Uncertainty estimation for KLT tracking

Sameer Sheorey[1], Shalini Keshavamurthy[1], Huili Yu[1], Hieu Nguyen[1] and Clark N. Taylor[2]

[1] UtopiaCompression Corporation, Los Angeles, California 90064
[2] Air Force Research Laboratory, Ohio, USA

Abstract. The Kanade-Lucas-Tomasi tracker (KLT) is commonly used for tracking feature points due to its excellent speed and reasonable accuracy. It is a standard algorithm in applications such as video stabilization, image mosaicing, egomotion estimation, structure from motion and Simultaneous Localization and Mapping (SLAM). However, our understanding of errors in the output of KLT tracking is incomplete. In this paper, we perform a theoretical error analysis of KLT tracking. We first focus our analysis on the standard KLT tracker and then extend it to the pyramidal KLT tracker and multiple frame tracking. We show that a simple local covariance estimate is insufficient for error analysis and a Gaussian Mixture Model is required to model the multiple local minima in KLT tracking. We perform Monte Carlo simulations to verify the accuracy of the uncertainty estimates.

1 Introduction

The Kanade-Lucas-Tomasi feature tracker (KLT), developed in [1], [2] and [3], is the most commonly used approach to feature point tracking in image sequences. KLT searches for the location of a given feature point in the next few images by matching the local image patch intensity. Hierarchical search using image pyramids improves the tracking range (Bouguet [4]). The KLT tracker's excellent speed and reasonable accuracy make it popular in many applications such as video stabilization, egomotion estimation, image mosaicing, 3D reconstruction, visual odometry, and Simultaneously Localization and Mapping (SLAM). There exist many other extensions of the standard KLT algorithm that aim to increase accuracy and efficiency of computations. Baker and Matthews [5] give an overview of the extensions.

While the standard KLT algorithm and its extensions are successful in performing feature tracking, they simply output the estimated displacement without any indication of its accuracy. KLT displacement estimates are noisy due to image intensity noise and corresponding errors in the original feature detection. Complex local image structure is also a major source of error. An error model of the KLT tracker will be useful in downstream applications that aggregate tracking results over many points and frames to produce their output. For example, bundle adjustment for structure and motion estimation naturally uses feature point location covariances. More accurate modelling of the likelihood function

using these covariances results in more accurate structure and motion estimation (Triggs [6]).

The objective of this paper is to characterize the output uncertainty of the KLT tracker as a function of the uncertainty in its input feature location. (e.g. uncertainty in corner detection.) We first analyse the standard single level KLT tracker in an error propagation framework using least squares estimation theory. We build upon the local covariance representation in Kanazawa and Kanatani [7] and Nickels and Hutchinson [8] and generalize it. Error propagation analysis allows us to extend uncertainty estimation to pyramidal KLT tracking as well as multi-frame KLT tracking. Due to the existence of local minima in KLT tracking, a local single Gaussian covariance representation is insufficient to model the error. To address this issue, our error analysis approach represents the error using a Gaussian Mixture Model (GMM). The GMM model quantifies the probability that KLT tracking will get stuck in different local minima. This GMM error model is the main novel contribution of this work. Further, we approximate the GMM by a single covariance matrix that accounts for multiple local minima for use in downstream applications such as bundle adjustment.

The rest of the paper is organized as follows. We start with a review of related work in uncertainty estimation in computer vision in Section 2. Section 3 describes the KLT tracker briefly and Section 4 presents an uncertainty analysis of the single level and pyramidal versions. We show experiments to validate our results in Section 5. Section 6 concludes the paper.

2 Related Work

The structure from motion (SfM) and ego-motion estimation pipelines consists of feature detection, feature tracking (for image sequences) or matching (for unordered image collections), structure and motion initialization and finally bundle adjustment. Since the final bundle adjustment stage is improved by error estimates of its input, previous work has targeted error analysis on the earlier stages. Some of these include location uncertainty estimation of Harris corner detection (Orgunner and Gustafsson [9]) and SIFT point features (Zeisl *et al* [10]). Approximate error estimation for feature detection also include Brooks *et al* [11] and Kanazawa and Kanatani [7]. Nickels and Hutchinson [8] have also used feature tracking error covariance for simple tracking and physical measurements. These approaches only evaluate local covariance matrices and do not consider the common scenarios of pyramidal and multi-frame tracking. We consider full error propagation in the SfM pipeline, including input error from the feature detector and output error to the downstream stages. Our theoretical analysis also shows that local covariance matrices are insufficient uncertainty estimates due to multiple local minima. A full Gaussian Mixture Model (GMM) error representation is required.

