The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Performance Evaluation of Visual SLAM Using Several Feature Extractors

Jonathan Klippenstein and Hong Zhang

Abstract— **Visual Simultaneous Localization And Mapping (SLAM) implementations must use feature extraction to reduce the dimensionality of image input, yet no comparison of feature extractors exists in the context of visual SLAM. This paper presents both a method for comparison of visual SLAM performance using several different feature extractors and the first experimental study using this method. Possible evaluation metrics are discussed and consistency testing and accumulated uncertainty are chosen to measure performance. Three feature extractors commonly used for visual SLAM are examined: the Harris corner detector, the Kanade-Lucas-Tomasi tracker, and the Scale-Invariant Feature Transform. All three are found to perform similarly in an indoor test environment, close to or within the limits of measurement. A modest scale change is handled without difficulty. We conclude that feature extractor choice is not significant in terms of visual SLAM performance and other criteria may be used to make the selection.**

I. Introduction

Simultaneous Localization And Mapping (SLAM) fuses the two complementary problems of mapping and localization to allow a mobile robot to navigate and map an unknown environment without any prior information [1]. As a robot travels through the world, uncertainty in its pose increases due to imperfect sensor readings. Landmarks observed in the world are added to a map, and re-observation acts to decrease the state uncertainty. Total uncertainty is kept from growing indefinitely and the robot is able to overcome noisy and incomplete sensor data.

Visual SLAM using cameras is motivated by the fact that cameras are information-rich, compact, and fast [2]. Access to the third dimension is easier than the traditional laser range finders that sweep out a two-dimensional plane parallel to the ground. Additionally, the maximum distance at which objects can be sensed is greater for cameras than laser- and acoustic-based sensors. In fact, points at "infinity" can be used to determine camera orientation [3]. We further consider the problem of bearing-only, or monocular, visual SLAM using a single camera. In contrast to stereo camera setups that return a range, or depth, estimate for a feature, a single camera only returns a bearing, or direction, to a feature. The camera model is not invertible since the projection process maps a three-dimensional world-space point to a two-dimensional image-space point. In the bearing-only case a feature initialization technique is required to track a feature over multiple images to estimate a depth value.

Supported in part by the Natural Sciences and Engineering Research Council of Canada and the Informatics Circle of Research Excellent

The authors are with the Department of Computing Science, University of Alberta, Edmonton, Canada T6G 2E8. E-mail: {jklipp,zhang}@cs.ualberta.ca

Although the amount of information in a single image–on the order of hundreds of thousands of pixels–is useful for discriminating between landmarks, it becomes necessary to reduce the dimensionality of the input to make the SLAM process computationally tractable. This reduction is known as feature extraction.

We consider visual SLAM performance with respect to different feature extractors at two stages in the SLAM process: directly after feature extraction before any SLAM processing (the "feature stage"), and after SLAM processing when the map and robot pose have been estimated (the "SLAM stage"). Measuring performance at the feature stage yields results that are not conditioned on the particular SLAM implementation, so fewer assumptions are made. However, performance of SLAM itself is not directly measurable a this stage, so a human-derived metric must be used as an approximation. Evaluation at the SLAM stage has the advantage of directly measuring the desired performance, but becomes dependent on the particular SLAM system.

Every visual SLAM implementation requires feature extraction, yet there are few comparisons of different techniques. Performance at the feature stage has been previously examined in [4] by applying the information retrieval metrics of recall and precision to the feature extraction and matching problem. Feature detectors are examined to see how well they can track points through a sequence of images in [5], and the separability of clusters representing feature points with different feature descriptors is measured in [6]. However, none of these approaches examines the actual performance of a SLAM algorithm. To overcome these restrictions, this paper presents two main contributions: a methodology for measuring SLAM performance, and an experimental study that examines three commonly used feature extractors.

We evaluate SLAM performance in terms of the estimated SLAM state. Typically, work describing the performance of SLAM is presented with a graphic of the final map, showing that the system "works." When ground truth is available, as in simulation, results are sometimes given in terms of final pose error or plots of error in individual variables with 2σ or 3σ error limits derived from the estimated covariance. If the average normalized error over multiple Monte Carlo runs stays within error bounds, it is judged to be consistent and offer an estimate compatible with ground truth. Consistency testing is described in general by Bar-Shalom and Fortmann [7] and applied to SLAM by Bailey *et al.* [8].

