Uncertainty Quantification of Lucas Kanade Feature Track and Application to Visual Odometry

Xue Iuan Wong
University at Buffalo
xuewong@buffalo.edu

Manoranjan Majji
Texas A&M University LASR Lab.
mmajji@tamu.edu

Abstract

An uncertainty quantification approach to estimate the errors incurred by the Kanade Lucas Tomasi (KLT) feature tracking algorithm is presented. The covariance analysis is based on the linearized sensitivity calculations of the KLT algorithm. Track uncertainty thus computed is utilized to quantify the errors associated with feature based relative pose estimation algorithms. This paper shows that the uncertainty analysis results serve as a means of measuring the reliability of feature correspondences. One of the application of resulting uncertainty analysis is to serves as a criteria to eject badly track feature. An experiment based on visual odometry result is given to demonstrate this functionality.

1. Introduction

Image features can be defined as the scale-space extrema[13]. A large body of work has been carried out in image analysis to automatically provide a feature based characterization of an image using its local texture. SIFT[15], SURF[2], GLOH[20], and FAST[25] are examples of a few popular feature extraction methods. Given a set of automatically extracted feature points, we are interested in using them to establish correspondence across images for localization and mapping. Feature tracking methods are among the most efficient approaches for providing a practical solution to the feature correspondence problem, owing to the construction of texture based descriptors[3, 16, 20]. Computational efficiency may be achieved in maintaining feature correspondence in continuous images in a stream by tracking them across successive images in the stream.

Kanade Lucas Tomasi(KLT) feature tracker developed by Kanade, Lucas and Tomasi [16, 30, 29, 1] remains a popular choice for feature tracking. Tracks generated in this propagation process are used in feature based relative navigation applications[23, 26, 10, 4, 19, 17]. The KLT tracker serves as an essential intermediate step in a sequence of operations for relative navigation and surface reconstruction. To this end, the uncertainty of KLT tracker is essential to quantify the accuracy of localization and mapping. A systematic approach to estimate uncertainty properties of KLT tracking result is an integral part of an image processing pipeline determining the health of established feature correspondences. False matches are common source of localization error. Traditionally, this problem is handled by an application of the Random Sample Consensus (RANSAC)[9] algorithm on the feature tracks. RANSAC stipulate an objective matric to distinguish outliers from inliers. An alternative approach is to reject outliers based on the probability score of the feature track[5, 32]. The result of proposed algorithm can be utilized to access this probability score.

Kanazawa and Kanatani [12] propose a method to compute the error covariance of a feature point from a gray scale image using local texture gradient. Although they do not consider propagation of feature uncertainty from one image to another, they show that uncertainty of a feature point is related to the curvature of its local intensity. Nickels and Hutchinson [21] propose a method to estimate uncertainty of sum of squared differences (SSD) based feature tracking approaches that include the KLT tracker. In their work, uncertainty of feature track is modeled as confidence score of matched features, computed by taking SSD of origin and tracked pixels patch's intensity. Although this approach provides a reasonable measure of tracking accuracy, it does not explicitly derive the covariance from the image gradients. Work done by Sheorey et. al.[28] uses probability theory to model the chance that KLT tracker gets trapped at a local minimum during the tracking operations. Gaussian mixture model is used for error covariance calculations. Dorini and Goldenstein[8] propose the unscented KLT tracker for refining the KLT tracking result and estimate the track's local uncertainty. Unscented KLT tracker first applies the unscented transform (UT)[11] to each of the original feature points and obtains a set of sigma points. The sigma points are tracked to next frame with the KLT tracker. Mean value and the covariance associated with the distribu-

tion of the tracked features are then computed from tracked sigma points. There is a little insight about the choice of the sigma points pertaining to the initial uncertainty. Reinitializing the KLT tracker on sigma points has the shortcoming that these points on image may not yield successful tracks, owing to the lack of texture away from the central feature. This precludes effective uncertainty quantification.

Recent work by Ye et. al. [32] utilizes a probabilistic voting model based on a Bayesian network to discard invalid feature tracks. Their method first computes the homography matrix based on an initial set of feature correspondences. A joint probability model of the homography obtained for the initial set of feature correspondences is taken as the sample variance of the transfer error of the set of neighboring points around the target feature. Estimated homography enables the transfer error computation by a projection of the original features on the target image. The minimum transfer error of pixels in the neighborhood is compared against a threshold value to determine the probabilistic voting model of the feature correspondence. While this method is very useful in determining whether a detected feature is reliable or not based on the assumption that the neighboring texture fits a plane that remains consistent through successive frames, it does not capture th uncertainty propagation of the tracker. Note that, if one is interested in the errors obtained by the tracking function alone, based on the texture gradients, the homography transformation may not capture the deformation process in time.

In this paper, we propose to model propagation of feature point uncertainty based on the sensitivity analysis of local linearized approximation of the track obtained by the KLT tracker. Note that KLT based methods subsume a wide set of approaches that range from original LT tracker [16] to the recently reported machine learning augmented KLT tracker (such as Extended Lucas Kanade (ELK)[22] and Conditional Lucas Kanade (CLK)[14]). These approaches rely on image gradient to predict the feature motion and deformation. Owing to the fact that the proposed gradient based uncertainty analysis uses exactly the same information as the tracker, proposed algorithm does no entail additional computational expense. This is also indicates that the analysis presented in this paper is compatible to other KLT based methods. Note that it is also more efficient than the unscented transformation, since additional feature tracks need not be generated. Further, the aforementioned issue of loss of feature tracks on sigma points is avoided.

In applications involving vision based Simultaneous Localization and Mapping(SLAM) and Visual Odometry(VO), the ability of registering common feature across images is crucial. Although it is more common to use feature descriptor matching to establish feature correspondence due to its robustness, this may be computationally expensive. Various successful pipeline architectures[23, 26, 10, 4, 19, 17] show that it is more computationally efficient to replace descriptor matching step with feature tracking for continuous images in a stream. Since navigation filters for SLAM depend upon uncertainty quantification of the feature measurements to derive statistically optimal solution, the feature track uncertainty is vital for effective feature based navigation.