Recently, Pfeiffer, Gehrig and Schneider [12] have shown that using confidence information can improve stereo computation. We hope that our work will lead

to similar improvements in ego-motion estimation / SLAM and Structure from Motion.

3 The KLT Feature Tracker

We will first give an overview of the KLT tracking algorithm. We will start its error analysis with the basic algorithm in the noise free case and progressively do a more realistic error analysis including image noise and finally conclude with pyramidal and multi-frame tracking.

The KLT tracker starts with a set of sparse image features in the current frame I and attempts to find their locations in the next frame J by matching an image patch around a feature to the corresponding image patch in the next frame. The *brightness constancy* assumption implies that the patch intensities will not change substantially in the next image. Using a patch allows distinguishing between neighboring points of similar intensity. A window function $w(\mathbf{x})$, usually a Gaussian function, is used to emphasize the pixels near the feature point more than those far away. This accounts for the fact that points closer to the feature point are more likely to have similar motion than those that are farther away. The window is scaled so that $\sum_W w(\mathbf{x}) = |W|$ (the number of pixels in W). Some implementations such as Bouguet [4] use a simpler square window with uniform weights. Matching proceeds by calculating the error function

$$\epsilon(\mathbf{d}) = \sum_{W(\mathbf{x}_0)} [J(\mathbf{x}+\mathbf{d}) - I(\mathbf{x})]^2 w\left(\frac{\mathbf{x}-\mathbf{x}_0}{\sigma_w}\right) \tag{1}$$

over the support $W(\mathbf{x}_0)$ of the window function centered around the feature point $\mathbf{x}_0$. Minimizing this weighted (or generalized) nonlinear least squares (weighted NLS) expression yields the estimate for the displacement $\mathbf{d}$ for the feature point located at $\mathbf{x}_0$ in the image I.

This error function is minimized with a Newton-Raphson style algorithm that iteratively linearizes the next image intensity function J using Taylor series about the current feature location estimate. Shi and Tomasi [3] proposed allowing affine deformations of the image patch to check feature tracks extending for more than 5-10 frames, while the simpler displacement model suffices for tracking between consecutive frames. Some implementations such as in the OpenCV library prefer to use the full affine model even when tracking between consecutive frames, though that can cause some slowdown in tracking. We do not analyse the affine extension here.

Standard KLT is unable to track features successfully for large inter frame motion. Bouguet [4] solves this problem by using a pyramidal implementation. A Gaussian image pyramid is created and for each feature, the search starts from the coarsest level. The minimum found at a level is propagated to the next finer level as an initialization. The final feature location is found at the finest (base) level. We will extend our analysis to pyramidal KLT as well.

3.1 Unmodeled Sources of Error

KLT search accounts for sources of error such as image noise and viewpoint change (using affine transformation). Since KLT is a local search, it can get stuck in local minima and miss the actual (global) minimum. Pyramidal KLT greatly helps to reduce this error. Some extensions even allow for lighting and contrast changes. These extensions are more complex to analyse and here we will limit ourselves to image noise and displacements. There are also other sources of error that are not modelled at all. These include specular highlights, defocus and motion blur. Further, if the feature point is at a depth discontinuity, a viewpoint change can completely alter the appearance of its local image patch causing KLT tracking to be erroneous. Similarly, if the point being tracked is on a moving object, the results will cause errors in downstream applications even if KLT tracking succeeds. The last two issues highlight the need for robust estimation in downstream applications. All these unmodeled errors will limit the accuracy of our error estimates.

4 Uncertainty Estimation

KLT tracking is a weighted least squares estimation (WLS). Assume that both images have i.i.d. Gaussian noise with zero mean and σ^2 variance in pixel intensities. We assume that the image J is a shifted version of I, with independent variably distributed (*i.v.d*) Gaussian noise added.

$$J(\mathbf{x}+\mathbf{d}) = I(\mathbf{x})+e(\mathbf{x}) \;\text{ with }\; e(\mathbf{x}) \sim \mathcal{N}(0, 2\sigma^2 \text{diag}(w(\mathbf{x}-\mathbf{x}_o)^{-1})) \;\text{ for }\; \mathbf{x} \in W(\mathbf{x}_0) \tag{2}$$

For a Gaussian window, the assumed Gaussian noise covariance increases rapidly as we move away from the center of the patch and indicates reduced confidence in the used motion model. The covariance is $2\sigma^2$ at $\mathbf{x} = \mathbf{x}_0$, since it corresponds to the difference of the i.i.d. Gaussian noise from the two images. Farther away pixels are not very likely to have the same displacement as our feature point and their displacement is assumed to be almost uniformly distributed (infinite variance). The simpler box model assumes *i.i.d* Gaussian noise inside the window and no constraints on the displacements outside it.

