Computer Vision and Image Understanding 115 (2011) 8–15

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Unscented feature tracking

Leyza Baldo Dorini [a,*], Siome Klein Goldenstein [b]

[a] UTFPR – Universidade Tecnológica Federal do Paraná, Departamento de Informática, Av. Sete de Setembro 3165, 80230-901 Curitiba, PR, Brazil
[b] Unicamp – Universidade Estadual de Campinas, Instituto de Computação, Caixa Postal 6176, 13083-971 Campinas, SP, Brazil

ARTICLE INFO

Article history:
Received 31 March 2008
Accepted 26 July 2010
Available online 4 August 2010

Keywords:
Feature tracking
Uncertainty tracking
Outlier rejection
Statistical correspondences

ABSTRACT

Accurate feature tracking is the foundation of many high level tasks in computer vision, such as 3D reconstruction and motion analysis. Although there are many feature tracking algorithms, most of them do not maintain information about the error of the data being tracked. Also, due to the difficulty and spatial locality of the problem, existing methods can generate grossly incorrect correspondences, making outlier rejection an essential post-processing step. We propose a new generic framework that uses the Scaled Unscented Transform to augment arbitrary feature tracking algorithms, and use Gaussian Random Variables (GRV) for the representation of features' locations uncertainties. We apply and validate the framework on the well-understood Kanade–Lucas–Tomasi feature tracker, and call it Unscented KLT (UKLT). The UKLT tracks GRVs and rejects incorrect correspondences, without a global model of motion. We validate our method on real and synthetic sequences, and demonstrate how the UKLT outperforms other approaches on both outlier rejection and the accuracy of feature locations.

1. Introduction

In computer vision, several problems rely on the accurate determination of feature correspondences on images. Feature tracking deals with the particular case of time-changing images, and updates each feature's location as the image changes. It is important to select image features that have enough information to track, and that do not suffer from the aperture problem effect [1].

The Kanade–Lucas–Tomasi (KLT) is one of the most well-known and studied methods for feature tracking [2,1]. It uses a matching criterion based on a rigid translational model, which is equivalent to the minimization of the sum of squared differences of window intensities. The proper selection of feature points can greatly increase the algorithm's performance [3].

Shi and Tomasi [3] extended the initial algorithm to consider the affine model, and proposed a technique to monitor the quality of the features being tracked. If the residual of the match between image regions in the first and the current frame exceeds a threshold, the feature is rejected. Later works [4,5] extended the model to account for changes in illumination and reflection.

Unfortunately, none of these algorithms consider uncertainties during tracking, or the estimate's reliability. If we could take into account uncertainty, we would have more accurate parameter fitting procedures to noisy data. Several researchers have attempted to estimate uncertainty using ad-hoc techniques without satisfactory results [6], therefore there is the need for theoretically sound uncertainty estimation methods for feature tracking that can be applied to different types of features.

On a different take, occlusions, ambiguity, and illumination changes make even the most elaborate feature tracking algorithms fail miserably. These problems lead to false matches, that is, outliers. Although there are several methods to mitigate the effects of outliers, their computational cost is usually high [7,8]. Torr et al. [9] adopt a RANSAC [10] based approach to eliminate outliers over an image sequence. Fusiello et al. [11] propose an extension to the KLT, an automatic feature rejection rule called X84. Although there are many outlier rejection methods, there is no single algorithm that works perfectly in all cases.

In this paper we extend the study, begun in [12], of the use of Gaussian Random Variables (GRVs) combined with the Scaled Unscented Transform (SUT), a controlled sampling method, to calculate the distribution propagation over a nonlinear transformation, in our case the standard KLT algorithm. Using random variables to describe image feature locations and their uncertainties improves both the accuracy and the robustness of the tracking process. Although we do not know what the real distributions are, SUT gives us a theoretical guarantee that the first two moments of the estimates are correct. Also, using the SUT deterministic samples for outlier detection allow us to increase the robustness of the final technique without any extra computational expense.

2. Uncertainty representation

We now introduce a new generic framework that augments arbitrary feature tracking algorithms to represent and track feature

* Corresponding author.
E-mail address: leyza.dorini@gmail.com (L.B. Dorini).

1077-3142/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2010.07.009

positions as Gaussian Random Variables (GRVs). We then show how it can be applied to one of the most commonly used methods, the KLT [1].

GRVs are a good option for the representation of the probability distribution function of an image feature location. They have an easy and straightforward mathematical formulation (mean vector and covariance matrix) and a compact computational implementation. They also have an exact closed algebraic formulation for linear transformations using linear algebra operations, and use the two first moments of the distribution as their parametric representation. Haralick [13] proposed the use of covariance propagation in computer vision, but he considered only a first order linearization.

Ease of use apart, there has been some valid questioning in the literature whether covariances measured from the local grayscale information of the image can properly represent the uncertainty of the feature locations [6].

With a reliable estimation of features' GRVs, we can improve any parameter fitting procedure, such as bundle adjustment, using a weighted least squares fit that takes into account the uncertainty information. As we show in Section 4.4, the GRV's covariance matrix leads us to a Mahalanobis-distance minimization.

Chowdhurry [14] has derived an explicit expression for the error covariance in motion and structure estimates as a function of the error covariance in the feature positions in the images, but his work does not consider the effects of outliers. Steele and Jaynes [15] proposed a method to improve the uncertainty estimates of the features' locations, propagating the covariance through the Jacobian of the feature location estimator. Zhu et al. [16] have used a confidence measure based on the gray level difference, also without considering the inherent error in the algorithm being used for tracking. Additionally, Morris and Kanade [17] and Irani and Anandan [18] explore the use of uncertainty in factorization methods.

