The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Multiple Plane Tracking using Unscented Kalman Filter

Visesh Chari and C. V. Jawahar

Abstract— **An important pre-requisite for many tasks like Visual Servoing and visual SLAM is the task of tracking the underlying features. The use of planar features for these purposes has gained importance recently. Complementing current planar tracking works in the robotics literature, which use multiple features, we formulate the tracking problem using *multiple planes*. Inspired by the maturity in understanding of geometric quantities like the homography in computer vision, we develop a system based on the Unscented Kalman Filter (UKF) that localizes the camera and estimates the plane parameters of a scene, using homographies as measurement. Homographies are estimated using tracked feature points. We show that this framework provides significant robustness and stability to the system under significant changes of illumination, occlusion etc. Finally, we also propose a Convex optimization based solution for the initialization of this system, which is capable of producing globally optimal estimates, and is a useful algorithm in its own right. Several synthetic and real results are presented to demonstrate the efficacy of our approach.**

I. Introduction

The problem of recovering the position and orientation of a target object in every frame of a video is called visual tracking. Tracking can be done in both 2D [1] and 3D [2], and is useful for various tasks like car following [3], SLAM [2], PBVS [4] etc.. In 3D tracking, the *pose* (rotation+translation) of the object w.r.t. a specified coordinate system is computed for every frame. When the video contains a single rigid object, the pose of the object and the pose of the camera are the same. When the object consists of multiple planes, tracking its pose corresponds to estimating the homography of these planes w.r.t. a reference coordinate system in every frame. In this paper, we present an approach using Unscented Kalman Filters (UKF) that combines the tracking of multiple planes with the tracking of pose to achieve robustness over tracking each of the planes individually.

Several approaches to tracking one or several planes have been proposed in the past [1], [5] for purposes like IBVS [6], reconstruction [7], augmentation [8], patch-based SLAM [9] etc.. In the robotics literature, tracking has been modeled using various filters, most notable among them being the Kalman filters [10]. While the traditional Kalman filter is useful only for linear systems with Gaussian noise, the Extended Kalman Filter (EKF) allows to model non-linear systems using a first order approximation. However, since homographies are highly non-linear quantities, we use an Unscented Kalman Filter (UKF) [11] in our work, which approximates non-linear functions better than the EKF.

In this paper, we track scene plane homographies and camera pose simultaneously in a UKF framework. Each individual homography might be computed using robust methods [1]. By combining information from multiple homographies for pose computation, we show how the resulting pose tracking is not only accurate, but also serves to *correct* the underlying homographies in case of errors due to illumination changes, occlusion etc. Section II presents works in the literature that our relevant to our case. Section III outlines the proposed tracking framework, using UKF [11]. Section IV discusses implementation issues. Experimental results on challenging sequences (Section V) under varying conditions of pose and illumination show the robustness of the tracker. Finally, we summarize with elucidation of future avenues for research in Section VI.

II. Related Work

Recently, tracking of multiple planes in a scene has attracted interest [12], [13]. The main advantage of tracking multiple planes is the fact that, under the assumption of a static rigid scene, various constraints based on the scene geometry may be added to simplify the tracking formulation [13] or to increase the accuracy of tracking [12].

In [13], the authors make use of the fact that homographies induced by multiple planes between two views of a camera are embedded in a lower dimensional space [14], to reduce the number of parameters required to track multiple planes. Another approach is used by [12], who formulate the tracking of multiple planes as a problem of estimating the pose parameters of the camera *given* the parameters of the plane. Tracking is done using an inverse compositional approach that uses a gradient descent optimization function over the pose space for the tracking. Alternatively, the authors of [15] use the same relationships to construct a set of linear equations with pose as the unknowns. These equations are then solved using singular value decomposition to obtain the least squared pose estimate for every frame.

While in [13], the errors in one plane may affect the tracking of others because of the coupled estimation strategy, the algorithm of [12] is not guaranteed to be robust to illumination change. In this paper, we use an approach close to [12] to compute homographies and pose simultaneously. We however incorporate information from multiple planes into a UKF framework, to ensure more robust tracking.