This paper is organized as follows. Section 2 provides a review of the mathematical formulation and principles of the KLT tracker. Section 3 formulates the linear covariance analysis of KLT tracker. Section 4 discusses the application of proposed uncertainty quantification model to visual odometry. Section 5 reports the experiment results. Section 6 draws some conclusions of the work presented in this paper.

2. KLT Tracker

Consider two successive images I and J of an object obtained from two close but distinct baseline locations and orientations. Let $\mathbf{u} \in \mathbb{R}^{2\times 1}$ be the image space coordinate of a feature point that assumes an intensity value of $I(\mathbf{u})$ in image I. The image space deformation model is assumed to be $\mathbf{v} = \mathbf{u} + \mathbf{d} \in \mathbb{R}^{2\times 1}$, where $\mathbf{d} \in \mathbb{R}^{2\times 1}$ is the translation vector in image space. Using this deformation model in the subsequent image, the brightness constancy assumption provides the relationship given by:

$$I(\mathbf{u}) = J(\mathbf{u} + \mathbf{d}) \tag{1}$$

New feature location $\mathbf{v}$ specified in image J corresponds to the point $\mathbf{u}$ in image I. The objective of KLT tracker is to estimate the translation vector $\mathbf{d}$. Note that more general deformation models can be utilized[1] for advecting the feature.

The KLT tracker estimates $\mathbf{d}$ with local intensity gradient information, under the brightness constancy assumption[1]. This assumption stipulates that $J(\mathbf{v}) = J(\mathbf{u} + \mathbf{d})$. The texture around $\mathbf{u}$ in image I also gets advected to neighborhood of $\mathbf{u} + \mathbf{d}$ in image J. This localized advection of the image field forms the basis for KLT tracker. Applying the first order Taylor series expansion to this equation leads to:

$$I(\mathbf{u}) - J(\mathbf{u}) \approx d_x \frac{\partial J(\mathbf{u})}{\partial x} + d_y \frac{\partial J(\mathbf{u})}{\partial y} = J_x d_x + J_y d_y \tag{2}$$

Consider a search window of size $w \times h$ with its center located at $\mathbf{u}$. The symbols w and h indicate the search window's width and height respectively. Also assume that the motion of all pixels within the search window is identical to motion of $\mathbf{u}$, therefore we have a set of $w \times h$ pixels for which Eq.2 is valid. Translation vector $\mathbf{d}$ is then solved by posing an optimization problem to find $\mathbf{d}$ that minimizes Eq.2 over the search window of interest. Writing Eq.2 for each pixel patch in the window, we get:

$$A \begin{bmatrix} d_x \\ d_y \end{bmatrix} = B \tag{3}$$

Where,

$$A = \begin{bmatrix} \frac{\partial J(\mathbf{u}_1)}{\partial x} & \frac{\partial J(\mathbf{u}_1)}{\partial y} \\ ... & ... \\ \frac{\partial J(\mathbf{u}_n)}{\partial x} & \frac{\partial J(\mathbf{u}_n)}{\partial y} \end{bmatrix} \tag{4}$$

$$B = \begin{bmatrix} I(\mathbf{u}_1) - J(\mathbf{u}_1) \\ ... \\ I(\mathbf{u}_n) - J(\mathbf{u}_n) \end{bmatrix} \tag{5}$$

and $\mathbf{u}_1$, $\mathbf{u}_2$, ..., $\mathbf{u}_n$ are image point coordinates in the search window. The translation vector that minimizes the sum of square difference of Eq.3 using least squares estimation is given by:

$$\mathbf{d} = (A^T W A)^{-1} A^T W B \tag{6}$$

where matrix W is a weighting matrix. Following Lucas and Kanade[16], diagonal elements of weighting matrix are defined to be proportional to the difference in the image gradient. This choice of W incorporates the intuition that the corresponding feature points are not only intensity invariant, but also similar in their 1st order derivatives. Therefore, each diagonal element of weighting matrix can be written as:

$$w_{ii} = \frac{1}{|I'(\mathbf{u}) - J'(\mathbf{u})|}$$

Due to the error introduced by ignoring the higher order terms of the Taylor series expansion, an estimate of $\mathbf{d}$ obtained using Eq.6 is generally inaccurate. The loss of accuracy can also be attributed to evaluation of the image gradients about the approximate feature location. To resolve this problem, the KLT tracker incorporates the use of Newton-Raphson style iteration scheme to update $\mathbf{d}$ iteratively until convergence is achieved. Convergence criteria is measured in terms of the error between the original window and the advected feature in subsequent frames, i.e. $||I(\mathbf{u}) - J(\mathbf{u} + \hat{\mathbf{d}})||$.

Our analysis of the KLT tracker uncertainty propagation model is naturally compatible with standard and pyramid implementations of the tracker. It also provides a basis for uncertainty analysis of the affine implementation. To ease the complexity of this paper, only the standard KLT tracker is considered for detailed analysis. Discussions pertaining to the details of pyramidal and affine implementations of the KLT tracker are delegated to excellent papers in the literature[1, 3, 29].

3. Uncertainty Analysis of KLT Tracker

The principle objective of the uncertainty analysis of the KLT tracker is to develop an approach that provides a probability value for the tracked feature being the true solution. In this research, we attempt to derive this information by using the track sensitivity. The term sensitivity refers to the variation of the response of the tracking algorithm owing to the changes in the initial feature point location. We assume that the source of errors that corrupt the tracking result are introduced from two distinct sources. They are the uncertainty of initial feature point location, and errors incurred by the implementation of the KLT tracker. Consider the relation between a deterministic feature point $\mathbf{u} \in \mathbb{R}_{2\times1}$, its KLT tracked translation $\mathbf{d} \in \mathbb{R}_{2\times1}$, and a deterministic location $\mathbf{v} \in \mathbb{R}_{2\times1}$ in tracked image written as:

$$\mathbf{v} = \mathbf{u} + \mathbf{d}$$

In practice, $\mathbf{u}$ is uncertain. To account for this, the initial feature uncertainty is modeled as $\delta\mathbf{u}$, such that the estimated feature location is written as:

$$\tilde{\mathbf{u}} = \mathbf{u} + \delta\mathbf{u} \tag{7}$$

where $\delta\mathbf{u}$ is a random vector of with a mean value of $\mathbf{0} \in \mathbb{R}_{2\times1}$ and covariance matrix of $\Sigma_{\mathbf{u}} \in \mathbb{R}_{2\times2}$.