4.1 Single Level KLT

We now analyze the error of the KLT tracking. Due to the noise in the first image, the initial location of the feature point $\mathbf{x}_{t-1}$ is uncertain. Figure 1 shows the resulting error function $\epsilon(\mathbf{d})$, where the red ellipse indicates the variance of the initial location of the feature point.

We first analyze the error for the case where there is no image noise. The KLT tracker essentially performs a local minimization on the error function starting from the feature point location in the current frame. Thus, it will converge to the local minimum $\mathbf{x}_t = \hat{\mathbf{b}}$ corresponding to the basin of attraction of the starting point. In Figure 1, the starting point is in B_1 and hence it will converge to $\hat{\mathbf{b}} = \mathbf{b}_1$.

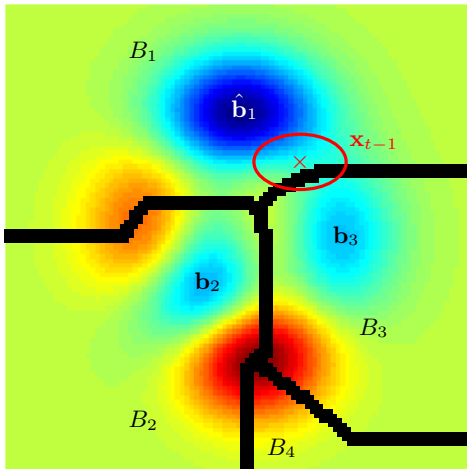

Fig. 1. KLT error surface $\epsilon(\mathbf{d})$. Uncertain detection in previous frame is represented by the red ellipse. Basins of attraction (B_i) on error surface are delineated by the thick black contours and their minima are shown in blue ($\mathbf{b}_i$). The basin of the initial feature determines the convergence point.

We can now state the probability of KLT converging to different basin minima as the probability of the starting point being in that basin. This also allows us to calculate the mean and variance. With

$$p_i := \Pr(\hat{\mathbf{b}} = \mathbf{b}_i) = \Pr(\mathbf{x}_{t-1} \in B_i) \tag{3}$$

$$\text{mean } \bar{\mathbf{x}}_t = \sum_i p_i \mathbf{b}_i \tag{4}$$

$$\text{covariance } \Sigma = \sum_i p_i (\bar{\mathbf{x}}_t - \mathbf{b}_i)(\bar{\mathbf{x}}_t - \mathbf{b}_i)^T \tag{5}$$

Assuming that the global minimum ($\hat{\mathbf{b}}$) corresponds to the actual feature point location, we then compute the bias

$$\text{bias } = \bar{\mathbf{x}}_t - \hat{\mathbf{b}} \tag{6}$$

We now analyze the uncertainty for the case with image noise. Let us assume that i.i.d. Gaussian noise with zero mean and variance σ^2 is added to each pixel intensity of the images I and J. This is equivalent to adding Gaussian noise with variance $2\sigma^2$ to the (shifted) difference image. We will analyze this problem with multiple local minima by partitioning the parameter space into the basins of attraction of the local minima. We then have a different NLS problem for each sub-domain.

$$P_i : \quad \min_{\mathbf{B}_i} \epsilon(\mathbf{d}) \tag{7}$$

The advantage is that each problem is well posed with a unique global minimum and can be analyzed by standard NLS techniques. Finally, the starting point $\mathbf{x}_{t-1}$ is randomly selected, with probability p_i given by Equation 3, with which these NLS problems will be solved.