In [9], the authors present a MonoSLAM system that uses the EKF to track the pose of the camera and the 3D coordinates of certain features in the environment. Additionally they also estimate the normals (orientation) of locally planar features. It is different from our work in the sense that pose

Visesh Chari is a PhD student at INRIA Rhône Alpes, Grenoble, France visesh.chari@inrialpes.fr

C. V. Jawahar is a Professor at the International Institute of Information Technology, Hyderabad, India jawahar@iiit.ac.in

978-1-4244-6676-4/10/$25.00 ©2010 IEEE

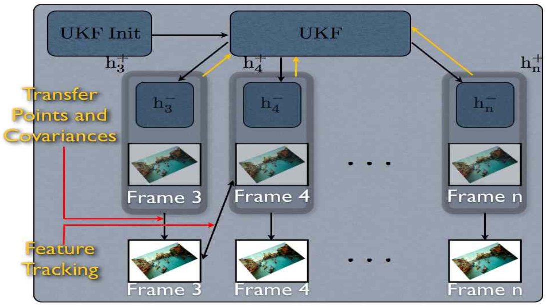


Fig. 1. The UKF tracking algorithm.

and plane tracking are not *coupled* together as in our case, but estimated separately. As a result, we have been able to show much more robustness in tracking.

Finally, the use of UKF is also not new in the robotics community. In [16], the authors present a system for SLAM based on the square root Unscented Kalman Filter (srUKF) that runs approximately at the same order and speed as the traditional EKF based SLAM, while providing much better accuracy.

III. Tracking Framework

We develop the tracking framework in the following section. Our framework employs plane and pose parameters as the state vector to be estimated, while homographies are treated as measurements. This provides both robustness and stability. Robustness because, the system is not affected by the tracking of an individual feature beyond a point, so long as the homography can be effectively estimated, for which there are many methods available [17]. Stability because, even with the complete loss of tracking for a particular plane, the *other* planes in the environment can contribute to both pose tracking for the camera and homography tracking for the missing plane.

A. Parametrization and Motion Model

In the presence of multiple planes (say m), we have multiple instances of homography $H_t = \{h_t^1, \ldots, h_t^m\}$ at time t that may be computed from point correspondences. Let us recall that the homography relating two views of a plane is related to the geometry of the views through the following relation

$$h_t^i \sim K \left[\begin{array}{cc} R_t & \frac{t_t n^{i\top}}{d^i} \end{array} \right] K^{-1} \tag{1}$$

where $\left[\begin{array}{cc} R_t & t_t \end{array} \right]$ represents the relative pose between the first and t^{th} views, n^i represents the plane normal at distance d^i from the coordinate system attached to the first view, and K represents the internal parameters of the camera.

In this case, assuming a rigid static scene and that m, K are known apriori, it is possible to estimate the pose of the camera along with the normals of each of the planes. Thus the parameters to be estimated are represented by the vector $s_t = \left[\begin{array}{ccccc} p_t & n_t^1 & d_t^1 \ldots & n_t^m & d_t^m \end{array} \right]$, where $p_t = \left[\begin{array}{cccccc} e_t^1 & e_t^2 & e_t^3 & t_t^1 & t_t^2 & t_t^3 \end{array} \right]$ is the pose of the object/camera and $n^i = \left[\begin{array}{ccc} n^{i_1} & n^{i_2} & n^{i_3} \end{array} \right]$ ($\| n^i \| = 1$) is the plane normal. Here, rotation is represented using Euler angles. Thus for m planes a total of $3m + 7$ parameters need to be tracked, with polar representation for normals. If $\hat{s}_t$ represents our current estimate of the state vector, then change owing to inter-frame displacement can be written as

$$\hat{\mathbf{s}}_t = \hat{\mathbf{s}}_{t-1} + \Delta\hat{\mathbf{s}}_t. \tag{2}$$

assuming a Markovian model with Brownian motion.

Given that at each iteration of the algorithm, we estimate the homographies induced by each plane as $\hat{H} = \{\hat{h}_t^1, \ldots, \hat{h}_t^m\}$ the optimal estimate of the state vector $\hat{\mathbf{s}}_t$ can be calculated as

$$\hat{\mathbf{s}}_t^* = \arg\min_{\hat{\mathbf{s}}_t} \mathcal{F}(\hat{H}_t - f(\hat{s}_t)) \tag{3}$$

where $\mathcal{F}$ is a suitable distance measure, f is the function given in Equation 1, and H_t is a robustly estimated homography. In the current work, $\mathcal{F}$ is the scaled Euclidean distance which sets the last element of the homography matrix to 1.