The methods mentioned above are only able to judge if a system provides correct estimates, not compare performance between different systems or configurations. To be able to produce a ranking of performance with different extractors,

we introduce the metric of accumulated uncertainty for quantifying SLAM performance.

In order to describe the SLAM performance evaluation methodology and explain the experimental study, we first discuss existing existing feature extraction and matching algorithms in Section II, followed by a description of our visual SLAM system in Section III. Performance evaluation metrics are discussed in Section IV. The methodology used to evaluate and rank feature extractors is presented in Section V, followed by experimental results in Section VI. Finally, conclusions are drawn in Section VII.

II. Feature Extraction and Matching

Feature extraction consists of a detector that finds points according to an interest metric, and a descriptor that describes the immediate area around an interest point. A matching algorithm compares descriptors between frames to find corresponding points.

A. Feature Detectors

The Harris-Stephens (or simply Harris) corner detector [9] is one of the most established and successful algorithms. For each pixel in an image, a matrix is formed that is related to the autocorrelation function. The matrix captures the principal curvatures of the image intensity, that is, how quickly the intensity changes in response to a small change in position. The eigenvalues of the matrix are proportional to the curvatures and are used to decide if a point is a corner, part of an edge, or a "flat" region of the image. A response function involving the trace and determinant of the matrix is used to avoid calculating the eigenvalues explicitly and local maxima points with response above a threshold are taken to be corners. The autocorrelation idea is also used in other detectors [10] but Harris has proven the most popular.

The theory of *scale-space* shows that in addition to a two-dimensional image position, a third dimension, scale, can be constructed using successive Gaussian convolution, which calculates the appearance of the image as if seen from further away [11]. It can be shown that points invariant to scale can be found by generating a pyramid of progressively Gaussian-smoothed images and then searching for extrema of the second derivative of the Gaussian convolution (or Laplacian of Gaussian). This is utilized by Lowe in the scale-invariant feature transform (SIFT) [11]. SIFT approximates the Laplacian of Gaussian function using a difference of Gaussians and extrema are found in the images formed by subtracting adjacent levels in the Gaussian pyramid. The method is engineered to be robust, invariant to scaling and rotation, and partially invariant to affine transformation.

Lucas and Kanade approach feature detection from a tracking perspective rather than using a human-derived metric for "interestingness" [12]. Features are chosen for their suitability for the method used to track features from frame to frame, so in some sense it is "optimal by construction." A Newton-Gauss style gradient-based search is used to track an image patch through consecutive frames, estimating a displacement that minimizes the sum of squared differences. Interestingly, this leads to a similar metric as the Harris detector, where a point is chosen if both eigenvalues of a matrix similar to the autocorrelation matrix are above a threshold. Shi and Tomasi extend the work to estimate an affine warp between the current and original image patches, and use dissimilarity to detect tracking failure (which could occur due to occlusion, a feature leaving the camera field of view, or tracking a depth discontinuity that appears as a point to the camera but is actually a virtual point) [13]. We refer to this extractor as the Kanade-Lucas tracker, or KLT.

While this is not an exhaustive list of all feature detectors, these three are commonly used for visual SLAM, and are thus the focus of this work. An overview of the state of the art with respect to feature detectors is given by Mikolajczyk *et al.* [10].

B. Feature Descriptors

Harris and KLT describe the local neighborhood of a point by directly storing the raw image intensity values from a small square window around the point. This has the advantage of simplicity of computation but is not invariant to lighting changes, rotation, or viewpoint changes which may warp the image in an affine or projective manner. To overcome the susceptibility to lighting change, the descriptor can be normalized by subtracting the mean and scaling the values to cover a certain range (for example, the maximum range of the data type used for representation).

The SIFT descriptor first computes gradient magnitude and orientation for every pixel in a small region around the point. The region is divided into 4×4 subregions, and an orientation histogram is formed for the subregion, with the contribution of each pixel orientation the histogram bins weighted by gradient magnitude. Scale and rotational invariance comes at the cost of additional computation and may not be usable in real-time systems.

While the descriptors described here can be applied to any of the feature detectors described above, in this paper we treat the detector and descriptor as a unit for simplicity. The Harris and KLT detectors are paired with raw image descriptors and the SIFT detector is paired with with the SIFT descriptor.

C. Feature Matching

Matching is performed using the nearest-neighbor with distance ratio (NNDR) method [11]. For each detected feature the first- and second-nearest SLAM map features in terms of descriptor Euclidean distance are found. Matches are only accepted if the ratio of distances to the first- and second-nearest neighbors is less than a threshold. The reasoning behind this test is that incorrect matches will tend to match equally poorly to multiple features, so the ratio will approach unity. Gating based on this ratio ensures that accepted matches are in some sense unique.