The additive model for initial feature uncertainty is a reasonable approximation in practice. It is particularly well suited for small feature displacements considered in the developments here. To accommodate our second source of uncertainty, we assume that the tracking process yields uncertainty $\delta\mathbf{d} \in \mathbb{R}_{2\times1}$ that is also a zero mean random variable with covariance $\Sigma_{\mathbf{d}} \in \mathbb{R}_{2\times2}$. The tracked feature location corresponding to the corrupted initial feature point due to the combined effects of the initial feature uncertainty and the implementation error can be written as:

$$\tilde{\mathbf{v}} = \tilde{\mathbf{u}} + \tilde{\mathbf{d}} \tag{8}$$

where $\tilde{\mathbf{d}}$ is the estimate of displacement vector solved by KLT tracker at location $\tilde{\mathbf{u}}$. This vector in general does not equal to the unknown exact displacement $\mathbf{d}$. This is because of the fact that the local image gradient is evaluated about the estimated initial feature location. Error in $\mathbf{v}$ can thus be modeled as:

$$\delta\mathbf{v} = \tilde{\mathbf{v}} - \mathbf{v} = \left(\tilde{\mathbf{d}} - \mathbf{d}\right) + \delta\mathbf{u} = \delta\mathbf{d} + \delta\mathbf{u} \tag{9}$$

Note that $\delta\mathbf{d}$ is correlated with $\delta\mathbf{u}$. This is because KLT process inherently depends on the local image gradients about the estimated feature location. Covariance of the estimated feature location's uncertainty in subsequent frame can thus be computed as:

$$\begin{aligned}\Sigma_{\mathbf{v}} &= E[\delta\mathbf{v}\delta\mathbf{v}^T] \\ &= E[\delta\mathbf{d}\delta\mathbf{d}^T] + E[\delta\mathbf{u}\delta\mathbf{u}^T] + E[\delta\mathbf{u}\delta\mathbf{d}^T] + E[\delta\mathbf{d}\delta\mathbf{u}^T]\end{aligned} \tag{10}$$

$$\Sigma_{\mathbf{v}} = \Sigma_{\mathbf{d}} + \Sigma_{\mathbf{u}} + E[\delta\mathbf{u}\delta\mathbf{d}^T] + E[\delta\mathbf{d}\delta\mathbf{u}^T] \tag{11}$$

Initial feature point uncertainty may be evaluated by various means in addition to models proposed by Shorey et. al.[28]. Scale space feature extraction and description insights from SIFT[15], SURF[2] and GLOH[20] may be used to determine $\Sigma_{\mathbf{u}}$. When tracking features over successive frames, $\Sigma_{\mathbf{u}}$ is the $\Sigma_{\mathbf{v}}$ from a propagated step, resulting in a recursive process for uncertainty quantification.

$\Sigma_{\mathbf{d}}$ is the covariance matrix of errors introduced by KLT tracker due to the variation of the initial feature point location. Therefore, it is a function of $\Sigma_{\mathbf{u}}$. We propose to estimate $\Sigma_{\mathbf{d}}$ based on the sensitivity analysis. Uncertainty in translation can be written as:

$$\delta\mathbf{d} = \frac{\partial\mathbf{d}}{\partial x}\delta x + \frac{\partial\mathbf{d}}{\partial y}\delta y = \mathbf{d}_x\delta x + \mathbf{d}_y\delta y \tag{12}$$

where x, y denote the image space coordinates of the feature and δx, δy indicate their variability. The partial derivatives representing the sensitivity are evaluated about the nominal feature point location.

Taking the expected value of the outer product, covariance function can be written as:

$$\Sigma_{\mathbf{d}} = E[\delta\mathbf{d}\delta\mathbf{d}^T] \tag{13}$$

Since gradient is evaluated at a point in image space, typically at the converged estimate $\mathbf{d}$, local intensity gradient is treated to be deterministic and evaluated about this realization.

$$E[\delta\mathbf{d}\delta\mathbf{d}^T] = \begin{bmatrix} \mathbf{d}_x & \mathbf{d}_y \end{bmatrix} E\left[\begin{bmatrix} \delta x\delta x^T & \delta x\delta y^T \\ \delta y\delta x^T & \delta y\delta y^T \end{bmatrix}\right] \begin{bmatrix} \mathbf{d}_x^T \\ \mathbf{d}_y^T \end{bmatrix} \tag{14}$$

Eq.14 can thus be written as:

$$\Sigma_{\mathbf{d}} = \begin{bmatrix} \mathbf{d}_x & \mathbf{d}_y \end{bmatrix} \Sigma_{\mathbf{u}} \begin{bmatrix} \mathbf{d}_x^T \\ \mathbf{d}_y^T \end{bmatrix} \tag{15}$$

Sensitivities $\mathbf{d}_x$, $\mathbf{d}_y$ in Eq.15 are computed by first considering:

$$A\mathbf{d} = B \tag{16}$$

where matrix A and B are defined in Eqs.4 and 5 respectively. Taking partial derivative of Eq.16 with respect to x leads to:

$$\mathbf{d}_x = \frac{\partial A}{\partial x}\mathbf{d} + A\frac{\partial\mathbf{d}}{\partial x} = \frac{\partial B}{\partial x} \tag{17}$$

Rearranging terms in Eq.17, we get:

$$\frac{\partial\mathbf{d}}{\partial x} = (A^T W A)^{-1} A^T W \left(\frac{\partial B}{\partial x} - \frac{\partial A}{\partial x}\mathbf{d}\right) \tag{18}$$

Terms $\mathbf{d}_y$ can derived following to the same procedure as in Eqs.17 and 18, yielding:

$$\mathbf{d}_y = \frac{\partial\mathbf{d}}{\partial y} = (A^T W A)^{-1} A^T W \left(\frac{\partial B}{\partial y} - \frac{\partial A}{\partial y}\mathbf{d}\right) \tag{19}$$

Note that the existence of solution of Eq.16 guarantees the existence of solution to the track sensitivity in Eqs.18 and 19. We contrast this solution with UT approach in that repeated evaluation of tracks about different sigma points may be necessary to provide a covariance estimate using the UT.