In a filtering framework, the search space for the optimal solution at every iteration is restricted to the noise covariance around the prior model. In our case, the two types of errors: process noise and measurement noise are detrimental to this search, and correspond to errors in the values of pose and plane parameters, and errors in the estimation of homographies. It has been demonstrated earlier that a Gaussian assumption for process noise is suitable for tracking the pose of the camera [2]. For homographies, the assumption of a Gaussian noise source is only an approximation of the true noise [18], to the first order. Still it is desirable to represent both noises using a Gaussian model because

- In practice, a Gaussian approximation for noise in homography has been found to be a useful model for cases when the underlying images are captured from similar view points (like successive frames of a video).
- a Gaussian model allows us to employ the widely used class of Kalman filters [11], that are a close approximation to the Bayesian filter, which is considered the most optimal estimator of states, given data.

Since the state transition and observation functions are highly non-linear in nature (Equation 1), and since a first order approximation to the observation noise is used, the Extended Kalman Filter (EKF) seems to be a poor choice for the filtering. It is for these reasons that an Unscented Kalman Filter [11] is employed in the current work.

B. Multi-Plane Tracking

We now define the process and measurement functions that are used in the Unscented Kalman Filter, under the assumption that m planes are being tracked.

$$\begin{aligned} s_t &= s_{t-1} + \mathcal{N}(0, R) && (4) \\ \left[\begin{array}{ccc} h_t^1 & \ldots & h_t^m \end{array} \right] &= f(\mathbf{s}_t \;, \; v_t) && (5) \\ v_t &= \mathrm{diag}(\Lambda_{h_t^1}, \ldots, \Lambda_{h_t^m}) && (6) \end{aligned}$$

where the measurement noise v_t is defined as a collection of the covariances of the individual homographies, as detailed in Section III-C. It assumes the independent measurement of the underlying homographies. In order to initialize the filter, we obtain a rough estimation of the plane normals by decomposition of the homography between the first two frames for each plane, followed by a process called *Coordinate Normalization*, detailed in Section IV-C. The whole method is summarized in Algorithm 1, and in Figure 1.

C. Cues From Multiple Planes

In order to represent the true process faithfully, we need a noise covariance that is close to the true covariance in the measurement process. Assuming that the tracked feature points are perturbed with Gaussian noise, first order uncertainty analysis may be used to measure the variance of the computed homography as a function of the variance of the individual features [18].

$$\Lambda_h \quad = \quad JSJ, \quad J = -\Sigma_{i=1}^{n} \frac{u_k u_k^\top}{\lambda_k} \tag{7}$$

where (u_k, λ_k) are the eigenvectors and eigenvalues of the matrix A in the objective function $Ah = 0$, which is solved using singular value decomposition to obtain the least square homography solution h (Please refer [18] for details). The value of S is obtained from A and the covariances of the individual features, which represent the uncertainty in the feature matching process.

We characterize the uncertainty in feature matching by relating it to the inverse covariance matrix of the image gradients in their local neighborhoods [19].

$$\Sigma_{x_i,y_i} = \begin{bmatrix} \frac{\partial^2 I(x_i,y_i)}{\partial x^2} & \frac{\partial^2 I(x_i,y_i)}{\partial x \partial y} \\ \frac{\partial^2 I(x_i,y_i)}{\partial x \partial y} & \frac{\partial^2 I(x_i,y_i)}{\partial y^2} \end{bmatrix}^{-1} \tag{8}$$

where $I(x_i, y_i)$ represents the image under consideration.

IV. Implementation Issues

In this section, we discuss some of the major implementation issues that are of importance to our algorithm.

A. Initialization

We use Singular Value Decomposition (SVD) of homography to provide solutions [20], [21] of camera pose and plane normals, for initializing the tracker. We have observed that initialization from either algorithm suffices for our case.

B. Data Normalization and Covariance Transfer

Two of the important steps in this algorithm are the normalization of image correspondences before the computation of the homography, and the transfer of covariance of the image features across a homography from initial frame to the previous frame. We describe the equations involved in these transformations below

1) Data Normalization: The computation of homography requires a numerical conditioning [22] of the data, which needs to be included while computing covariances. When input points are transformed by affine transformations T_1 and T_2 respectively, the resultant homography is transformed as $T_2 * H_{\text{actual}} * \text{inv}(T_1)$. Thus covariances for the ith point in the kth frame and jth plane are computed as

$$\begin{aligned} \{\sigma_i^{\text{est}}\}_1^j &= J_{x1}^{\text{est}} \{\sigma_i\}_1^j (J_{x1}^{\text{est}})^\top && (9) \\ \{\sigma_i^{\text{est}}\}_k^j &= J_{xk}^{\text{est}} \{\sigma_i\}_k^j (J_{xk}^{\text{est}})^\top && (10) \\ \Lambda_{\{h\}_k^j} &= J_{hk}^{\text{est}} \Lambda_{\{h^{\text{est}}\}_k^j} (J_{hk}^{\text{est}})^\top && (11) \end{aligned}$$