3. Tracking random variables

There are two main categories of tracking errors: *location imprecision* and *false matches* [8]. In the case of imprecisions, feature points differ by only a few pixels from their true position. False matches occur when a feature is mapped to a different location, causing gross mistakes.

The concepts of predictive filters can help us obtain more accurate estimates. Predictive filters propagate the parameters and their uncertainties through a dynamical system, and combine preliminary estimates with data obtained from the system's observations [19].

There are different types of predictive filters. The linearity constraints required by the well-known Kalman Filter [20] are not satisfied in many practical applications. The Extended Kalman Filter linearizes all models, so that we can use the Kalman Filter formulation, but it can lead to poor representations of the underlying probability distributions functions of interest.

The Unscented Kalman Filter (UKF) [21], based on the Unscented Transform, uses the true nonlinear dynamical model, and calculates the statistics of a random variable that undergoes a nonlinear transformation.

In our framework, we represent the features locations as GRVs and use the Scaled Unscented Transform to propagate them through a nonlinear transformation that, in this paper, is represented by the KLT feature tracking algorithm.

3.1. The Scaled Unscented Transform

Consider an n-dimensional random variable$\mathbf{x}$ that undergoes a nonlinear transformation $\mathbf{y} = g(\mathbf{x})$. In our case, we will always use a two-dimensional random variable for the feature location and the transformation will be the tracking algorithm.

Let $\bar{\mathbf{x}}$ and Σ_x be the mean and covariance matrix of $\mathbf{x}$, respectively. The Unscented Transform calculates the mean and covariance of $\mathbf{y}$, using $2n+1$ deterministic points, called sigma points $\mathcal{X}_i$. To ensure positive definiteness on the resulting covariance matrices, we use the Scaled Unscented Transform (SUT), an extension of the Unscented Transform that uses pre-scaled sigma points, defined by [22,21] as:

$$\begin{aligned}
\mathcal{X}_0 &= \bar{\mathbf{x}}, \\
\mathcal{X}_i &= \bar{\mathbf{x}} + (\sqrt{(n+\lambda)\Sigma_x})_i, \quad i = 1, \ldots, n, \\
\mathcal{X}_i &= \bar{\mathbf{x}} - (\sqrt{(n+\lambda)\Sigma_x})_{(i-n)}, \quad i = n+1, \ldots, 2n,
\end{aligned} \tag{1}$$

with $\lambda = \alpha^2(n+\kappa) - n$ representing the scale parameter, where α determines the spread of the sigma points around $\bar{\mathbf{x}}$ and κ is a secondary scale parameter. Finally, $(\sqrt{(n+\lambda)\Sigma_x})_i$ is the ith row of the matrix square root. This deterministic choice of the sigma points guarantees that they completely capture the true mean and covariance of the prior random variable $\mathbf{x}$. In all our sequences, we use $\alpha = 0.9$, a small spread to avoid sampling non-local effects and the occurrence of false outliers, and $\kappa = 0$, since in our case $n = 2$ and never changes. Neither the spread nor the location of the sigma points is related to the size of the window used inside the KLT algorithm.

Each sigma point has an associated weight W_i:

$$\begin{aligned}
W_0^m &= \frac{\lambda}{n+\lambda}, \quad W_0^c = \frac{\lambda}{n+\lambda} + 1 - \alpha^2 + \beta, \\
W_i^m &= W_i^c = \frac{1}{2(n+\lambda)} \quad i = 1, \ldots, 2n,
\end{aligned} \tag{2}$$

where β is used to incorporate knowledge of the higher order moments of the distribution. We use $\beta = 2$ for all sequences, the value for Gaussian priors [22]. The superscript m indicates the weight for the mean calculation, and c indicates the weight for the covariance calculation.

To estimate the mean and covariance of the resulting distribution after the nonlinear transformation $g(\cdot)$, we find the transformed sigma points $\mathcal{Y}_i = g(\mathcal{X}_i)$, and

$$\bar{\mathbf{y}} = \sum_{i=0}^{2n} W_i^m \mathcal{Y}_i \quad \text{and} \quad \Sigma_y = \sum_{i=0}^{2n} W_i^c (\mathcal{Y}_i - \bar{\mathbf{y}})(\mathcal{Y}_i - \bar{\mathbf{y}})^T. \tag{3}$$

The SUT approach results in an approximation that is accurate to at least the second order [22,21]. The Extended Kalman Filter, on the other hand, calculates the posterior mean and covariance accurately only to the first order [21]. We summarize the SUT in Algorithm 1.

Algorithm 1. SUT applied to 2D features

```
1: for each feature point do
2:     generate 5 sigma points (2 × dim + 1);
3:     propagate the sigma points using the KLT tracker;
4:     compute the estimated mean and covariance matrix (3);
5: end for
```

Fig. 1 illustrates the method.

3.2. Unscented KLT: UKLT

Existing algorithms that estimate uncertainties in feature tracking consider only the relationship between the noise models and the feature points' covariances. This emphasizes the local image

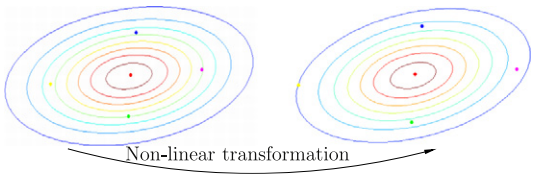


Fig. 1. The Unscented Transform.

characteristics and does not take into account the error of the tracking algorithm. In this paper, by using the KLT feature tracking algorithm as the nonlinear transformation, we consider the uncertainty of the nonlinear transformation represented by the tracking procedure itself.