Following the definition of matrix B in Eq.5, we have:

$$\frac{\partial B}{\partial x} = \begin{bmatrix} \frac{\partial I(\mathbf{u})}{\partial x} - \frac{\partial J(\mathbf{u})}{\partial x}|_{\mathbf{u}=\mathbf{u}_1} \\ \dots \\ \frac{\partial I(\mathbf{u})}{\partial x} - \frac{\partial J(\mathbf{u})}{\partial x}|_{\mathbf{u}=\mathbf{u}_n} \end{bmatrix} \tag{20}$$

$$\frac{\partial B}{\partial y} = \begin{bmatrix} \frac{\partial I(\mathbf{u})}{\partial y} - \frac{\partial J(\mathbf{u})}{\partial y}|_{\mathbf{u}=\mathbf{u}_1} \\ \dots \\ \frac{\partial I(\mathbf{u})}{\partial y} - \frac{\partial J(\mathbf{u})}{\partial y}|_{\mathbf{u}=\mathbf{u}_n} \end{bmatrix} \tag{21}$$

where $\mathbf{u}_1, ..., \mathbf{u}_n$ are the image space coordinates in the search window about the i^{th} feature $\tilde{\mathbf{u}}_i$. The column vectors $(\partial A/\partial x)\mathbf{d}$ and $(\partial A/\partial y)\mathbf{d}$ are written as:

$$\frac{\partial A}{\partial x}\mathbf{d} = \begin{bmatrix} \frac{\partial^2 J(\mathbf{u})}{\partial x^2}d_x + \frac{\partial^2 J(\mathbf{u})}{\partial x\partial y}d_y|_{\mathbf{u}=\mathbf{u}_1} \\ \dots \\ \frac{\partial^2 J(\mathbf{u})}{\partial x^2}d_x + \frac{\partial^2 J(\mathbf{u})}{\partial x\partial y}d_y|_{\mathbf{u}=\mathbf{u}_n} \end{bmatrix} \tag{22}$$

$$\frac{\partial A}{\partial y}\mathbf{d} = \begin{bmatrix} \frac{\partial^2 J(\mathbf{u})}{\partial x\partial y}d_x + \frac{\partial^2 J(\mathbf{u})}{\partial y^2}d_y|_{\mathbf{u}=\mathbf{u}_1} \\ \dots \\ \frac{\partial^2 J(\mathbf{u})}{\partial x\partial y}d_x + \frac{\partial^2 J(\mathbf{u})}{\partial y^2}d_y|_{\mathbf{u}=\mathbf{u}_n} \end{bmatrix} \tag{23}$$

Note that since the uncertainty propagation model is linearized around converged solution, translation vector $\mathbf{d}$ used in Eqs.22 - 23 is the solution of Eq.6 at final iteration (i.e. converged solution).

Using the definition of $\delta\mathbf{d}$ in Eq.12, we can derive the last two terms of Eq.11 as follows.

$$E\{\delta\mathbf{u}\delta\mathbf{d}^T\} = E\left\{\delta\mathbf{u}\delta\mathbf{u}^T \begin{bmatrix} \frac{\partial\mathbf{d}}{\partial x} & \frac{\partial\mathbf{d}}{\partial y} \end{bmatrix}^T\right\} = \Sigma_{\mathbf{u}} \begin{bmatrix} \frac{\partial\mathbf{d}}{\partial x} & \frac{\partial\mathbf{d}}{\partial y} \end{bmatrix}^T \tag{24}$$

$$E\{\delta\mathbf{d}\delta\mathbf{u}^T\} = \begin{bmatrix} \frac{\partial\mathbf{d}}{\partial x} & \frac{\partial\mathbf{d}}{\partial y} \end{bmatrix} \Sigma_{\mathbf{u}} \tag{25}$$

Using identities of Eqs.24-25 along with the expression for $E\left\{\delta\mathbf{d}\delta\mathbf{d}^T\right\}$, Eq.11 can be written as:

$$\Sigma_{\mathbf{v}} = \begin{bmatrix} \mathbf{d}_x & \mathbf{d}_y \end{bmatrix} \Sigma_{\mathbf{u}} \begin{bmatrix} \mathbf{d}_x^T \\ \mathbf{d}_y^T \end{bmatrix} + \Sigma_{\mathbf{u}} + \Sigma_{\mathbf{u}} \begin{bmatrix} \mathbf{d}_x^T \\ \mathbf{d}_y^T \end{bmatrix} + \begin{bmatrix} \mathbf{d}_x & \mathbf{d}_y \end{bmatrix} \Sigma_{\mathbf{u}} \quad (26)$$

Note that our formulation of $\mathbf{d}_x$ and $\mathbf{d}_y$ involve the local linearized approximations evaluated at converged translation estimate $\mathbf{d}$. To legitimize the truncation of the high order terms in the Taylor series expansion of $\mathbf{d}$, it is necessary that the initial feature uncertainty be small (i.e. $\Sigma_{\mathbf{u}}$ is small) in order to maintain the bounded growth of the track uncertainty. Nature of uncertainty propagation indicates heuristically that propagated error can not be smaller than original uncertainty: $\Sigma_{\mathbf{u}} \leq \Sigma_{\mathbf{v}}$
This is because no new information is available to improve upon the error of the original feature, about which the gradients are evaluated. Further, the errors incurred in the forward track propagation process are non decreasing, depending on the viewpoint, illumination and image deformation associated with the image formation process. Based on this intuition, we know that propagated uncertainty will eventually violate the assumptions in the KLT tracker. To quantitatively monitor this growth, we propose a health metric associated with each feature track as a function of its propagated uncertainty. When the principle axis length of $\Sigma_{\mathbf{v}}$ is larger than a predefined threshold, we infer that such feature can no longer be tracked accurately and therefore should be re-initialized. Although our uncertainty propagation is conservative, this heuristic measure is important for guidance and control decision logics based upon surface reconstruction and feature based localization operations. Furthermore, when tracks associated with independently moving objects are available in the same field of view, the eigenvalues and eigenvectors of $\Sigma_{\mathbf{v}}$ serve as a means of prediction of feature track intersection. Such features with overlapping covariance ellipses can therefore be suppressed to avoid gross errors due to feature disambiguation.

