0^N, \phi_0^N, z_0)$ as

$$\begin{aligned} & Pr(z_t | \theta_0^N, \phi_0^N, z_0) \\ = \quad & \gamma(1 + \lambda e^{-\frac{2\alpha_1 | \|H_{1t}\| - \|H_{2t}\| |}{\|H_{1t}\| + \|H_{2t}\|} - \frac{\alpha_2 |H_{1t}^T H_{2t}|}{\|H_{1t}\| \|H_{2t}\|}}) \end{aligned} \quad (13)$$

where α_1 and α_2 are user-determined positive constants, λ is the importance weight for the observation z_t (in comparison with the other observations $z_{1:(t-1)}$), and γ is a constant normalizing term..

Let s_0^c and s_t^c denote the component boundaries associated with z_0 and z_t respectively. We set the weight λ as

$$\lambda = e^{-\alpha_3 \frac{Ar(s_0^c)}{Ar(s_t^c)}} (1 - e^{-\alpha_4 \|H_{0t} - I\|}) \quad (14)$$

where α_3 and α_4 are user-determined positive constants, Ar is a function that returns the area of the corresponding boundary shape, H_{0t} is the normalized homography transformation between s_0^c and s_t^c, and I is a 3×3 identity matrix. The normalization procedure for H_{0t} can be found in Lemma 5.18 in [17]. Intuitively the weight λ will be assigned to a high value with large observed boundary s_t^c and large motion between frame 0 and frame t.

Due to the recursive formulation in Eq. 11 and the importance weight assignment in Eq. 13, we name our estimation method as WINEP (Weighted Incremental Normal Estimation for Planes). In the special case where λ is set to a positive constant, each observation z_k $(1 \leq k \leq t)$ makes the same contribution to the estimation result. We refer to WINEP with a constant λ as INEP in our experiments.

4.6. Pose Estimation

Once we have N_0, at any time t the homography matrix H_t in Eq. 5 can be obtained between the component plane and the image plane. Based on the constraints in Eq. 6 and Eq. 7, we can approximate the rotation/translation matrices from the component space to the world/camera space by

$$\begin{aligned} R_{1t} &= H_{1t} / \|H_{1t}\| \\ R_{2t} &= H_{2t} / \|H_{2t}\| \\ R_{3t} &= R_{1t} \times R_{2t} \\ T_t &= 2H_{3t} / (\|H_{1t}\| + \|H_{2t}\|) \end{aligned} \quad (15)$$

The 3D pose for the component can then be obtained though a transformation based on the computed R_t, T_t, and the reference normal N_0.

4.7. Discussion

The homography decomposition method takes two input frames and provides two physically possible solutions, one

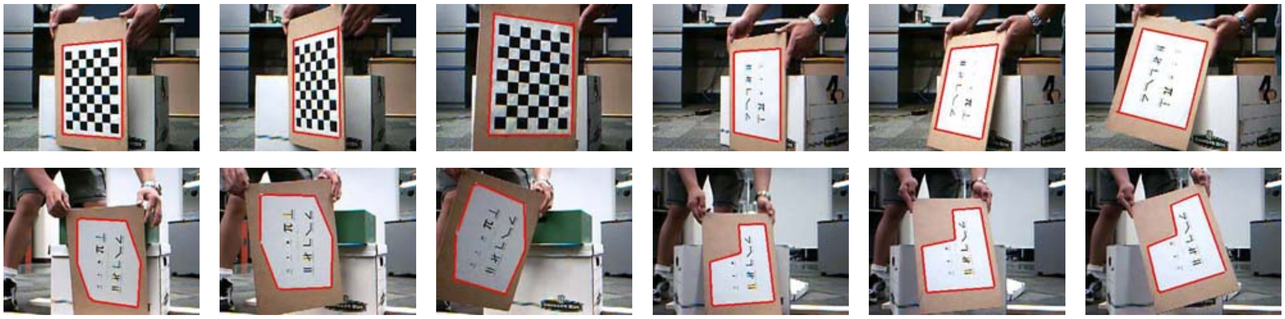

Figure 2. Tracking examples for a checker board, a rectangular letter board, a hexagonal letter board and a concave letter board. Tracked components are shown with red contours. Tracking is based on both point features and line features.