Let $u(\mu_i, \Sigma_i)_k$ be the state vector of our system, where for each discrete time step k we have the mean and covariance of every feature point i. At time step k, we apply the SUT (Algorithm 1) to each feature point i in the state vector $u(\mu_i, \Sigma_i)_k$. That is, we generate the sigma points according to (1), propagate them using the KLT tracker, and finally calculate the corresponding mean and covariance (3). Here, we use the *good features to track* [3] as the criteria for the initial feature points, but further extensions of this work can consider scale and rotation invariant features [23] as starting points. The covariance of the initial features is found by the inverse of the matrix

$$C = \begin{bmatrix} \nabla^2 x & \nabla x \nabla y \\ \nabla x \nabla y & \nabla^2 y \end{bmatrix}, \tag{4}$$

where ∇x and ∇y are the gradient in the x and y directions.

For each time step k, we make an additional observation of the system, denoted by $v(\mu_i, \Sigma_i)_k$, that considers local image characteristics (gray level variation). The mean is the coordinate estimated by the KLT tracker and has the same value as χ_0, and the covariance is estimated using C^{-1} as in Eq. (4).

We look at the SUT estimate as the state prediction, and at the image local characteristics as an observation of the system. As illustrated in Fig. 2, we combine these two partial estimates using a Maximum Likelihood Estimation (MLE), an inference strategy that consists of choosing the parameters that maximize the observed measured probabilities. In other words, the MLE selects the model with the highest posterior probability given the prediction and observation [24].

Let two GRVs $\mathbf{z}_1 \sim N(\bar{\mathbf{z}}_1, \Sigma_{\mathbf{z}_1})$ and $\mathbf{z}_2 \sim N(\bar{\mathbf{z}}_2, \Sigma_{\mathbf{z}_1})$, $\mathbf{z}_1$ the prediction and $\mathbf{z}_2$ the observation. The two GRVs $\mathbf{z}_1$ and $\mathbf{z}_2$ are conditionally independent given the correct position $\mathbf{x}$ of the feature, they consist of the real position disturbed by Gaussian random noise. In Bayesian Networks, this independence property is called Markov Blanket [25].

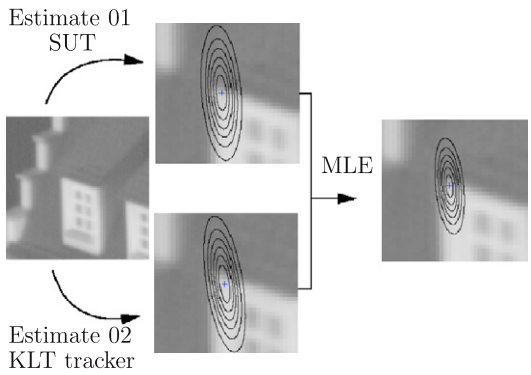


Fig. 2. Graphical interpretation of the UKLT.

The GRV that best explains two Gaussian observations is the result of a simple Maximum Likelihood Estimator MLE. The covariance estimation is the inverse of the sum of the inverses, and the mean is the weighted average (matrix-wise) using the inverse of the covariances.

$$\Sigma_{\mathbf{x}}^{-1} = \Sigma_{\mathbf{z}_1}^{-1} + \Sigma_{\mathbf{z}_2}^{-1} \tag{5}$$

$$\bar{\mathbf{x}} = \Sigma_{\mathbf{x}} \left(\Sigma_{\mathbf{z}_1}^{-1} \mathbf{z}_1 + \Sigma_{\mathbf{z}_2}^{-1} \mathbf{z}_2 \right) \tag{6}$$

The UKLT algorithm is summarized in Algorithm 2.

Algorithm 2. Unscented UKLT

```
1: select feature points and calculate their GRV
2: for each sequence frame do
3:     for each feature point do
4:         obtain the prediction estimate, z1, using the SUT
       (Algorithm 1);
5:         obtain the observation estimate, z2, based on the
       predicted state;
6:         obtain an a posteriori estimate by fusing z1 and z2
       using MLE;
7:     end for
9: end for
```

3.3. Rejecting outliers

Feature tracking is difficult and all existing matching methods eventually yield bad correspondences, making outlier rejection an essential step. Using the formalization of Section 3 and the feature quality criteria in the KLT tracking [3] applied to every sigma point, we reject outliers on-line without a global model of motion.

Sigma point tracking results lead to three simple outlier rejection criteria:

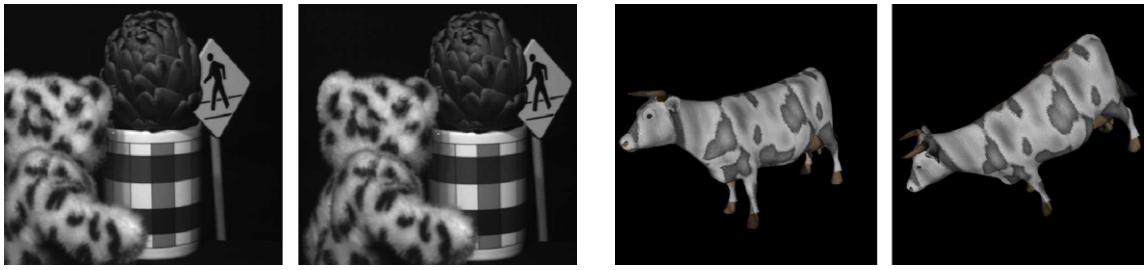
(1) Reject if KLT loses tracking of any sigma point (step 3 of Algorithm 1).
(2) Reject if the SUT estimation is a non-positive definite matrix.
(3) Reject if the sigma point tracking results have a displacement magnitude inconsistent among each other (large magnitude displacement standard deviation).