where $J_{x1}^{\text{est}} = \partial\{x_i^{\text{est}}\}_1^j / \partial\{x_i\}_1^j$, $J_{xk}^{\text{est}} = \partial\{x_i^{\text{est}}\}_k^j / \partial\{x_i\}_k^j$ and $J_{hk}^{\text{est}} = \partial\{h\}_k^j / \partial\{h\text{est}\}_k^j$, and $^{\text{est}}$ is used to denote quantities in the transformed space.

2) Covariance Transfer: Once the pose estimate for the previous frame has been established, wrong measurements (homographies) are corrected by simply assigning the homographies and associated covariances, values of the posterior measurement and noise estimates from the previous iteration of the UKF tracker. This corresponds to the first step within the loop in Algorithm 1.

Once homographies are corrected, the next step is to correct feature points to ensure robust tracking. The transferred variance of the ith feature point in the kth frame and jth plane may be obtained as [18]

$$\begin{aligned} \{\sigma_i\}_k^j &= \begin{bmatrix} B & h_k^j \end{bmatrix} \begin{bmatrix} \Lambda_k^j & \mathbf{0} \\ \mathbf{0} & \{\sigma_i\}_1^j \end{bmatrix} \begin{bmatrix} B^\top \\ h_k^j \end{bmatrix} && (12) \\ \mathbf{x} &= \begin{bmatrix} (\{x_i\}_1^j)^\top & 1 \end{bmatrix}, B = \begin{bmatrix} \mathbf{x} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{x} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{x} \end{bmatrix}_{3\times 9} && (13) \end{aligned}$$

Finally, in order to convert homogeneous coordinates to in-homogeneous coordinates, a scaling operation is imposed on the transferred points, where each point is divided by its 3rd coordinate to make it 1. The corresponding change in the covariance matrices is given by

$$\begin{aligned} \{\sigma_i^{2\times 2}\}_k^j &= \nabla f \{\sigma_i\}_k^j \nabla f^\top && (14) \\ \nabla f &= 1/W^2 \begin{bmatrix} W & 0 & -X \\ 0 & W & -Y \end{bmatrix} && (15) \end{aligned}$$

where $\{x_i\}_k^j = [X\ Y\ W]^\top$. Since Equation 14 is a first order approximation of the Taylor series expansion around the in-homogeneous point, the approximation can be poor for large values of $(\frac{X}{W}, \frac{Y}{W})$. To avoid this problem, we maintain full 3D covariances at all times.

C. Coordinate Normalization

Each decomposition by the algorithms of Faugeras [20] and Zhang [24] produce estimates of $\{R, t, n\}$ assuming a coordinate system in which the perpendicular distance between the origin and the plane in consideration is 1. We call the process of adjusting the perpendicular distances,"coordinate normalization".

Algorithm 1 The UKF tracking algorithm

```
$\{x_i\}_1$ = EF($I_1$). // Extract Features
$\{x_i^*\}_2$ = TF($I_1$, $I_2$, $\{x_i\}_1$). // Track Features
$(\{x_i\}_1^{1\ldots m}, \{x_i^*\}_2^{1\ldots m})$ = SP( $I_1$, $I_2$, $\{x_i\}_1$, $\{x_i^*\}_2$).
$\{\sigma_i\}_1^{1\ldots m}$ = CC($I_1$, $\{x_i\}_1^{1\ldots m}$). // Compute Covariances
$\{\sigma_i^*\}_2^{1\ldots m}$ = CC($I_2$, $\{x_i\}_2^{1\ldots m}$).
$\{h^*, \Lambda\}_2^{1\ldots m}$ = CH$(\{x_i, \sigma_i\}_1^{1\ldots m}, \{x_i^*, \sigma_i^*\}_2^{1\ldots m})$.
$(R_2, \{, t^*, n^*\}_2^{1\ldots m})$ = DH$(\{h^*\}_2^{1\ldots m})$.
$(t_2, \{n, d\}^{1\ldots m})$ = CN$(\{, t^*, n^*\}_2^{1\ldots m})$ (Section IV-C).
Initialize UKF tracker.
for $(k = 3, \ldots, n)$ do
    $\{h, \Lambda\}_{k-1}^{1\ldots m}$ = HM$(R_{k-1}, t_{k-1}, \{n\}^{1\ldots m}, \{d\}^{1\ldots m})$ (Optional).
    $(\{x_i\}_{k-1}^{1\ldots m}, \{\sigma_i\}_{k-1}^{1\ldots m})$ = TPC$(\{h, \Lambda\}_{k-1}^{1\ldots m}, \{x_i\}_1^{1\ldots m})$ (Section IV-B) (Optional).
    $\{x_i^*\}_k^{1\ldots m}$ = TF($I_{k-1}$, $I_k$, $\{x_i\}_{k-1}^{1\ldots m}$) [23].
    $\{h^*, \Lambda^*\}_k^{1\ldots m}$ = CH$(\{x_i, \sigma_i\}_1^{1\ldots m}, \{x_i^*, \sigma_i^*\}_k^{1\ldots m})$ (Section IV-B).
    $\{\Lambda\}_k^{1\ldots m}$ = RW$(\{\Lambda^*\}_k^{1\ldots m})$. (Section IV-D).
    $(R_k, t_k, \{n, d\}^{1\ldots m})$ = UKF$(\{h, \Lambda\}_k^{1\ldots m}\})$.
end for
// SP, CN - Segment Planes, Coordinate Normalization
// CH, DH - Compute, Decompose Homographies
// HM, RW - Homography Measurement, Robustness Weights
// TPC - Transfer Points and Covariances
```