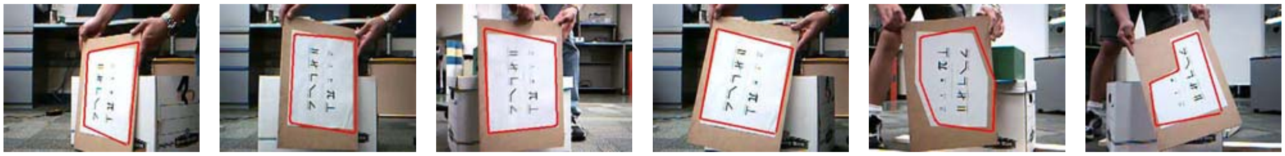

Figure 3. Tracking failures when boundary correction (line features) is disabled. The failures are caused primarily by either accumulated feature position error or accumulated parameter estimation error.

correct and the other incorrect. Using more pairs of frames, the correct solution could potentially be selected by checking the consistency among solution candidates [4]. But this may be difficult in practice in that the incorrect solution may also have good consistency, especially when there is no significant motion between the frames. In comparison, the WINEP method is based on all the observations up to the current frame, and guarantees a unique optimal solution.

Since the WINEP estimation is recursive such that at each time step only the current observation is used to update the estimation, this method does not increase computation with increasing number of frames. Instead, the conventional nonlinear optimization method will have to estimate more parameters with more input frames, so usually it's not possible to get a globally optimal solution. In comparison with the homography decomposition method, the computational cost for WINEP is higher because it needs to maintain a distribution of the normal parameters.

The homography decomposition and nonlinear optimization methods do not maintain a distribution of the normal parameters as WINEP does, so it's hard to impose prior knowledge onto the two conventional methods.

The input features to WINEP can be any point/line features either inside the component or on the component boundary, as long as these features can be matched across all the concerned images. We only use features on the boundary in this paper because in our experiments boundary features are more robust than point features inside the component.

The WINEP method will also apply to a set of static images (as opposed to videos), as long as feature correspondence can be obtained among the images.

5. Experiments

To evaluate our work, we collected 8 videos with each containing a moving object. The moving objects include a checker board (Dataset1), a rectangular letter board (Dataset2-Dataset6), a hexagonal letter board (Dataset7) and a concave letter board (Dataset8).

5.1. Tracking Results

In the test videos, the boundary of the interest component is tracked over time by our hybrid tracker. Some typical tracked frames from the videos are shown in Fig. 2.

Our hybrid tracker incorporates boundary information (line features) with point features. To demonstrate the importance of integration of boundary information, we also tested our tracking algorithm where the boundary correction step is disabled. That is, only point features are used. This test was done for two cases, (i) the same features are tracked over time, and (ii) features are detected at each frame and tracked only in the next frame. In case (i), the current boundary can be mapped from the reference boundary, by a homography transformation calculated based on the features in the current frame and the reference frame. In case (ii), the current boundary can be obtained the same way, except that the homography transformation has to be accumulated by transformations calculated from features between each pair of adjacent frames.

In either case (i) or (ii), tracking only point features worked fine for the checker board, because it is highly-textured and the corner points are very salient. But for all the other videos, tracking only point features was obviously not sufficient. Some failed tracking frames are shown in