Concerning the first criteria, we discard every feature which had one or more of its five sigma points not properly tracked by the underlying KLT algorithm (step 3 of Algorithm 1, which constitutes the prediction $\mathbf{z}_1$) of UKLT.

Similarly, before dropping the sigma points, the KLT tracker might get incompatible positions, that may lead to a matrix that is not positive definite. This is a simple check of consistency of the sigma point tracking result that can be considered as a rejection criteria.

A final criteria to decide when the five sigma points do not have a consistent movement is to look at the standard deviation of the magnitude of their displacements. A large value implies inconsistent movement, which would displace the correct feature point location.

These cases usually happen when the feature is located between regions with rich and poor texture patterns, where the underlying KLT algorithm may no longer consider the results reliable. We recall that the KLT algorithm has a criteria, based on window deformations, to decide when a feature has changed enough compared to its original shape, and its result should no longer be considered reliable.



(a) Artichoke synthetic sequence. (b) Cow synthetic sequence.

Fig. 3. First and last frame of each sequence. These sequences have a ground truth.

4. Results

We evaluate the performance of our algorithm for feature tracking accuracy and for outlier rejection using four different sequences, two synthetic and two real. We generated the first synthetic sequence, Fig. 3a, through a sequence of controlled translational warpings of the well-known *Artichoke* image. For the second synthetic sequence, Fig. 3b, we rendered the animation of a rotating and translating textured 3D cow – thus, we know the exact 2D projected image coordinates of every point, for every frame. The real sequences are the *House*[1] and *Balorig*[2] static scenes observed by a moving camera (Fig. 4a and b, respectively).

We compare the accuracy of the correspondences estimated by UKLT with that estimated by the standard KLT frame by frame. Also, to analyze the robustness of the final UKLT feature set, we apply RANSAC [10] and the X84 rejection rule [11] for outlier detection in the final KLT correspondence set. The X84 rule selects a rejection threshold based on the Median Absolute Deviation (MAD) and rejects correspondence pairs that are more than k MAD from the median.

We use three different metrics for evaluation:

(1) The Root Mean Square (RMS) distance of the tracked points from their epipolar lines at the last frame (using an estimate of the fundamental matrix). If the epipolar geometry is estimated exactly, all points should lie on epipolar lines.
(2) The distance between the estimated and real position of each feature point (synthetic sequences only).
(3) The difference between the fundamental matrix [8] estimated from the ground truth and from tracked correspondences (synthetic sequences only). This metric complements the previous one, thus showing that we are working with compatible measures.

4.1. Feature tracking analysis

As we have the ground truth for the synthetic sequences, we measure the difference between the estimated and the real position of each feature point. We compare the difference frame by frame just between the standard KLT and UKLT, since in the other two algorithms (RANSAC and X84) rejection is performed only on the final set of correspondences. We carried out tests using the following initial number of points for each sequence: 75 for the Artichoke, 85 for the Cow, 237 for the House, and 55 for the Balorig.

Fig. 5a shows a plot of the mean distance error (in pixels) over all the inlier points for the Artichoke sequence and Fig. 5b for the Cow sequence. Our method gets better estimates in both sequences. The smaller error is partially due to the UKLT rejection of incorrectly tracked features, which are kept by the KLT algorithm (Section 3).

Fig. 6 shows the mean distance of the feature points from the corresponding epipolar lines in each sequence frame. We compute the fundamental matrices using the UKLT and KLT correspondences (without using robust methods). Fig. 6a shows the results for the Cow sequence, Fig. 6b for the Artichoke sequence, Fig. 6c for the House sequence, and Fig. 6d for the Balorig sequence. Based on these experimental results, we conclude that our method tracks features more accurately and robustly than the KLT.

As mentioned before, better estimates are partially due to the UKLT rejection of incorrectly tracked features, which are kept by the KLT algorithm (Section 3) Fig. 7 illustrates the number of feature points kept for each sequence frame. Observe that UKLT discards more features than KLT. In the experimental tests, the KLT algorithm keeps from 80% to 95% of the features, while UKLT keeps from 52% to 82%.

We also compare the accuracy of both methods only for the features considered as inliers by UKLT. This analysis shows that our algorithm performs better not only because of outlier rejection, but also due to the improvement of the feature positions' estimates. Fig. 8 shows the results for all the test sequences being considered.

We also carried out computational tests on an image sequence with 200 frames. The first and last frames of the *Medusa* sequence are illustrated in Fig. 9.

Fig. 10a shows the mean distance of the feature points from the corresponding epipolar lines in each sequence frame. As before, we compute the fundamental matrices using the UKLT and KLT tracked points. Fig. 10b shows the number of tracked features in each sequence frame. When analyzing the discarded features in terms of the outlier rejection mechanisms presented in Section 3.3, we observe that 81% of the features are lost because one of the sigma points is not correctly tracked, 13% due to the large displacement of the features and 6% because the SUT estimation is a non-positive definite matrix.

Since the proposed approach discards features more rapidly than KLT, we introduce a simple mechanism to include new features as a possible solution to this problem. When 50% of the initial features are lost, a selection occurs, which happened five times for this experiment (observe the peaks in Fig. 11b). Fig. 11a shows the mean distance of the feature points from the corresponding epipolar lines in each sequence frame. Here, we compute the fundamental matrix between the current and the last frame were a feature selection occurs. In such a way, the distance to the epipolar lines is zero in each new selection and, thus, it does not make sense to compare this measures against KLT.