Let the solution of translation obtained by decomposing h_2^j of the jth plane be denoted by t^j. Thus the actual translation vector is represented by $t = t^j d^{*j}$, where d^{*j} is the optimum of an objective function. Since, estimates obtained from the various planes must converge, we are interested in the optimum values $\left[\begin{array}{c} d^{*1}, d^{*2}, \ldots, d^{*m} \end{array}\right]$ such that

$$\left[\begin{array}{c} d^{*1}, \ldots, d^{*m} \end{array}\right] = \min \sum_{l=1}^{m} \sum_{j=1}^{m} \|t^j d^j - t^l d^l\|_2 \tag{16}$$

To obtain a global minima by a Convex reformulation of Equation 16, we introduce a new set of variables (t^*) which represent the *actual* translation of the view up to scale. The modified functions now become $f_j(t^*, d^j) = \|t^* - t^j d^j\|_2$. Looking closely at f_j we find

$$f_j(t_i^*, d^j) = \|t^* - t^j d^j\|_2 = \| \left[\begin{array}{cc} t^* & d^j \end{array}\right] \left[\begin{array}{c} 1 \\ -t^j \end{array}\right] \|_2 \tag{17}$$

Re-framing the parameter vector as $q_j = \left[(t^*)^\top \ d^1 \ldots d^m\right]^\top$, we can rewrite the above function as $f_j = \|A_j q + b\|_2$, for some A_j's, which represents a convex function in the variables [25]. In order to obtain an optimal result, we finally introduce another variable γ that puts a bound on the maximum value taken by each of the m convex functions. This results in a function to be minimized with convex constraints, defined by the following set of equations

$$\begin{aligned}
\min_{x} \quad & \gamma && (18)\\
\text{such that} \quad & f_j(q) = \|A_j q + b\|_2 < \gamma && (19)\\
& q \in \mathbb{R}^{3+m}, && (20)\\
& A_j \in \mathbb{R}^{j,3+m}, b = \mathbf{0}_{1,1} && (21)
\end{aligned}$$

There are excellent mathematical packages like SeDuMi[26] for obtaining *globally optimal* solutions for such cases.

D. Robustness

Outliers in the data or input do not have a representation in the estimation process of Kalman filters. Thus, such a tracker will be sensitive to bad estimates of homography. The solution to this problem is to *detect* and *exclude* such measurements from the estimation process. By not including outlier measurements, we ensure an accurate estimate of the state vector, and with an accurate estimate of the state vector, we can obtain an accurate *replacement* of the outlier measurements.

A bad estimate of the homography may be obtained due to either incorrect feature correspondences, or due to bad conditioning of the underlying homography estimation process which may be the result of a smaller set of feature correspondences than desired. Assuming an algorithm like RANSAC [17] is used to weed out both outlier correspondences, and badly conditioned homographies, we are only left with the case when we need a measure of the *accuracy* of the feature correspondence.

In order to get an estimate of *how reasonable*a particular feature correspondence is, we use the *Normalized Cross Correlation (ncc) function* [17], and incorporate it as an inverse scale factor in Equation 8 to use this information.