The mathematical formulation of the feature uncertainty propagation through sensitivity analysis of KLT-tracker is now detailed. Fig.1 is a summary of the feature track uncertainty characterization process. First, the feature point is initialized through a chosen feature extraction algorithm, and the feature track uncertainty is initialized with image feature uncertainty method. For every KLT-tracking step, we estimate the propagated feature uncertainty using Eq.26. Resulting $\Sigma_{\mathbf{v}}$ is the estimated feature uncertainty in the tracked frame. Its principle axes lengths are then tested against pre-defined threshold values. If the principle axis length of $\Sigma_{\mathbf{v}}$ is larger than the threshold, we conclude that the feature uncertainty is too large and is unsuitable for further tracking, and the track is marked for termination. On the other hand, if $\Sigma_{\mathbf{v}}$ is smaller than the threshold value, $\Sigma_{\mathbf{u}}$ is updated with $\Sigma_{\mathbf{v}}$ for propagation on to sub-sequent frames.

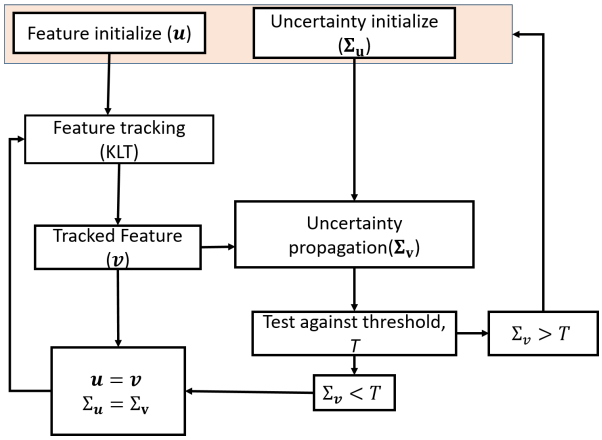


Figure 1: Flowcart of proposed feature uncertainty propagation pipeline

Note that all the information required for computation of sensitivity used in uncertainty quantification process beside of the second order derivatives terms are available from intermediate steps of the KLT tracker. Therefore, the computational cost of proposed algorithm is less than alternative approaches for uncertainty quantification.

4. Application to Visual Odometry

Given the fact that the core of several VO solutions is an optimization framework that minimize the error between the feature track measurements and the measurement model output with estimated parameters, the uncertainty information of the feature tracks can be use as a statistical confidence measure to improve the estimation accuracy.

Wrongly established feature correspondences or outliers are one of the critical source of errors in VO solutions. There is a need of the systematic and robust procedures to identify outliers prior to the estimation process. Common solution to this problem relies on iterative outlier ejection framework such as RANSAC. Alternatives to RANSAC such as ARRSAC[24] and PROSAC[6] uses the prior knowledge of the feature correspondence confidence measurement to accelerate the outlier ejection process in an iterative fashion identical to RANSAC. PROSAC has shown its improved performance over RANSAC, but majority of the VO solution are still preferring RANSAC. This is due to the fact that the lack of an efficient tool to compute the prior information required by PROSAC. The uncertainty quantification model proposed in early section of this paper can be serve as the source of the prior information required by PROSAC.

When the image motion satisfies assumptions of the KLT tracker, feature correspondences established by KLT tracker

has less outliers in comparison to other descriptor matching based algorithms. However, feature correspondence established by KLT tracker suffers from drift issues caused by affine transformation and image noise. The drift in feature location and the small amount of outliers cause the error propagation in VO solution. This paper proposes an aggressive threshold technique, i.e. implement a small threshold value T in Fig.1. This strategy allows only the feature correspondences with good uncertainty characteristic to propagate into future image frame, and be ultimately used in VO algorithm for further processing and thus reduces the VO drift.

5. Experiment

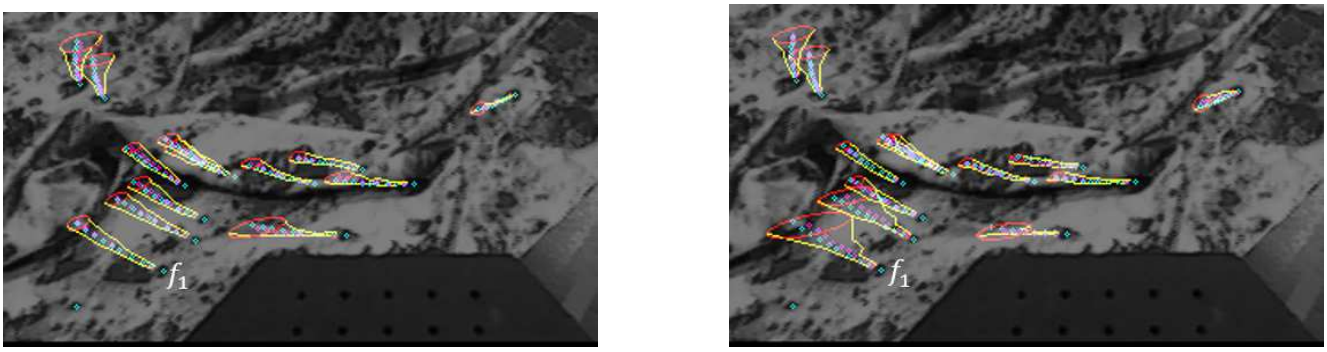
Two experiments are presented in this section to demonstrates the functionality of proposed uncertainty quantification method and its application to VO. The first experiment is designed to validate the uncertainty quantification of KLT tracker established feature tracks. A sequence of 10 images that are observations of an unstructured terrain-landing experiment[18] are inputs to this experiment. Asus Xtion Pro RGBD camera mounted on the Holonomic Omnidirectional motion emulator robot (HOMER)[7] is using for data collection. A set of 12 feature points are extracted from the first image, and tracked across 10 images with both estimated standard and pyramidal implementation of the KLT tracker. Proposed algorithm is then applied to each feature tracks for computes their covariance bound. When the image motion is small, both standard and pyramid implemented KLT tracker are expected to return identical results. However, one expects the pyramidal KLT tracker to yield more accurate results when there are large feature motions across images. Proposed KLT tracker uncertainty propagation approach is expected to be able to capture the tracking error incurred across different implementations. Experimental ground truth for this experiment is provided by pyramidal affine KLT tracker which is known to be of higher accuracy. These results are shown in Fig.2.