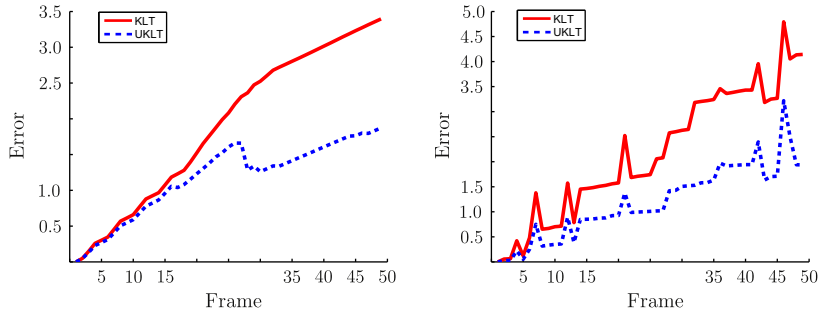
4.2. Outlier rejection analysis

The automatic on-line rejection of mismatched feature points done by UKLT discards more outliers than the other robust KLT-

[1] http://vision.ucla.edu/MASKS/.
[2] http://sampl.ece.ohio-state.edu/database.htm.

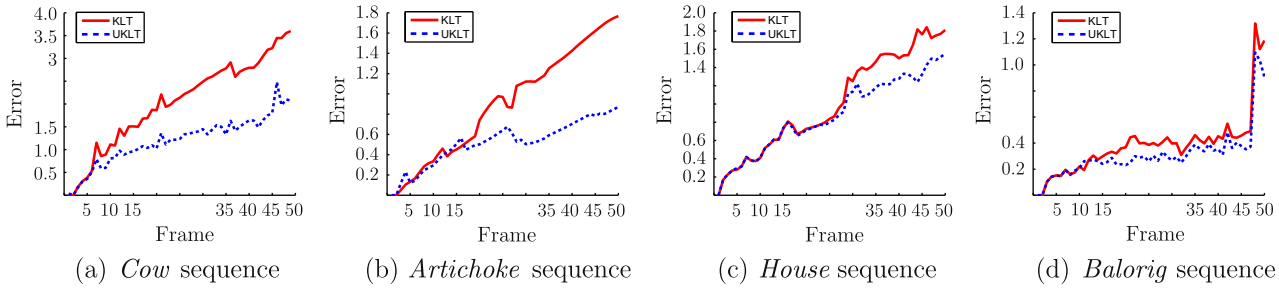
(a) House real sequence. (b) Balorig real sequence.

Fig. 4. First and last frame of each sequence.



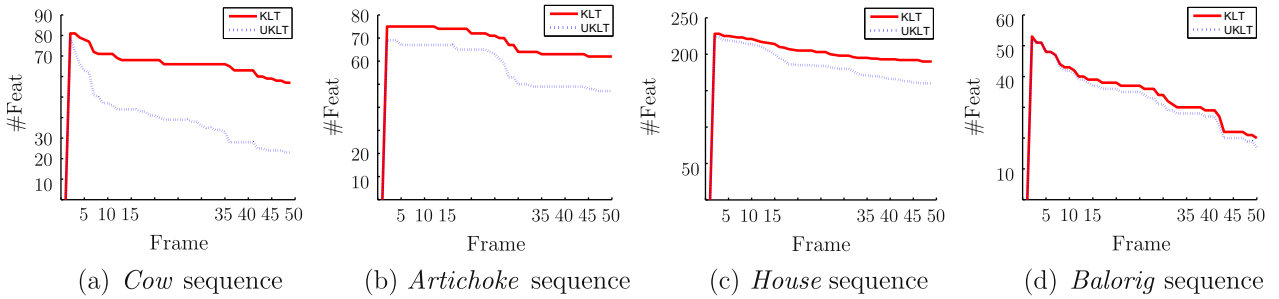
(a) *Artichoke* sequence (b) *Cow* sequence

Fig. 5. Euclidean distance of the estimated feature positions from the real ones (synthetic sequences) (lower is better).



(a) *Cow* sequence (b) *Artichoke* sequence (c) *House* sequence (d) *Balorig* sequence

Fig. 6. RMS distance between the feature points and their corresponding epipolar lines (lower is better).



(a) *Cow* sequence (b) *Artichoke* sequence (c) *House* sequence (d) *Balorig* sequence

Fig. 7. Number of tracked feature points in each sequence frame.

based algorithms. In Fig. 12, we show a part of the tracking results in the last frame of the *Cow* sequence, for the algorithms UKLT and KLT (with the RANSAC application on the final correspondence set). The symbols + indicate the estimated positions and □ the real ones. The KLT errs significantly under rotation. The UKLT is more robust in this case, detecting the false matches made by KLT.

UKLT rejects more features than KLT, and the amount of rejections vary from sequence to sequence depending on the image conditions (as mentioned earlier, UKLT tracks five feature points, thus considering a small neighborhood around the central feature. This procedure may be influenced by image characteristics like texture). As mentioned before, in a general way, the KLT algorithm keeps from 80% to 95% of the features, while UKLT keeps from 52% to 82%.

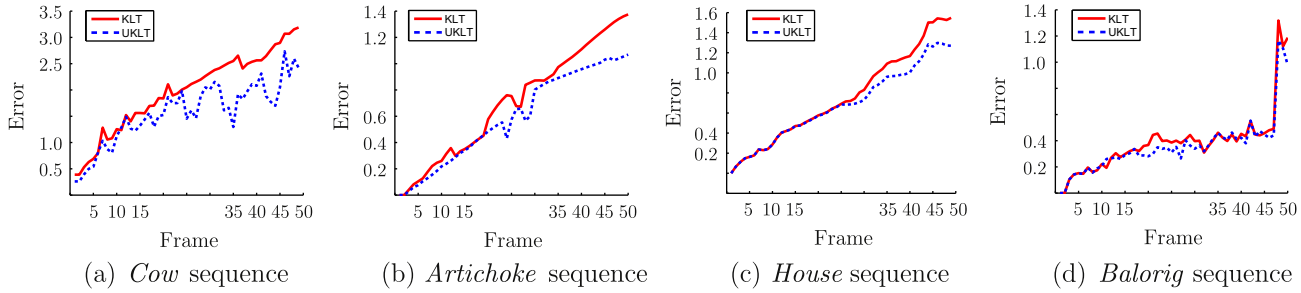


Fig. 8. RMS distance between the feature points and their corresponding epipolar lines when considering the same features for KLT and UKLT on real sequences (lower is better).