With the above robustness measures, we may introduce suitable weights into the estimation process. When in the presence of illumination or occlusion, however, the homography estimation may fail even with the above robustness heuristics. It is here that the Kalman Filter takes over to *predict* the *maximum likelihood* observation (homography) based on the measured state vector values (pose and plane parameters).

V. Experiments and Results

We conducted a series of synthetic and real experiments to systematically verify each of the various components of our algorithm.

a) **Initialization***:* The first experiment pitted the coordinate normalization algorithm (Section IV-C) against a standard SVD based solution. This experiment was aimed at testing the accuracy and error resilience of the initialization. Ten random planes with errors were projected on to two random cameras, and resulting homographies were decomposed to obtain initial estimates. This was then passed to both the algorithms to generate estimates of the perpendicular distances. Figure 4 shows the resultant errors in estimates of the perpendicular distances. The convex optimization solution has a very low root mean square error as compared to the SVD based solution (Figure 4(a) is shown in log space). Also, the convex optimization solution is *much* more resilient to error than the SVD based solution.

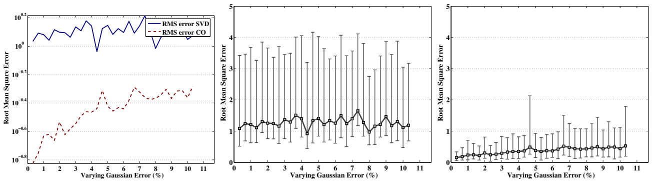
b) **Parameter Estimation***:* To test the parameter estimation process, a set of points on 3 random planes were generated. Then, image points formed by 200 randomly generated cameras were perturbed with noise. The homographies computed were then fed into an appropriately initialized UKF tracker. The initialization of the plane parameters was *highly* erroneous because of the noisy homography. Figure 5 illustrates the results of UKF based tracking of the plane and pose parameters. As expected, the estimation of plane parameters shows a sharp decrease over time, suggesting that even with a large initialization error in a plane parameter due to noisy homography, the other planes contribute to a decrease in error by keeping the pose error, which in turn "stabilizes" the value of the plane parameters. This experiment verifies our claim that other planes contribute to the overall robustness and stability of the system.

c) **Real Data***:* Finally, three experiments on real data were conducted to illustrate the application to pose estimation and the robustness of the tracker to conditions like change in perspective, change in illumination and occlusion. Figure 2 shows pose estimation on a sequence shot along a corridor. Features were mainly found on the corners of the doors, and on the ceiling. Ground truth was generated with manually marked correspondences. As can be seen, the pose estimation is quite accurate.

The second and third experiments use 3 planes over considerable changes in perspective to test the algorithm (Figure 3, top two rows). The second experiment has large changes in illumination, while the third experiment introduces an occluding object in the video. In the second experiment, our algorithm is able to *correct* homographies wrongly estimated by the failing KLT tracker, which is then re-initialized. In the third experiment, our algorithm is able to estimate the homography of the third plane by using the normal and pose estimates obtained from the previous frames and the other planes respectively. In both cases, the simple KLT tracker fails (Figure 3 bottom two rows).

VI. Conclusion

In this paper, we proposed a tracking framework that tracks multiple planes robustly, by fusing the various homography estimates using an Unscented Kalman Filter. The framework has applications to the areas of Visual Servoing and visual SLAM. In the future, we plan to investigate extensions along this front, along with algorithms for real-time planar segmentation. To conclude, we believe that our framework has good potential for research and application to various problems in robotics.



(a)

(b) Noise variation of SVD initialization.

(c) Noise variation of convex optimization based initialization.

Fig. 4. The graphs show the superiority of the convex optimization solution over the traditional SVD based solution. Both figures (b-c) plot the root mean squared error between the estimated and ground truth parameters, along with bars to indicate the variation of the error, estimated through a Monte-Carlo test. The convex optimization solution shows excellent resilience.

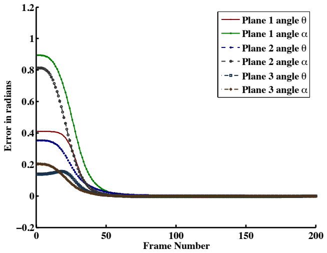


Fig. 5. The plot shows the decrease in the error of plane parameters, estimated using the UKF for a synthetic sequence. Notice how the initial estimates have large error. But because of the presence of multiple planes, the error in the plane parameters reduces very quickly (almost exponentially) once the homography covariance stabilizes.