We will start the analysis with results on NLS for the ith problem (Seber and Wald [13, Sections 2.1.4, 2.8.8]) The displacement parameter $\mathbf{d}$ is now constrained to lie in $\mathbf{B}_i$. Let $\mathbf{H}_i$ be the Hessian matrix of the error function evaluated at the (now global) minimum $\mathbf{b}_i$. If we decompose the error function $\epsilon(\mathbf{d})$ into a sum of its terms $\epsilon_j(\mathbf{d}, \mathbf{x})$ as

$$\epsilon(\mathbf{d}) = \sum_j w\left(\frac{\mathbf{x}_j - \mathbf{x}_0}{\sigma_w}\right)\epsilon_j^2 \text{ with } \epsilon_j := J(\mathbf{x}_j + \mathbf{d}) - I(\mathbf{x}_j) \tag{8}$$

We have

$$\mathbf{H}_i = \mathbf{F}(\mathbf{b}_i)^T \operatorname{diag}(w(\mathbf{x}))\mathbf{F}(\mathbf{b}_i), \tag{9}$$

$$\text{where } \mathbf{F}(\mathbf{d}) := [F_j(\mathbf{d})] \text{ with } F_j(\mathbf{d}) := \frac{\partial \epsilon_j(\mathbf{d})}{\partial \mathbf{d}} \text{ is the image gradient.} \tag{10}$$

Here $\operatorname{diag}(w(\mathbf{x}))$ is a diagonal matrix with $w(\mathbf{x})$ as the diagonal. If there are n pixels (n regressors) in the image patch to be compared, $\mathbf{F}(\mathbf{b})$ is an $n \times 2$ gradient matrix. $\mathbf{F}_j$ is a row of $\mathbf{F}$ and contains the gradient of the current image with respect to the shift for each pixel in the patch. Weighted least squares estimation theory tells us that the estimate $\hat{\mathbf{b}}_i$ is asymptotically normally distributed according to

$$\Pr(\hat{\mathbf{b}}_i) \sim \mathcal{N}(\mathbf{b}_i, 2\sigma^2\mathbf{H}_i^{-1}). \tag{11}$$

Now let us consider the original problem with the full domain. The starting point selects the ith problem for solution with probability p_i, which results in an estimate that is asymptotically normally distributed according to Equation 11. Consequently, the final estimate $\hat{b} = \mathbf{x}_t$ is distributed according to a Gaussian Mixture Model and we have

$$\Pr(\mathbf{x}_t) = \sum_i p_i g(\mathbf{b}_i, 2\sigma^2\mathbf{H}_i^{-1}) \tag{12}$$

$$\text{covariance } \Sigma = \sum_i p_i(\mathbf{b}_i\mathbf{b}_i^T + 2\sigma^2\mathbf{H}_i^{-1}) - \bar{\mathbf{x}}_t\bar{\mathbf{x}}_t^T \tag{13}$$

Mean and bias are given by Equation 4 and Equation 6. The function $g(\mu, \Sigma)$ is the Gaussian probability density function with mean μ and covariance matrix Σ. We now have the basic theoretical framework for error analysis of the KLT tracker.

Estimating image noise: We can use weighted least squares theory to estimate the Gaussian noise variance σ present in the image from the error residue at the global minimum as

$$2\hat{\sigma}^2 = \epsilon(\hat{\mathbf{b}}), \tag{14}$$

given that the weights are scaled such that $\sum_W w(\mathbf{x}) = 1$. We will use this value of σ in Equation 12 and Equation 13. Unfortunately, this noise estimate is very sensitive to model fidelity — if the image transformation cannot be accurately represented as a shift in the window around the feature point, this is likely to be a gross overestimate. Hence assuming that the entire frame has the same noise, we calculate this as the minimum over all points tracked in a frame.

4.2 Pyramidal KLT and Multi-frame Tracking

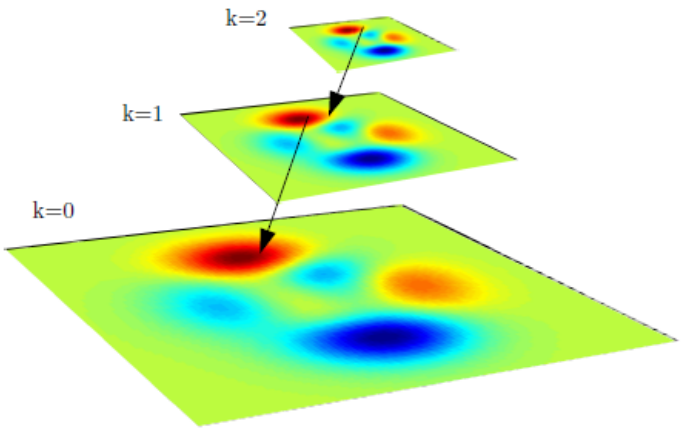

Fig. 2. Error Propagation in Pyramidal KLT Tracking.