III. Visual SLAM System

This section outlines the visual SLAM system used for the experiments. The system requires a camera and a mobile

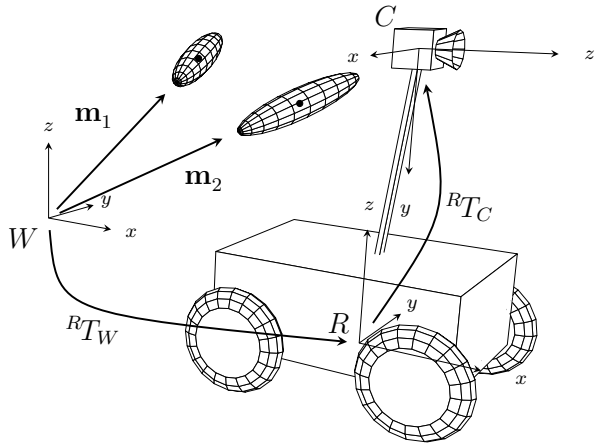

Fig. 1. Physical setup and references frames for the visual SLAM system. Robot and world frames $\mathcal{R}$ and $\mathcal{W}$ are related through the robot position $\mathbf{r}$ and corresponding transform ${}^{\mathcal{R}}\mathbf{T}_{\mathcal{W}}$. Landmarks $\mathbf{m}_i$ are represented as Gaussians in the world frame. Robot frame and camera frame $\mathcal{C}$ are allowed to differ by an arbitrary transform ${}^{\mathcal{R}}\mathbf{T}_{\mathcal{C}}$. In these experiments the camera is actually mounted pointing to the side of the robot.

robot capable of returning odometry data, and consists of: a SLAM filter based on the standard EKF-SLAM formulation [14], [15] and a visual front-end that processes images into useful SLAM observations. EKF-SLAM is the application of the Extended Kalman Filter (EKF) to the SLAM problem. The EKF, being an instantiation of the more general Bayes filter, estimates a state $\mathbf{x}$ in two steps: a prediction step that predicts a new estimate $\mathbf{x}_k^-$ from the previous estimate at time $k-1$ and a control vector $\mathbf{u}_k$ that represents the odometry input, and a correction step that generates a corrected estimate $\mathbf{x}_k^+$ using an observation $\mathbf{z}_k$ of the world. Since the filter is recursive and assumes the Markov property, $\mathbf{x}_k^+$ becomes the input for the next filter iteration.

We assume a robot moving in the two-dimensional plane, making observations of landmarks in a three-dimensional environment. The estimated state is

$$\mathbf{x} = \left(\mathbf{r}^T, \mathbf{m}_1^T, \mathbf{m}_2^T, \ldots, \mathbf{m}_N^T\right)^T \tag{1}$$

where $\mathbf{r} = (x, y, \theta)$ is the robot position and heading (pose) and $\mathbf{m}_i = (x_i, y_i, z_i)$ is the ith landmark location. One of the underlying assumptions of the EKF is that state variables can be represented as Gaussian random variables, which are fully characterized by a mean and covariance. The filter maintains the full covariance $\mathbf{\Sigma}$ of $\mathbf{x}$ which for SLAM is

$$\mathbf{\Sigma} = \begin{bmatrix} \mathbf{\Sigma}_r & \mathbf{\Sigma}_{rm} \\ \mathbf{\Sigma}_{rm}^T & \mathbf{\Sigma}_m \end{bmatrix} \tag{2}$$

where $\mathbf{\Sigma}_r$ is the robot pose covariance, $\mathbf{\Sigma}_m$ is the covariance of the map landmarks, and $\mathbf{\Sigma}_{rm}$ is the cross-covariance between the two. A graphical representation is shown in Fig. 1. Landmarks and robot pose are represented in the world reference frame $\mathcal{W}$ and the robot and camera reference frames $\mathcal{R}$ and $\mathcal{C}$ may differ by an arbitrary transformation.