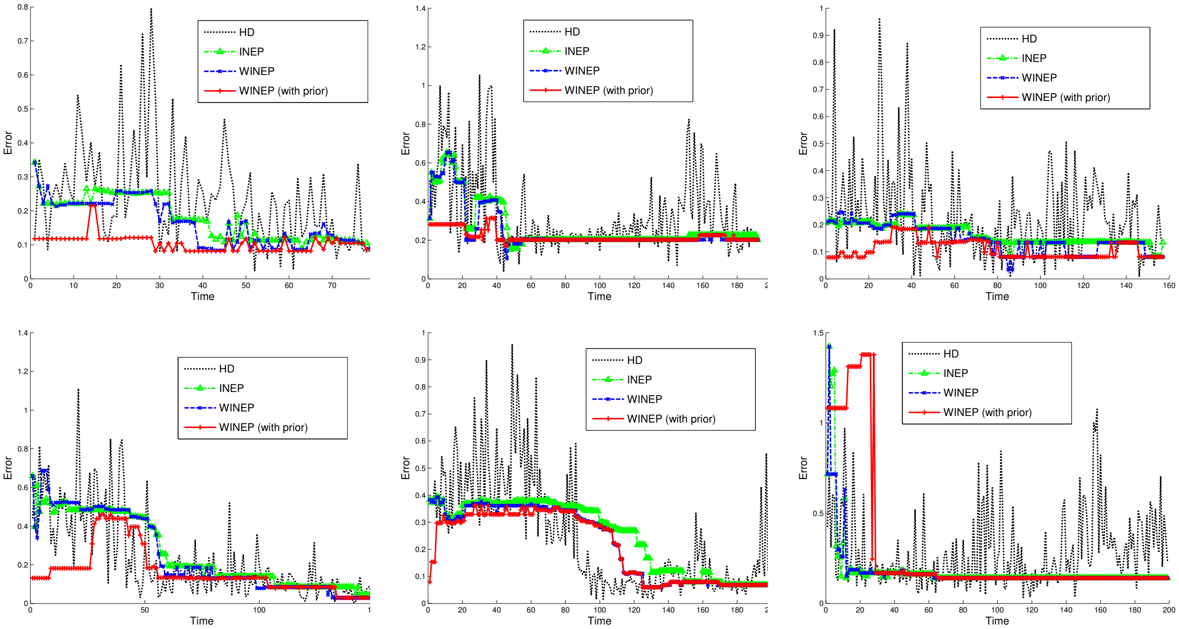

Figure 4. Normal estimation errors (best viewed in color). The error is computed as the 2-norm of the difference between the estimated normal and the ground truth normal. Due to page limit, we only show the estimation errors for Dataset 2,3,5,6,7 and 8. Since the observations in these low quality videos are noisy and the number of features used for pose estimation is low, the estimation results for the HD method are very unstable. But our estimation methods give very robust results, because they maintain a distribution of the normal parameters (INEP), assign importance weights to observations (WINEP), and incorporate prior knowledge (WINEP with prior). In the last figure, the WINEP (with prior) method gives worse estimation in the beginning because the concave letter box is not centro-symmetric.

Fig. 3. The primary reason for failures in case (i) is that features drifted slowly away from the correct positions. The failures in case (ii) are caused primarily by accumulated parameter estimation errors. It's also worth noticing in case (i) some features may be lost in a long sequence of images.

5.2. Normal Estimation Results

The components are tracked by the hybrid tracker, and the corner points on the component boundary are the input features to the HD and WINEP methods.

We obtained the ground truth data of the component normals for the 8 datasets from two laser rangefinders (horizontal and vertical). Because the camera was manually aligned with the laser sensors, we expect a small error of the ground truth data.

We compare the WINEP and HD methods by measuring the estimation errors. The error is computed as the 2-norm of the difference between the estimated normal and the ground truth normal. In all our experiments, the parameters in Eq. 13 and Eq. 14 were set as $\alpha_1 = 5$, $\alpha_2 = 5$, $\alpha_3 = 0.5$ and $\alpha_4 = 1$.

While WINEP always gives a unique solution, the HD method in general provides up to two physically possible solutions. This ambiguity can be resolved using multiple frames by assuming consistent poses between adjacent frames [4]. But in practice, when the motion between the frames is small, or the observations are noisy, it is very hard to choose the correct solution, either because the consistency of the correct solution may not be guaranteed or because the incorrect solution gives consistency as good as the correct one.

To show the robustness and accuracy of WINEP, we intentionally chose the solutions that are closer to the ground truth data for the HD method. Note that this is unfair to WINEP, because in practice we generally don't have knowledge about the ground truth data. But even so, the WINEP method demonstrated better performance. Fig. 4 shows the comparisons of normal estimation errors of the HD and WINEP methods. In these comparisons, we tested the special case of WINEP with constant $\lambda = 1$ in Eq. 13, referred as "INEP". We also tested WINEP with centro-symmetry priors of the component shapes, referred as "WINEP (with prior)". The quantitative estimation results are summarized in Table 1, where on average the estimation error decreases from HD, to INEP, to WINEP, to WINEP (with prior). WINEP runs in near real-time in our experiments.