The Pyramidal KLT tracker [4] constructs a Gaussian pyramid of each image in the sequence by low pass filtering and downsampling. The next coarser level of the pyramid is constructed by filtering with a Gaussian of standard deviation 1 (usually approximated by the low pass filter [1 4 6 4 1]/16) and downsampling by 2. Features are first tracked at the coarsest level. The tracks are then propagated to the next finer level and tracking is repeated using the coarse level initialization. The finest level (original image) tracking results are used as the final tracking results. The tracking process is shown in Figure 2. Pyramidal KLT offers improved tracking of features with large displacements. Since smoothing and downsampling reduce local minima, the coarse level tracking is more successful. Smoothing also reduces image noise, and consequently the tracking error. The lower levels further refine the displacement. We can propagate errors from the coarsest level L to the finest level 0 (original images) of the pyramid. Since each new level halves the image noise, the noise standard deviation at a level k is $\sigma^{(k)} = 2^{-k}\sigma$.

We will use the residue at the finest scale to estimate σ, i.e. $2\hat{\sigma}^2 = \epsilon^{(0)}(\hat{\mathbf{b}})$, since the translation model is most faithful at this scale. The error distribution

at level k is then

$$\Pr(\mathbf{x}_t^{(k)}) = \sum_i p_i^{(k)} g(\mathbf{b}_i^{(k)}, 2^{1-2k}\sigma^2 {\mathbf{H}_i^{(k)}}^{-1}), \text{ with} \tag{15}$$

$$p_i^{(k)} := \Pr(\hat{\mathbf{b}}^{(k+1)} \in B_i^{(k)}) \tag{16}$$

$p_i^{(k)}$ is the probability that the next coarser level $(k+1)$ KLT converges to a point $\hat{\mathbf{b}}^{(k+1)}$ that lies inside the level k error function basin $B_i^{(k)}$. The matrix $H_i^{(k)}$ is the Hessian matrix at level k for basin i. Equation 15 can be iterated to propagate the error distribution from the coarsest level k to the finest level 0. The final error is distributed according to a Gaussian mixture model and we can compute the net bias and covariance using Equation 6 and Equation 13.

A very similar error propagation analysis can be done for KLT tracking across multiple frames.

5 Evaluation

We conduct simulations to evaluate the performance of the uncertainty estimation method for the KLT tracker. To evaluate consistency, we conduct Monte Carlo simulations on the KLT and use the Average Normalized Estimation Error Squared (ANEES) as the evaluation metric. The ANEES is a standard metric to evaluate the consistency of an estimator [14], and it is defined by

$$\text{ANEES} = \frac{1}{nN} \sum_{i=1}^{N} \epsilon_i, \tag{17}$$

where n is dimension of the parameter vector, N is the total number of Monte Carlo runs, and ϵ_i is the NEES in the i^{th} Monte Carlo run, which is given by

$$\epsilon_i = (\bar{\theta}_i - \hat{\theta}_i)^\top \mathbf{P}^{-1} (\bar{\theta}_i - \hat{\theta}_i), \tag{18}$$

where $\bar{\theta}_i$ is the true parameter vector, $\hat{\theta}_i$ is the estimated parameter vector returned by the i^{th} Monte Carlo run, and $\mathbf{P}$ is the estimator-provided error covariance, which is computed by Equation 13. The ANEES value of a consistent estimator should be close to 1. Since ANEES is an average ratio, it is best observed on a log scale.

5.1 Simulations Using a Sequence of Shifted Images

In this section, we evaluate the performance of the proposed KLT uncertainty estimation method using a sequence of shifted images. We create a test image sequence by shifting an image I by known values. Points features (such as Harris corners or minimum eigenvalue features) are detected in I and tracked through the image sequence by the KLT tracker. The theoretical error is computed by calculating the KLT error surface and its local minima. The watershed transform

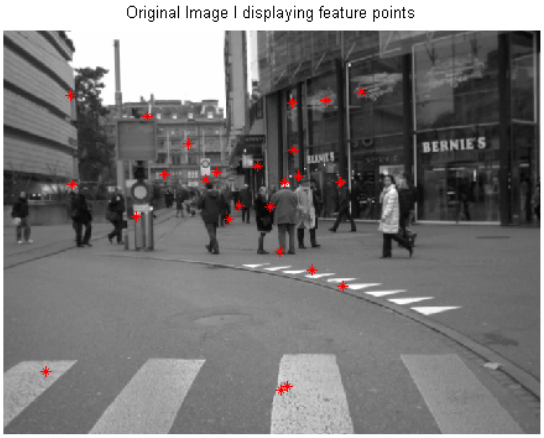

Fig. 3. Street scene image used for KLT error evaluation and the 25 tracked feature points.