Fig.2 shows that some feature tracks slowly diverge from true solution because the pure translation model is used here. The true motion is a combination of rigid rotation and translation that cannot be approximated by pure translation model over large time periods. It is expected that the tracked feature diverges from true solution. It is of consequence to observe that the proposed uncertainty propagation model is able to capture the error divergence accurately. When comparing the tracking result between the standard KLT tracker and the pyramidal implementation, we see that some of the same feature tracks yield different results. This is caused by the magnitude of image motion across image frames. The proposed algorithm correctly captures the errors incurred by implementations in Fig.2(a) and (b), such that it is compatible with both pyramid implemented and standard KLT tracker.

For better illustration of the proposed algorithm's performance, feature f_1 in Fig.2 is zoomed in and the propagated covariance at each frame is plotted in Figs.3. Fig.3 shows the track of f_1 across the 10 frames, within a bounding box of 20×20 pixels. Covariance of the track at each frame is shown using the red ellipse. Fig.3(a) shows an overlay of the affine and pyramidal tracks of the feature. Similar plots for the results obtained using the standard KLT tracker are shown in Fig.3(d). Figs.3 clearly show that true feature location is always bounded by estimated $3-\sigma$ covariance through proposed algorithm. And the estimated uncertainty is representative of the actual error in the feature track. A comparison of Fig.3(a) and Fig.3(e) reveals that the standard KLT tracker incurs more error than the pyramidal implementation. However, the sensitivity calculations of the present work form a data driven basis for both of these algorithms to evaluate their own errors. A comparison of Figs.3(c-d) and Figs.3(f-h) reveals this fact.

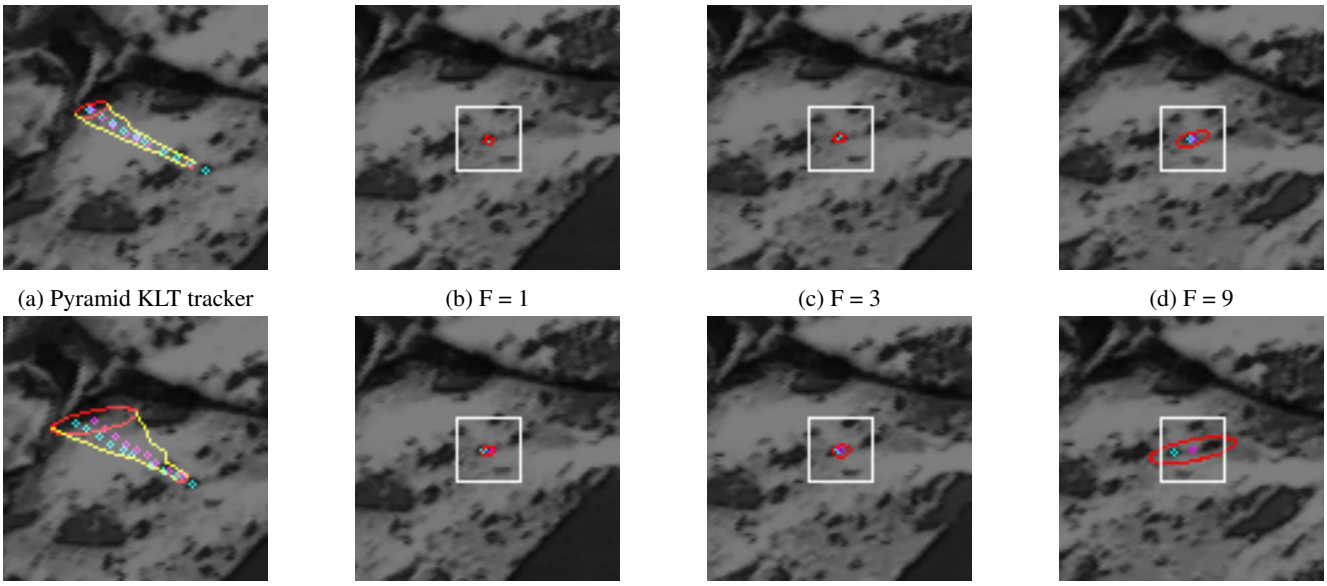
The second set of experiment is to demonstrates the application of proposed algorithm to VO pipeline. The VO algorithm that is used in this experiment is a stereo camera VO algorithm that recently reported in [31]. The input data to this experiment is also part of the HOMER project, but is captured with a Point Grey Bumblebee XB3 stereo camera. Initial features in this experiment are first extracted with SIFT method. KLT tracker is applied to computes both stereo and temporal feature correspondence, while the feature track uncertainties are computed with proposed algorithm. 200 stereo measurement frames are captured for this experiment, the relative camera pose across each frame is computed by stereo VO method introduced in [31]. A 3D surface is estimated from stereoapsis for every 20 frame, and transforms to initial camera frames for estimation of global 3D map. The covariance threshold value is configures to 2 for the covariance thresholding technique.The choice of the covariance threshold value can be computed from the uncertaity analysis of the VO algorithm. The covariance threshold is choose emperically, but it is also posible to select a statiscally meaningful threshold value based on the sensitivity analysis of the downstream pipeline. For example, when there is a specific accuracy in the stereo reconstruction 3D map is given, a sensitivity analysis base on the stereopsis can be used to compute the maximum covariance allowable from the feature track. And the length of the covariance principle axis can be selected as the covariance threshold. However, do note that the proposed uncertainty propagation procedure in KLT tracker is based on local linearization, uncertainty propagation is only valid within the range where the linearization assumption is valid.