6. Conclusion and Future Work

We have presented a new method to robustly track 3D planar objects and estimate their poses in an image sequence. The tracking algorithm incorporates boundary in-

Table 1. Normal Estimation Results

Error	HD	INEP	WINEP	WINEP(prior)
Error-1	0.2835	0.1315	0.1173	0.0279
Error-2	0.2402	0.1760	0.1643	0.1042
Error-3	0.3360	0.2651	0.2536	0.2176
Error-4	0.3245	0.3570	0.3519	0.2773
Error-5	0.2314	0.1623	0.1495	0.1096
Error-6	0.2316	0.2570	0.2466	0.1629
Error-7	0.2490	0.2507	0.2232	0.2090
Error-8	0.2899	0.1786	0.1741	0.2932
Average	0.2733	0.2223	0.2101	0.1752

formation (line features) into point feature tracking. Compared with point features, line features are generally more robust to image noise and can generally provide more accurate position estimation, which allows our tracking method to work better than using only point features.

The normals for the planar objects are estimated based on the tracked boundary locations. We maintain a distribution over the normal parameters by dynamically updating the distribution using the most recently observed boundary features. This method provides an optimal solution based on all the past observations. In the case where the observations are noisy or the number of available features is small, this method gives more robust and accurate results than the conventional homography decomposition method.

In future work, we will investigate how 3D models can be constructed for objects composed of planar surfaces and how the constructed model can in turn help tracking and pose estimation.

acknowledgment. This work has taken place in the Intelligent Robotics Lab at the Artificial Intelligence Laboratory, The University of Texas at Austin. Research of the Intelligent Robotics lab is supported in part by grants from the Texas Advanced Research Program (3658-0170-2007), and from the National Science Foundation (IIS-0413257, IIS-0713150, and IIS-0750011).

References

[1] A. Agarwal, C. Jawahar, and P. Narayanan. A survey of planar homography estimation techniques. *Technical Reports, International Institute of Information Technology, Hyderabad, 2005*. 3

[2] F. Bourel, C. Chibelushi, and A. Low. Robust facial feature tracking. *BMVC*, 2000. 2

[3] J. Canny. A computational approach to edge detection. *PAMI*, 1986. 3

[4] D. Cobzas, M. Jagersand, and P. Sturm. 3D SSD tracking with estimated 3D planes. *Journal of Image and Vision Computing*, 2009. 2, 6, 7

[5] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *PAMI*, 2003. 2

[6] R. Duda and P. Hart. Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 1972. 1, 3

[7] C. Engels, H. Stewenius, and D. Nister. Bundle adjustment rules. *Photogrammetric Computer Vision*, 2006. 2

[8] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981. 2

[9] J. Gall, B. Rosenhahn, and H. Seidel. Drift-free tracking of rigid and articulated objects. 2008. 2

[10] I. Gordon and D. Lowe. What and where: 3D object recognition with accurate pose. *Lecture Notes in Computer Science*, 2006. 2

[11] J. Guerrero and C. Sagüés. Robust line matching and estimate of homographies simultaneously. *Pattern Recognition and Image Analysis: First Iberian Conference*, 2003. 2, 3, 4

[12] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003. 2, 3, 4

[13] M. Isard and A. Blake. CONDENSATION - conditional density propagation for visual tracking. *IJCV*, 1998. 2

[14] B. Kuipers. Drinking from the firehose of experience. *Artificial Intelligence In Medicine*, 2008. 2

[15] D. Kumar and C. Jawahar. Robust homography-based control for camera positioning in piecewise planar environments. *Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP)*, 2006. 2

[16] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 2

[17] Y. Ma. *An invitation to 3-D vision: From images to geometric models*. Springer Verlag, 2004. 2, 3, 4, 5

[18] M. Pressigout and E. Marchand. Real-time 3d model-based tracking: Combining edge and texture information. *ICRA*, 2006. 2

[19] J. Shi and C. Tomasi. Good features to track. *CVPR*, 1994. 1, 2

[20] S. Tran and L. Davis. Robust object trackinng with regional affine invariant features. *ICCV*, 2007. 2

[21] L. Vacchetti, V. Lepetit, and P. Fua. Combining edge and texture information for real-time accurate 3d camera tracking. *The 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality*, 2004. 2

[22] L. Zeinik-Manor and M. Irani. Multiview constraints on homographies. *PAMI*, 2002. 2

[23] T. Zinsser, C. Grassl, and H. Niemann. Efficient feature tracking for long video sequences. *Pattern Recognition: 26th DAGM Symposium, Tübingen, Germany*, 2004. 2