(Meyer and Beucher [15]) is used to calculate the basins of attraction of the minima near the initial point. Since the watershed transform does not assign boundary pixels to basins, this computation is done at a higher resolution to prevent ambiguities at the basin borders. For pyramidal KLT, the computation is done by upsampling the higher level image back to the finest scale to maintain accuracy of the minima locations. The error propagation starts with the feature point detection error and is then propagated through pyramidal KLT levels as well as different image frames as long as experimental KLT tracking converges.

Next, Monte Carlo simulations are performed by adding Gaussian noise to each image before tracking. We calculate ANEES by aggregating the experimental KLT results from 25 Monte Carlo iterations and using the theoretical error covariance given by Equation 17. We discard the KLT tracks that do not converge due to numerical issues.

We detect 25 minimum eigenvalue feature points in a test image, as shown in Figure 3. Our first experiment evaluates the error estimates for a pair of frames for both single level and pyramidal KLT trackers. The graphs in Figure 4 plot the ANEES value versus different added noise levels for the tracked feature points. The ANEES values are close to 1 for most points even as noise levels increase. The deviation from 1 reflects the limitation of the Hessian approximation for error estimation in non-linear least squares.

The next experiment evaluates the estimates over a sequence of 5 frames (numbered 0–4) for both single level and pyramidal KLT tracking. These ANEES plots are shown in Figure 5 for the single level KLT and in Figure 6 for the 2 level pyramidal KLT.

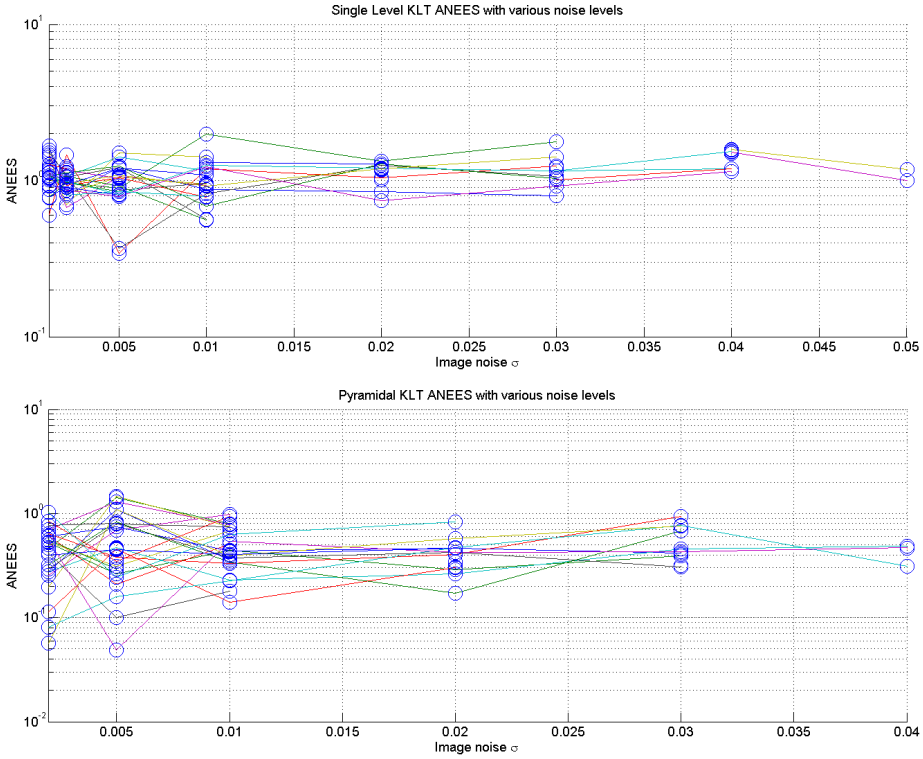

Fig. 4. ANEES values for single level (top) and 2 level pyramidal (bottom) KLT tracker with increasing noise levels. A log plot is used since ANEES is the average of ratios, so equal upwards and downwards deviations from the ANEES=1 line correspond to equal estimation errors. Each trajectory corresponds to the tracking of a single point. The decreasing number of points for larger noise levels corresponds to the fact that tracking fails more often for high noise. Image pixel values are in the range [0,1].