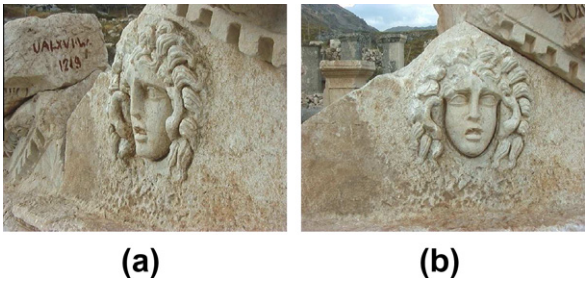


Fig. 9. First and last frame of the *Medusa* sequence.

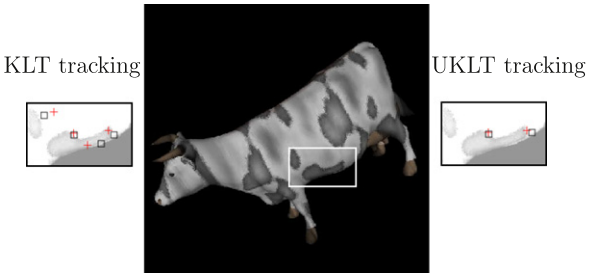


Fig. 12. Tracking results comparison.

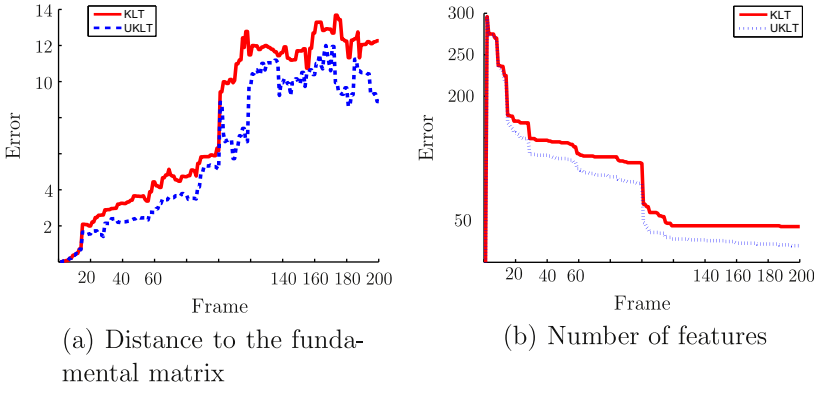


Fig. 10. Analysis of a larger image sequence.

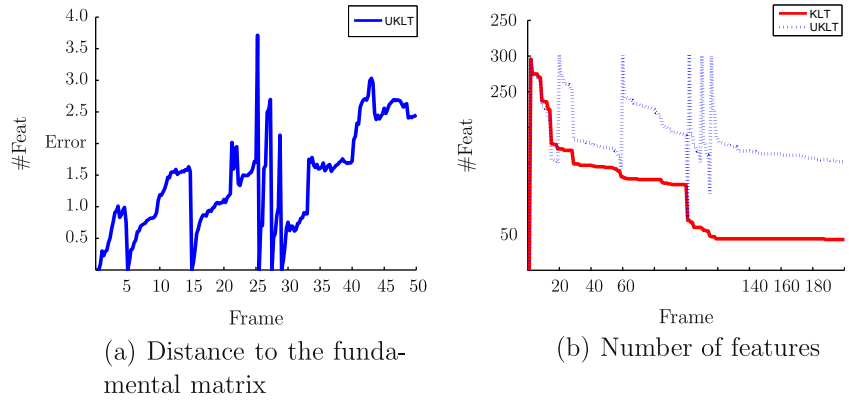


Fig. 11. RMS distance between the feature points and their corresponding epipolar lines when considering the same features for KLT and UKLT on real sequences (lower is better).

The decreasing on the number of features as the sequence evolves depends on different aspects. Besides the fact that the underlying algorithm lose track of some features, one must also consider that some image regions disappear. We emphasize that

Table 1
Error metrics. dEpL: RMS distance to the epipolar line; dGT: distance between the real and the estimated position; and Zhang: fundamental matrix comparison using the Zhang's method. Measurements for the last frame (lower is better). Results between first and last frames.

Sequence	UKLT			KLT		
	dEpL	dGT	Zhang	dEpL	dGT	Zhang
Artichoke	1.31	1.87	57.10	2.26	3.40	178.66
Cow	2.37	4.18	45.17	5.18	7.81	66.52
House	1.51	–	–	1.77	–	–
Balorig	0.61	–	–	0.70	–	–
Artichoke	2.21	3.28	93.03	1.92	2.81	149.53
Cow	4.04	6.54	49.02	4.49	7.24	59.28
House	1.80	–	–	1.75	–	–
Balorig	0.69	–	–	0.68	–	–

this would also constitute a problem for the other approaches (KLT, for example).

4.3. Comparative results

Table 1 shows the results for the error metrics that compare the accuracy of the tracking and of the fundamental matrix estimation. We use the correspondence set estimated by each of the methods (KLT, KLT + X84, KLT + RANSAC and UKLT) for the last frame of each sequence (as suggested in [11] and due to the characteristic of the RANSAC algorithm, which considers an estimation model).