At the start of each iteration of estimation, the predicted state estimate $\mathbf{x}_k^-$ is generated from the previous state $\mathbf{x}_{k-1}^+$ and control input $\mathbf{u}_k$ by a motion model $\mathbf{f}$, where the control input is assumed to be a Gaussian random variable with covariance $\mathbf{Q}$. We use an odometry-based motion model that predicts forward the current robot pose based on odometry measurements from the robot wheels. The motion model updates the estimated mean deterministically and the covariance is updated by linearizing the model and using the resulting Jacobian matrices to transform the previous covariance $\mathbf{\Sigma}_{k-1}^+$ and and add the control covariance $\mathbf{Q}$.

$$\begin{aligned} \mathbf{x}_k^- &= \mathbf{f}\left(\mathbf{x}_{k-1}^+, \mathbf{u}_k\right) \\ \mathbf{\Sigma}_k^- &= \mathbf{F}_x \mathbf{\Sigma}_{k-1}^+ \mathbf{F}_x^T + \mathbf{F}_u \mathbf{Q} \mathbf{F}_u^T \end{aligned} \tag{3}$$

where $\mathbf{F}_x = \partial\mathbf{f}/\partial\mathbf{x}|_{\mathbf{x}_k^-}$ and $\mathbf{F}_u = \partial\mathbf{f}/\partial\mathbf{u}|_{\mathbf{x}_k^-}$ are the relevant Jacobians. SLAM assumes a static environment, so landmark positions are not modified by the motion model update.

After the prediction step, features are extracted from an image captured from the camera and matched to the features representing map landmarks. Once an extracted feature with image position $\mathbf{z}_k$ had been matched to a map landmark, the projection $\hat{\mathbf{z}}_k$ of the map landmark in the image frame is calculated using the observation model $\mathbf{h}$. We use the standard pinhole camera model accounting for second-order radial distortion [16]. The innovation $\boldsymbol{\nu}_k$ is the error between projected and observed points in image-space, and has a corresponding covariance $\mathbf{S}_k$ that captures the uncertainty in robot pose, landmark position, and the observation model. The innovation covariance is calculated by linearizing the model and using the resulting Jacobian matrices to transform the predicted covariance $\mathbf{\Sigma}_k^-$ and add the covariance $\mathbf{R}$ representing observation model uncertainty.

$$\begin{aligned} \boldsymbol{\nu}_k &= \mathbf{z}_k - \hat{\mathbf{z}}_k = \mathbf{z}_k - \mathbf{h}(\mathbf{x}_k^-) \\ \mathbf{S}_k &= \mathbf{H}_x \mathbf{\Sigma}_k^- \mathbf{H}_x^T + \mathbf{H}_\pi \mathbf{R} \mathbf{H}_\pi^T \end{aligned} \tag{4}$$

where $\mathbf{H}_x = \partial\mathbf{h}/\partial\mathbf{x}|_{\mathbf{x}_k^-}$ and $\mathbf{H}_\pi = \partial\mathbf{h}/\partial\boldsymbol{\pi}|_{\mathbf{x}_k^-}$ are the relevant Jacobians, and $\boldsymbol{\pi}$ is a vector of observation model parameters. The innovation and innovation covariance are then used to correct the predicted SLAM estimate.

$$\begin{aligned} \mathbf{W} &= \mathbf{H}_x \mathbf{\Sigma}_k^- \mathbf{S}_k^{-1} \\ \mathbf{x}_k^+ &= \mathbf{x}_k^- + \mathbf{W}\boldsymbol{\nu}_k \\ \mathbf{\Sigma}_k^+ &= \mathbf{\Sigma}_k^- - \mathbf{W}\mathbf{S}_k\mathbf{W}^T \end{aligned} \tag{5}$$

The innovation covariance $\mathbf{S}_k$ is used to gate matches so measurements are only used if they are compatible with the current state estimate, reducing mismatches. Measurements are rejected if the innovation is larger than the 99% confidence limit defined by the innovation covariance matrix.

Implementation details of the system with bearing-only SLAM specifics and derivations are given in [17].

IV. Performance Evaluation

Ideally, SLAM performance would be assessed directly by comparing measured robot pose and landmark positions to the SLAM state vector estimate. Measuring a ground truth is difficult, especially in arbitrary environments. In our office-style environment it is possible to measure robot pose by tracking known points on the ceiling, but it is infeasible to

measure landmark positions. With these constraints in mind, evaluation is performed using only the robot pose.

Any filter, whether used for SLAM or not, should produce estimates that are *consistent* such that the estimates are compatible with ground truth. However, testing for consistency does not provide a ranking between different estimates, since estimates are either consistent or not. An additional metric must be found that provides a means of comparing the performance between estimates generated, in this case, using different feature extractors. In the methodology described later, consistency testing is used as a verification step before a comparative metric is applied to create a ranking.