References

[1] A. A. Hafez, V. Chari, and C. Jawahar, "Combining texture and edge planar trackers based on a local quality metric," in *IEEE Int. Conf. on Robotics and Automation, ICRA'07*, Roma, Italia, April 2007.

[2] A. Davison, I. Reid, N. Molton, and O. Stasse, "MonoSLAM: Real-Time Single Camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, June 2007. [Online]. Available: http://pubs.doc.ic.ac.uk/monoslam/

[3] S. Benhimane, E. Malis, P. Rives, and J. R. Azinheira, "Vision-based control for car platooning using homography decomposition," in *ICRA*, 2005, pp. 2161–2166.

[4] P. I. Corke, "Visual control of robot manipulators - a review," in *Visual Servoing*. World Scientific, 1993, pp. 1–31.

[5] M. Pressigout and E. Marchand, "Real-time planar structure tracking for visual servoing: a contour and texture approach," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'05*, vol. 2, Edmonton, Canada, August 2005, pp. 1701–1706.

[6] S. Benhimane and E. Malis, "Homography-based 2d visual tracking and servoing," *Int. J. Rob. Res.*, vol. 26, no. 7, pp. 661–676, 2007.

[7] C. Baillard, A. Zisserman, and O. O. England, "Automatic reconstruction of piecewise planar models from multiple views," in *In Proc. IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society Press, 1999, pp. 559–565.

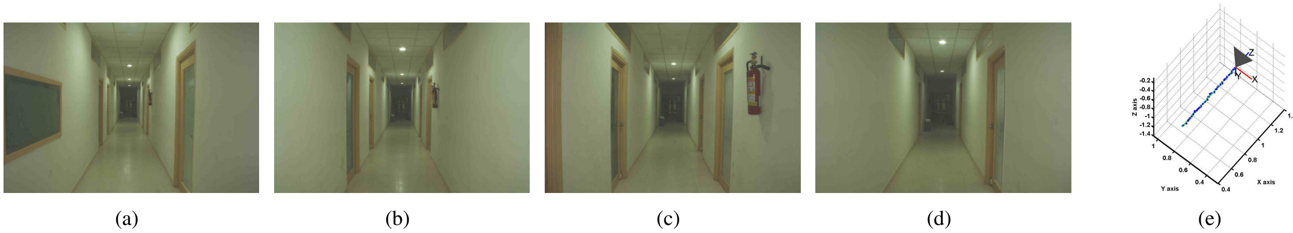
[8] G. Simon, A. Fitzgibbon, and A. Zisserman, "Markerless tracking using planar structures in the scene," in *Proc. International Symposium on Augmented Reality*, Oct. 2000, pp. 120–128. [Online]. Available: http://www.robots.ox.ac.uk/ vgg

[9] N. Molton, A. Davison, and I. Reid, "Locally planar patch features for real-time structure from motion," 2004. [Online]. Available: citeseer.ist.psu.edu/article/molton04locally.html

[10] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[11] E. A. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pp. 153–158, 2000. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=882463

[12] D. Cobzas and P. Sturm, "3d ssd tracking with estimated 3d planes," in *Image and Vision Computing, CRV Special Issue*, 2005.



(a) (b) (c) (d) (e)

Fig. 2. A sequence with the camera moving headlong into a corridor. The first four figures are images of the sequence, and the final figure shows the path of the camera plotted in 3D. The blue path is the computed trajectory whereas the green one is the ground truth. "Jitters" occur because of the Brownian motion modeling, although the path is accurately predicted.