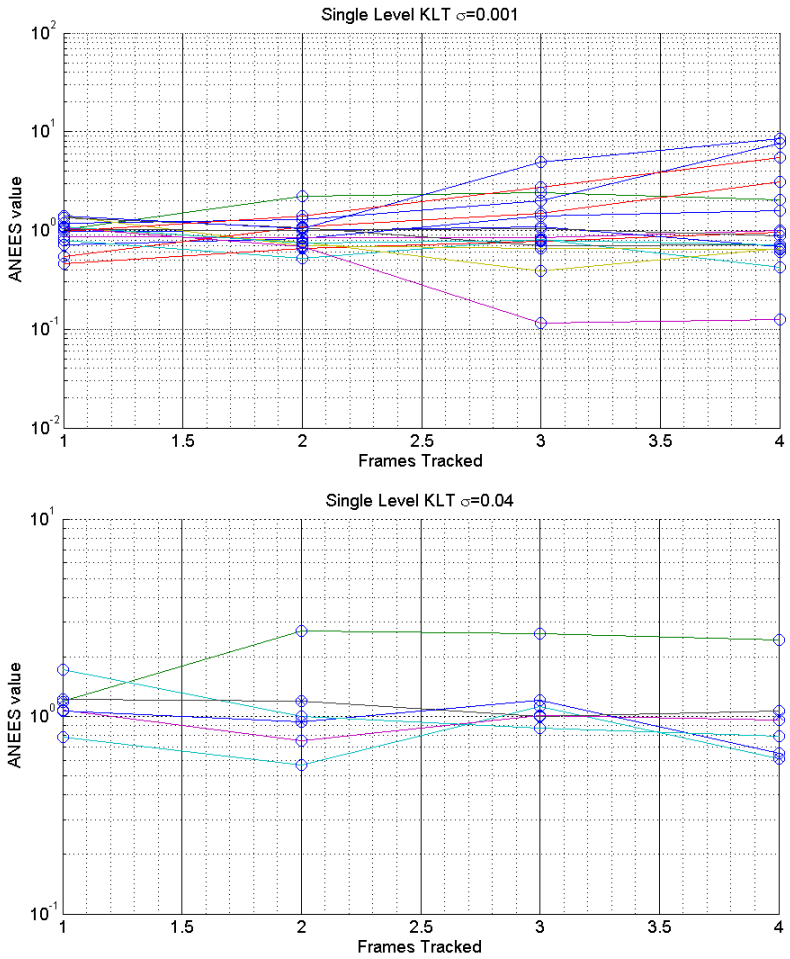

Fig. 5. ANEES for multi-frame single level KLT tracking at two different noise levels. All points out of 25 for which the KLT converged are shown. Image pixel values are in the range [0,1].