Due to the lack of ground truth, the VO solutions of the paper are compared with the transformation estimated by iterative closest points(ICP) algorithm implemented in the



(a) (b)

Figure 2: Feature tracks and associated covariance estimates from (a) pyramid implemented KLT tracker(b) standard KLT tracker. Yellow line indicating location of principle $3-\sigma$ points at each frame, red ellipse indicating $3-\sigma$ bound at final frame, magenta dot indicating feature location tracked by KLT tracker, and cyan dot indicating true feature location



(a) Pyramid KLT tracker (b) F = 1 (c) F = 3 (d) F = 9

(e) Original KLT tracker (f) F = 1 (g) F = 3 (h) F = 9

Figure 3: (a - d) Feature tracks and propagated covariance of feature f_1 in Fig.2(a) with pyramid implementation of KLT. (e - h) with original LT tracker. Yellow line indicates propagated σ bound along with largest principle axis direction, red ellipse indicates σ bound at last frame. White dots indicate KLT tracked feature points. Blue dots indicate affine-pyramid KLT tracked feature points. (b - d), (f - h) Tracked feature and estimate error covariance bound at each frame, where F indicates frame index.

PCL library[27]. Although there is no guarentee the converged ICP solution is the absolute ground truth, it may be considered as the statistical optimal solution, owing to the fact that the number of 3D points using in ICP are much larger than that used by the VO. The errors in position estimated by VO with and without using thresholding technique are shown in Fig.5. Fig.5 clearly shows that the proposed thresholding technique yields better drift performance, as compared to the VO estimates obtained with no thresholding. Fig.4 shows the error in relative translation accros each stereo measurement step and the standard deviation bounds of the relative translation computed by the VO with feature track covariance as one of the input. Fig.4 shows that the differences between the error of relative translation is very small with RMS (root mean square) error of covariance thresholding given [0.008, 0.0045, 0.0012], while the RMS error of original VO solution as [0.0081, 0.0046, 0.0013]. Note that the accumulation of small errors leads to signif-

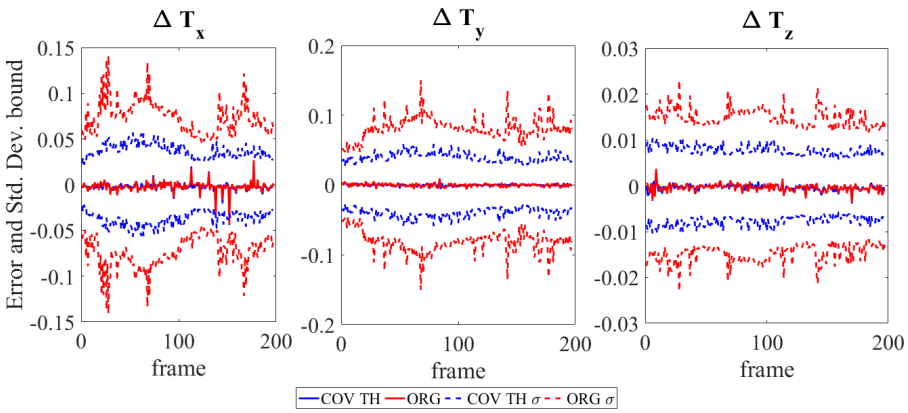


Figure 4: Error in VO estimate camera translation across each time step and corresdonding VO estimated standard deviation bound with covariance thresholding(COV TH) and original VO output(ORG)

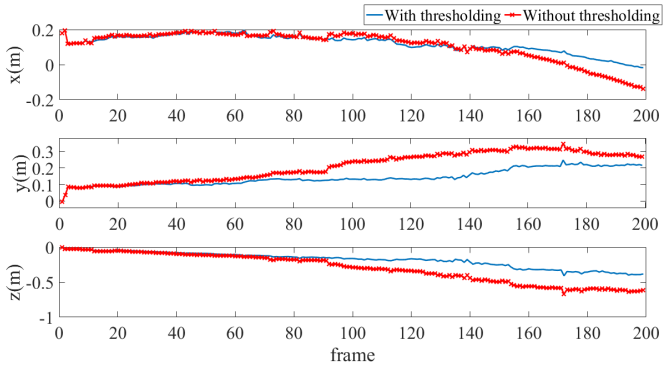


Figure 5: Error in position compare to labarotary truth generated with ICP

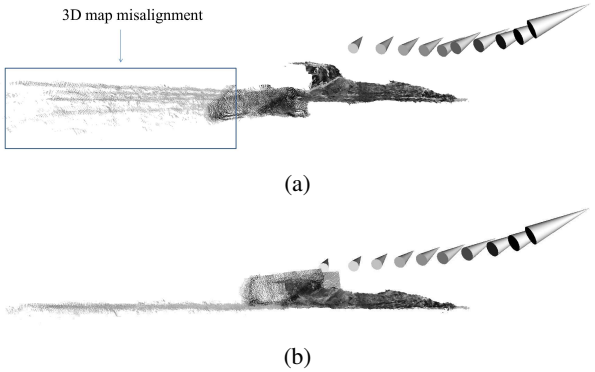


Figure 6: Side view of the reconstructed 3D map with VO algorithm, where cone indiactes camera location when a local 3D map is generated: (a) Without covariance thresholding technique. (b) With covariance thresholding technique. The alignment of stereo reconstructed 3D map indicates the accuracy of estimate camera pose.

icant drift after a long propagation time. Estimated standard deviation bounds of the VO output also indicates that covariance thresholding yields more accurate estimates. A comparison of the reconstructed global 3D map is shown in Fig.6, the 3D map misalignment in the original VO solution suggests the same observation where covariance thresholding results in a better drift characterization.