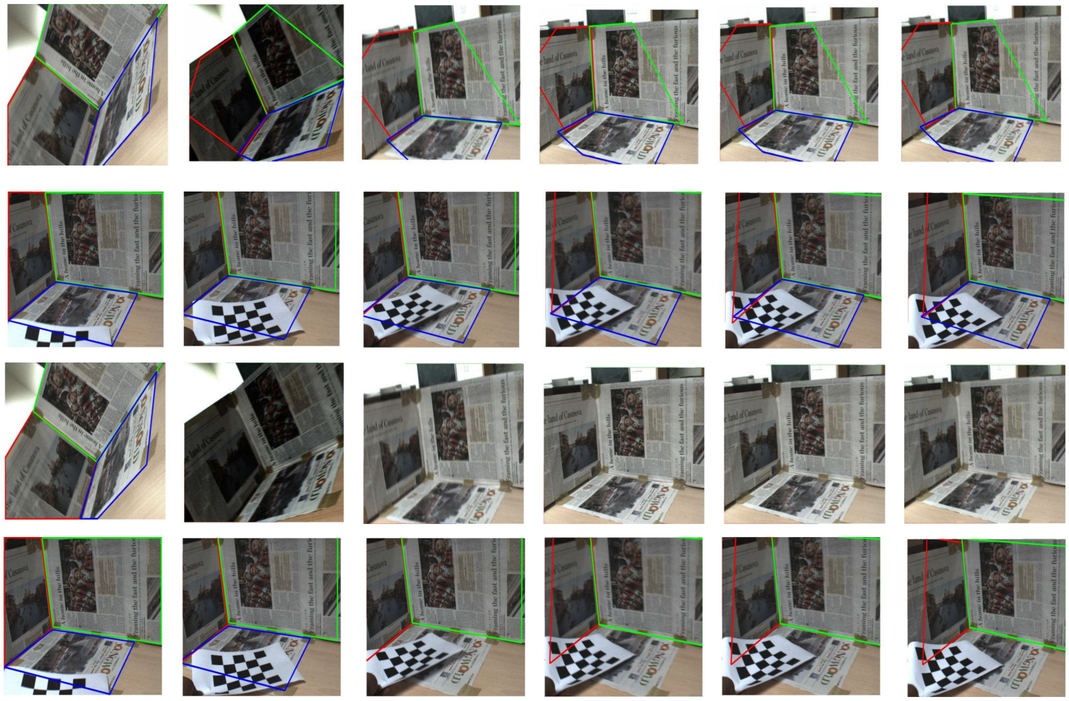


Fig. 3. (Top two rows): Results of UKF tracking on two sequences. Figures in the first row show results when there is considerable change of perspective, illumination and even blur in some cases. Figures in the second row show the case when there is a large occlusion of one of the planes. Our tracker still robustly tracks the plane, showing that coupled estimation of the planes can be *much* more robust than estimation of the planes individually. (Bottom two rows): Results of tracking for the KLT tracker. In the first set of figures (first row), the KLT tracker looses all its planes because of the illumination change. In the second set (second row), the tracker loses one plane because of the occlusion.

[13] O. Kähler and J. Denzler, "Rigid motion constraints for tracking planar objects," in *Annual Symposium of the German Association for Pattern Recognition, DAGM*, 2007.

[14] A. Shashua and S. Avidan, "The rank 4 constraint in multiple (≥ 3) view geometry," in *ECCV '96: Proceedings of the 4th European Conference on Computer Vision-Volume II*. London, UK: Springer-Verlag, 1996, pp. 196–206.

[15] G. Simon and M.-O. Berger, "Pose estimation for planar structures," in *IEEE Computer Graphics and Applications*, 2002.

[16] S. Holmes, G. Klein, and D. Murray, "A square root unscented kalman filter for visual monoslam," *IEEE International Conference on Robotics and Automation (ICRA)*, 2008.

[17] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[18] A. Criminisi, I. Reid, and A. Zisserman, "A plane measuring device," *Image and Vision Computing*, vol. 17, no. 8, pp. 625–634, 1999. [Online]. Available: http://www.robots.ox.ac.uk/ vgg

[19] Z. Sun, V. Ramesh, and A. M. Tekalp, "Error characterization of the factorization method," *Computer Vision and Image Understanding: CVIU*, vol. 82, no. 2, pp. 110–137, 2001. [Online]. Available: citeseer.ist.psu.edu/sun01error.html

[20] O. Faugeras and F. Lustman, "Motion and structure from motion in a piecewise planar environment," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 2, no. 3, pp. 485–508, 1988.

[21] Z. Zhang, "Determining the epipolar geometry and its uncertainty: A review," *International Journal of Computer Vision, IJCV*, vol. 27, no. 2, pp. 161–195, 1998.

[22] R. Hartley, "In defense of the eight-point algorithm," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 6, pp. 580–593, Jun 1997.

[23] J. Shi and C. Tomasi, "Good features to track," in *IEEE International Conference on Computer Vision and Pattern Recognition, CVPR'94*, Seattle, Washington, June 1994, pp. 593–600.

[24] Z. Zhang and A. R. Hanson, "3d reconstruction based on homography mapping," *Proceedings of ARPA*, pp. 1007–1012, 1996.

[25] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

[26] J. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11–12, pp. 625–653, 1999, special issue on Interior Point Methods (CD supplement with software). [Online]. Available: citeseer.ist.psu.edu/sturm99using.html