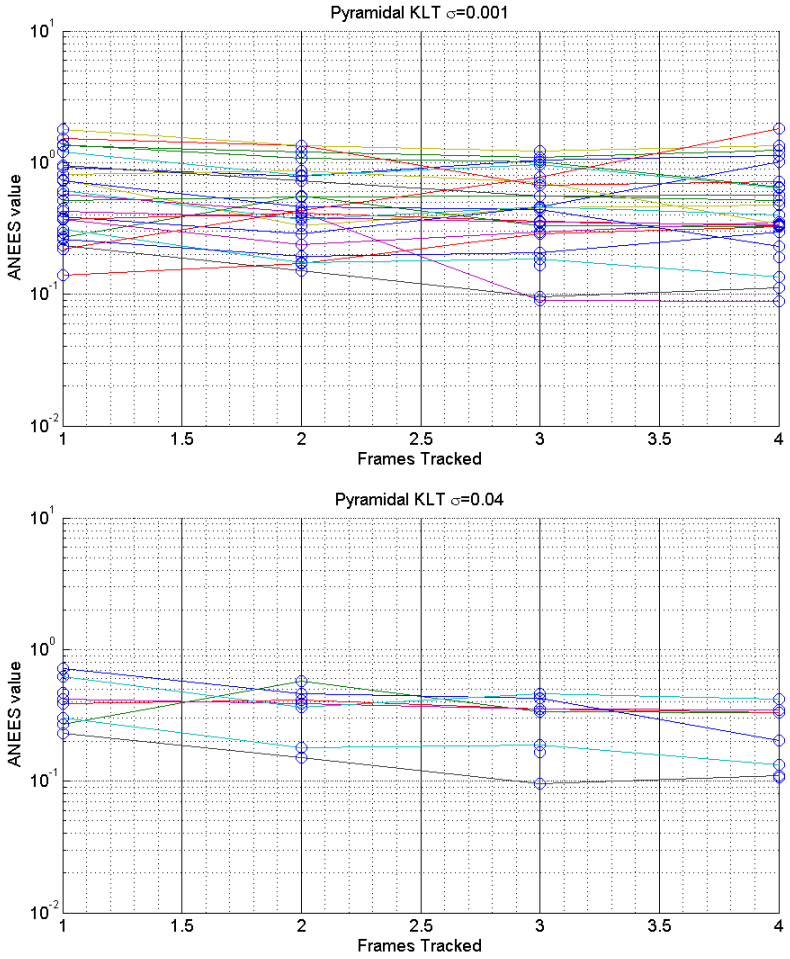

Fig. 6. ANEES for multi-frame pyramidal KLT tracking at two different noise levels. All points out of 25 for which the KLT converged are shown. Image pixel values are in the range [0,1].

6 Conclusion and Future Work

We have presented a novel comprehensive error analysis of the KLT tracker. We show that the error of the single level, pyramidal as well as the multi-frame KLT tracker is given by a Gaussian Mixture Model. The components of the mixture model correspond to the local minima that can trap the KLT tracker. Our Monte Carlo simulations show that our uncertainty estimates are accurate for these common use cases.

Our next steps will be to show improvements in real world applications by using the KLT error estimates. This will require improving the speed with which the error estimates are calculated. Since computing watershed transforms for each tracked point can make real time tracking difficult, further work is necessary before the error estimates can be used in practical systems. We would also like to extend the error analysis to allow affine image patch deformations during tracking. Another research direction is exploring new ways to modify the KLT error function to reduce the prevalence of local minima, without compromising its speed.

Acknowledgement

This research was supported by Air Force Research Laboratory (AFRL) under contract FA8650-13-M-1701 with UtopiaCompression Corporation.

References

1. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of the 7th international joint conference on Artificial intelligence - Volume 2. IJCAI'81, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (1981) 674–679
2. Tomasi, C., Kanade, T.: Detection and tracking of point features. Technical report, School of Computer Science, Carnegie Mellon Univ. (1991)
3. Shi, J., Tomasi, C.: Good features to track. In: , 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94. (1994) 593–600
4. Bouguet, J.Y.: Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. Intel Corporation **5** (2001)
5. Baker, S., Matthews, I.: Lucas-Kanade 20 Years On: A Unifying Framework. Int. J. Comput. Vision **56** (2004) 221–255
6. Triggs, B., McLauchlan, P., Hartley, R., Fitzgibbon, A.: Bundle adjustment – a modern synthesis. Vision algorithms: theory and practice (2000) 153–177
7. Kanazawa, Y., Kanatani, K.i.: Do we really have to consider covariance matrices for image features? In: Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on. Volume 2. (2001) 301–306
8. Nickels, K., Hutchinson, S.: Estimating uncertainty in SSD-based feature tracking. Image and vision computing **20** (2002) 47–58

9. Orguner, U., Gustafsson, F.: Statistical characteristics of harris corner detector. In: Statistical Signal Processing, 2007. SSP 2007. IEEE/SP 14th Workshop on. (2007) 571–575
10. Zeisl, B., Georgel, P.F., Schweiger, F., Steinbach, E.G., Navab, N., Munich, G.E.R.: Estimation of location uncertainty for scale invariant features points. In: BMVC. (2009) 1–12
11. Brooks, M.J., Chojnacki, W., Gawley, D., Van Den Hengel, A.: What value covariance information in estimating vision parameters? In: Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on. Volume 1., IEEE (2001) 302–308
12. Pfeiffer, D., Gehrig, S., Schneider, N.: Exploiting the power of stereo confidences. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE (2013) 297–304
13. Seber, G.A.F., Wild, C.J.: Nonlinear regression. Wiley, New York (1989)
14. Li, X.R., Zhao, Z., Jilkov, V.P.: Practical measures and test for credibility of an estimator. In: Proc. Workshop on Estimation, Tracking, and Fusion–A Tribute to Yaakov Bar-Shalom, Citeseer (2001) 481–495
15. Meyer, F., Beucher, S.: Morphological segmentation. Journal of visual communication and image representation **1** (1990) 21–46