6. conclusion

An approach to evaluate the uncertainties associated with the feature tracks computed by using the KLT tracker and its application to visual odometry is presented in the paper. Linear covariance analysis is derived by utilizing the texture based image gradients evaluated about the feature track that uses a simple displacement model for image deformation. It is shown that the proposed approach yields consistent covariance bounds for both standard and pyramidal implementations of the tracker. More accurate affine invariant tracker is used in lieu of the true feature tracks when experiment data with no ground truth is available for validation of the proposed approach. Results of the proposed approach are utilized to evaluate the uncertainties of the relative pose estimates obtained under perspective and orthographic projection models. Uncertainties of the relative motion estimates obtained by the factorization approach are derived as a function of the feature track uncertainties that are supplied as an input to the algorithm. Although conservative, the approaches presented in the paper form a basis for optimism for the realization of an automated means of feature correspondence management. This is necessary in robust simultaneous localization and mapping applications. Images obtained from physical experiments are used to validate the methods presented in the paper.

References

[1] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *International journal of computer vision*, 56(3):221–255, 2004. 1, 2, 3

[2] H. Bay, T. Tuytelaars, and L. V. Gool. Surf : Speeded up robust features. In *Computer VisionECCV 2006*. Springer Berlin Heidelberg, 2006. 1, 4

[3] J.-Y. Bouguet. Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. *Intel Corporation*, 5:1–10, 2001. 1, 3

[4] Y. Cheng, M. Maimone, and L. Matthies. Visual odometry on the mars exploration rovers. In *2005 IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 903–910. IEEE, 2005. 1, 2

[5] M. Cho and K. M. Lee. Progressive graph matching: Making a move of graphs via probabilistic voting. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 398–405. IEEE, 2012. 1

[6] O. Chum and J. Matas. Matching with prosac-progressive sample consensus. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 220–226. IEEE, 2005. 5

[7] J. Davis, J. Doebbler, and J. Junkins. *Holonomic Omnidirectional Motion Emulation Robot at Texas A&M university Land, Air and Space Robotic Laboratory*. 6

[8] L. B. Dorini and S. K. Goldenstein. Unscented feature tracking. *Computer Vision and Image Understanding*, 115(1):8–15, 2011. 1

[9] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 1

[10] A. E. Johnson, S. B. Goldberg, Y. Cheng, and L. H. Matthies. Robust and efficient stereo feature tracking for visual odometry. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 39–46. IEEE, 2008. 1, 2

[11] S. J. Julier. The scaled unscented transformation. In *Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301)*, volume 6, pages 4555–4559. IEEE, 2002. 1

[12] Y. Kanazawa and K. Kanatani. 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, pages 301–306. IEEE, 2001. 1

[13] K. Koendrick. Solid shape, 1989. 1

[14] C.-H. Lin, R. Zhu, and S. Lucey. The conditional lucas & kanade algorithm. In *European Conference on Computer Vision*, pages 793–808. Springer, 2016. 2

[15] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999. 1, 4

[16] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *IJCAI*, 81, 1981. 1, 2, 3

[17] M. Maimone, Y. Cheng, and L. Matthies. Two years of visual odometry on the mars exploration rovers. *Journal of Field Robotics*, 24(3):169–186, 2007. 1, 2

[18] M. Majji, J. Davis, J. Doebbler, J. Junkins, B. Macomber, M. Vavrina, and J. Vian. Terrain mapping and landing operations using vision based navigation systems. In *Proc. AIAA Guid., Navigat., Control Conf*, 2011. 6

[19] L. Matthies, M. Maimone, A. Johnson, Y. Cheng, R. Willson, C. Villalpando, S. Goldberg, A. Huertas, A. Stein, and A. Angelova. Computer vision on mars. *International Journal of Computer Vision*, 75(1):67–92, 2007. 1, 2

[20] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE transactions on pattern analysis and machine intelligence*, 27(10):1615–1630, 2005. 1, 4

[21] K. Nickels and S. Hutchinson. Estimating uncertainty in ssd-based feature tracking. *Image and vision computing*, 20(1):47–58, 2002. 1

[22] S. Oron, A. Bar-Hille, and S. Avidan. Extended lucas-kanade tracking. In *European Conference on Computer Vision*, pages 142–156. Springer, 2014. 2

[23] M. Pollefeys, D. Nistér, J.-M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S.-J. Kim, P. Merrell, et al. Detailed real-time urban 3d reconstruction from video. *International Journal of Computer Vision*, 78(2-3):143–167, 2008. 1, 2

[24] R. Raguram, J.-M. Frahm, and M. Pollefeys. A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus. *Computer Vision–ECCV 2008*, pages 500–513, 2008. 5

[25] E. Rosten and T. Drummond. Fusing points and lines for high performance tracking. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1508–1515. IEEE, 2005. 1

[26] N. Rowell, S. Parkes, and M. Dunstan. Image processing for near earth object optical guidance systems. *IEEE Transactions on Aerospace and Electronic Systems*, 49(2):1057–1072, 2013. 1, 2

[27] R. B. Rusu and S. Cousins. 3d is here: Point cloud library (pcl). In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1–4. IEEE, 2011. 7

[28] S. Sheorey, S. Keshavamurthy, H. Yu, H. Nguyen, and C. N. Taylor. Uncertainty estimation for klt tracking. In *Computer Vision-ACCV 2014 Workshops*, pages 475–487. Springer, 2014. 1, 4

[29] J. Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 593–600. IEEE, 1994. 1, 3

[30] C. Tomasi and T. Kanade. *Detection and tracking of point features*. School of Computer Science, Carnegie Mellon Univ. Pittsburgh, 1991. 1

[31] X. I. Wong and M. Majji. A linear least square solution for stereo vision based localization and mapping application. In *IROS 2017 (Submitted for review, paper attached in supplementary material)*. IEEE/RSJ, 2017. 6

[32] S. Ye, C. Liu, Z. Li, and A. Al-Ahmari. Iterative optimization for frame-by-frame object pose tracking. *Journal of Visual Communication and Image Representation*, 44:72–81, 2017. 1, 2