As discussed in the following, UKLT performs better than the other approaches for the three metrics used in all sequences. Note that the improvement in synthetic sequences is greater than in the real ones. In this sense, it is necessary to take into account that the synthetic sequences have controlled parameters regarding illumination changes and noise, for example. Besides the nature of the sequences, the improvement also depends on different factors that may not be considered alone. The nature of motion, for example, is relevant, but must be combined with image characteristics like texture and noise. The greater the complexity of these parameters, the greater the probability of a feature ro be lost.

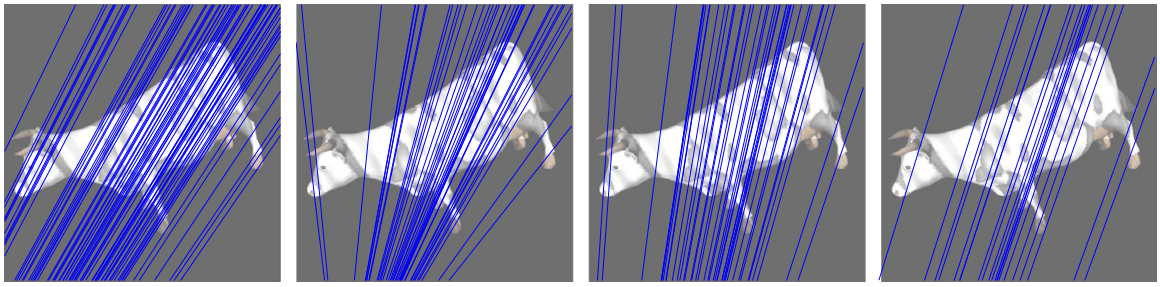
In Fig. 13, we draw the epipolar lines from the fundamental matrix estimated from the feature points extracted from each method.

In usual filtering applications, it is common to use the Square-Root Unscented Kalman Filter (SQUKF) variation [26] for robust estimation. The SQUKF has similar accuracy than the Unscented Kalman Filter, but with an increase gain in performance, as the dimension of the state vector increases. In our application the state vector is always bidimensional, so the additional complexity to propagate and handle the square-root of the covariance matrix, and the extra Cholesky decomposition procedure is unnecessary for performance goals only. Another usual benefit of the SQUKF is its numerical stability, which keeps the positive definitiveness properties of the matrices longer in the recursive process. In our case, we have shown how the loss of positive symmetry is another criteria for outlier rejection, so adding more robustness in the estimation of the covariance matrix would be detrimental.

4.4. Application to bundle adjustment

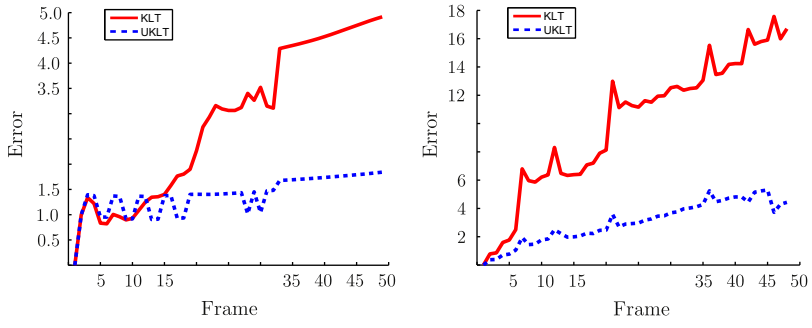
We use features to estimate camera motion and 3D scene depth using a structure from motion algorithm. To estimate the unknown 3D features (the structure of the scene) and the camera parameters, we minimize the total prediction error.

Bundle adjustment is the model refinement part of this process, refining the visual reconstruction to obtain both 3D structure and



(a) Ground truth (b) KLT + RANSAC (c) KLT + X84 (d) UKLT

Fig. 13. Epipolar lines (from the fundamental matrix estimated by each method).



(a) *Artichoke* sequence (b) *Cow* sequence

Fig. 14. Reprojection error magnitude.

Table 2
Running times.

Sequence	UKLT	KLT
Artichoke sequence	49.66	19.89
Balorig sequence	48.19	15.47
Cow sequence	89.87	22.11
House sequence	159.02	41.84

viewing optimal parameters estimates. It involves the minimization of a cost function [24] related to the model fitting error.

Nonlinear least squares (NLS) is a classic formulation of bundle adjustment computations. Suppose that we have vectors of observations v_i predicted by a model $z_i = z_i(\mathbf{x})$, where $\mathbf{x}$ is a vector of model parameters. NLS takes as estimates the parameters values that minimize the weighted Sum of Squared Error cost function [24]:

$$f(x) \equiv \frac{1}{2} \sum_i \Delta z_i(\mathbf{x})^T W_i \Delta z_i(\mathbf{x}), \tag{7}$$

where $\Delta z_i(\mathbf{x})$ is the feature prediction error and W_i is an arbitrary symmetric positive definite weight matrix. In this paper, we set W to the covariance matrix obtained by our method, giving us an uncertainty measure of the structure and motion parameters.

Here, we use the structure from motion and bundle adjustment procedures described in [27]. To evaluate the quality on the 3D reconstruction, we reproject the estimated feature points and measure the error magnitude to the ground truth, in the case of the synthetic sequences. Fig. 14a and b illustrates the results for the synthetic sequences. Note that for the *Cow* sequence, instabilities are evident in the frames that undergo abrupt changes. As before, these results reflect how KLT errs significantly under rotation, leading to gross mistakes.

4.5. Implementation considerations

We have implemented our algorithm using Matlab and C. The running times on a Intel Core 2 Duo 2 GHz, with 2 Gb of memory are in Table 2.

Note that the cost of the UKLT algorithm its higher, since it involves the tracking of five times more points. However, as the outlier rejection is done on-line, we do not need additional steps (and time) for this.

Even though the pyramid decomposition (part of the KLT algorithm, as discussed in [27]) is done only once, there is an extra overhead for every feature to find the five sigma points, and to later merge them back into a GRV. All things considered, UKLT takes approximately five times longer than the standard KLT at the initial frames. As the sequence evolves, the execution time of the proposed approach is amortized (due to the outliers rejected) and the UKLT approach becomes about 3.5 slower than KLT.

On the other hand, a good model-free outlier rejection method is included for free, and there is the extra benefit of the distribution estimation that can be used as a building block by other vision applications.

5. Conclusions

In this paper, we describe a framework to incorporate GRVs into point tracking algorithms using the Scaled Unscented Transform and a Maximum Likelihood Estimator. We showed how to augment the well-understood KLT tracking algorithm.

Using random variables to represent the features's locations improves both the accuracy and the robustness of the tracking process. Although we cannot be sure if the real underlying distributions are Gaussians or not, the SUT gives us a guarantee that the first two moments of the estimates are correct.

Our comparison and validations confirm that our method is also better at discarding wrong features. The on-line outlier rejection avoids post-processing steps, increasing the process robustness. The limitations of the proposed approach are the same of the underlying tracking algorithm. Future work includes the use of other trackers and the extension of the results to 3D and range-scan based algorithms.

References

[1] C. Tomasi, T. Kanade, Detection and Tracking of Point Features, Tech. Rep. CMU-CS-91-132, Carnegie Mellon University, April 1991.
[2] B. Lucas, T. Kanade, An iterative image registration technique with an application to stereo vision, in: IJCAI81, 1981, pp. 674–679.
[3] J. Shi, C. Tomasi, Good features to track, Proceedings of IEEE Computer Vision and Pattern Recognition (1994) 593–600.
[4] G. Hager, P. Belhumeur, Efficient region tracking with parametric models of geometry and illumination, PAMI 20 (1998) 1025–1039.
[5] H. Jin, P. Favaro, S. Soatto, Real-time feature tracking and outlier rejection with changes in illumination, in: Proceedings of the IEEE International Conference on Computer Vision, 2001, pp. 684–689.
[6] Y. Kanazawa, K. Kanatani, Do we really have to consider covariance matrices for image features? in: Proceedings of the IEEE International Conference on Computer Vision, 2001, pp. 301–306.
[7] J. Salvi, X. Armangue, J. Pages, A survey addressing the fundamental matrix estimation problem, in: International Conference on Image Processing, 2001, pp. 209–212.
[8] Z. Zhang, Determining the epipolar geometry and its uncertainty: a review, IJCV 27 (2) (1998) 161–198.
[9] P. Torr, A. Zisserman, S. Maybank, Robust detection of degeneracy, in: Proceedings of the IEEE International Conference on Computer Vision, 1995, pp. 1037–1044.
[10] M. Fischler, R. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Communications of the ACM 24 (6) (1981) 381–395.
[11] A. Fusiello, E. Trucco, T. Tommasini, V. Roberto, Improving feature tracking with robust statistics, Pattern Analysis and Applications 2 (1999) 312–320.
[12] S. Dorini, L.B. Goldenstein, Unscented KLT: nonlinear feature and uncertainty tracking, in: Brazilian Symposium on Computer Graphics and Image Processing, 2006, pp. 187–193.
[13] R. Haralick, Propagating covariance in computer vision, in: 12th IAPR International Conference on Computer Vision & Image Processing, 1994, pp. 493–498.
[14] A.K.R. Chowdhury, Statistical analysis of 3d modeling from monocular video streams, Ph.D. thesis, University of Maryland, 2002.
[15] P.M. Steele, C. Jaynes, Feature uncertainty arising from covariant image noise, Proceedings of IEEE Computer Vision and Pattern Recognition 1 (2005) 1063–1070.
[16] J. Zhu, S. Schwartz, B. Liu, Object tracking: feature selection and confidence propagation, CRV (2004) 18–21.
[17] D. Morris, T. Kanade, A unified factorization algorithm for points, line segments and planes with uncertainty models, in: Proceedings of the IEEE International Conference on Computer Vision, 1998.
[18] M. Irani, P. Anandan, Factorization with uncertainty, in: Proceedings of the European Conference on Computer Vision, 2000.
[19] S. Goldenstein, A gentle introduction to predictive filters, Revista de Informática Teórica e Aplicada (RITA) XI (1) (2004) 61–89.
[20] R.G. Brown, P. Hwang, Introduction to Random Signals and Applied Kalman Filtering, John Wiley and Sons, 1997.
[21] S. Julier, J. Uhlmann, A new extension of the Kalman filter to nonlinear systems, SPIE (1997).
[22] E. Wan, R. van der Merwe, Kalman Filtering and Neural Networks, Wiley Publishing, 2001.
[23] D. Lowe, Distinctive image features from scale-invariant keypoints, IJCV 60 (2) (2004) 91–110.
[24] P. Torr, A. Zisserman, S. Maybank, Bundle adjustment – a modern synthesis, in: Proceedings of the International Workshop on Vision Algorithms: Theory and Practice, 1999, pp. 298–372.
[25] R. Neapolitan, Learning Bayesian Networks, Prentice Hall, 2003.
[26] R. van der Merwe, E. Wan, The square-root unscented Kalman filter for state and parameter estimation, in: IEEE International Conference on Acoustics, Speech, and Signal Processing, 2001, pp. 3461–3464.
[27] Y. Ma, S. Soatto, J. Kosecka, S. Sastry, An Invitation to 3D Vision-From Images to Geometric Models, Springer, 2004.