A. Consistency Testing

Successful SLAM runs should be consistent in the sense that the "state errors should be zero-mean (unbiased) and compatible with the covariance yielded by the filter" [7, p. 71]. Consistency is tested by calculating the *normalized estimation error squared* (NEES)

$$\epsilon_k = \left(\mathbf{r}_k - \hat{\mathbf{r}}_k\right)^T \mathbf{\Sigma}_{r,k}^{-1} \left(\mathbf{r}_k - \hat{\mathbf{r}}_k\right) \tag{6}$$

using the SLAM estimated robot pose $\mathbf{r}_k$, ground truth robot pose $\hat{\mathbf{r}}_k$, and estimated pose covariance $\mathbf{\Sigma}_{r,k}$ [7], [8]. As stated above, the NEES should ideally be calculated with the entire SLAM state, however since ground truth landmark positions are unavailable, we only consider robot pose. With the hypothesis of a consistent filter with correct assumptions of Gaussianity and linearity, ϵ_k follows a χ^2 distribution with degrees of freedom equal to the dimensionality of $\mathbf{r}_k$.

A single trial does not yield enough information to determine if the system produces consistent results (a single run of an inconsistent filter may generate consistent results, and vice versa, depending on the environment) [8]. Instead, the average NEES value $\bar{\epsilon}_k$ is considered, calculated from N Monte Carlo runs of the filter

$$\bar{\epsilon}_k = \frac{1}{N} \sum_{i=1}^{N} \epsilon_k^{(i)} \tag{7}$$

where $\epsilon^{(i)}$ is the NEES for trial i. The robot pose predicted by the motion model and observation predicted by the observation model are randomly sampled according to the (Gaussian) robot pose covariance and innovation covariance, respectively. The χ^2 acceptance test from [7] is then used to check the hypothesis H_0 that the system errors are consistent with the estimated covariance. This hypothesis is accepted if the average NEES values lie within a confidence interval $\bar{\epsilon}_k \in [r_1, r_2]$, where the interval is calculated such that

$$P\left\{\bar{\epsilon}_k \in [r_1, r_2] | H_0\right\} = 1 - \alpha \tag{8}$$

where α is a small number such as 0.05 or 0.01, defining a 95% or 99% confidence interval, respectively. This defines a region for which NEES values are consistent. Values below the lower bound are conservatively inconsistent, meaning the estimated covariance is compatible with the estimation error, but could be made smaller and remain consistent. Above the upper bound, values become optimistically inconsistent, meaning that the error is outside a reasonable region defined by the estimated covariance.

B. Comparative Metrics and Accumulated Uncertainty

A useful metric for comparing performance should consider the state error and uncertainty and allow for a quantitative ranking of estimates obtained with the different feature extractors. This section discusses the shortcomings of some possible comparative metrics before describing a novel metric that satisfies the desired properties.

Accumulated error is an obvious metric for comparing performance, as minimizing the magnitude of state error is part of the goal of a SLAM system, and the result is easily comparable. However, it does not account for the uncertainty in state estimates and is thus not directly applicable to probabilistic systems. In this application for example, the case of small absolute error but near-zero uncertainty that is inconsistent with ground truth should not be ranked more favorably than the case of large absolute error and large uncertainty that is consistent with ground truth. Accumulated error is therefore not appropriate.

To account for the uncertainty, we could weight error by covariance and accumulate normalized error (the NEES). However, it is difficult to rank performance based on accumulated NEES. The best performance is not the lowest score, since a zero score is obtainable by setting the covariance to infinity at all timesteps, which is obviously not a desirable solution. Instead, the best performance would have consistent NEES values, but since NEES is classified in a binary manner (consistent or not) there is no concept of rank.

All things being equal, it is considered desirable to have less uncertainty, as the robot position becomes better known. Therefore, accumulating uncertainty rather than normalized error is chosen as a solution to the problem of choosing a metric. The best performance, ideally, is considered to be the system that remains consistent while obtaining the smallest uncertainty. The volume V of the ellipsoid that represents the estimated covariance matrix is considered to represent "uncertainty" at each timestep. V is found using the lengths of the principal axes r_i of the covariance, which are equivalent to the square roots of the eigenvalues λ_i, and is easily calculated using the determinant of $\mathbf{\Sigma}_r$

$$V(\mathbf{\Sigma}_r) = \frac{4}{3}\pi r_1 r_2 r_3 = \frac{4}{3}\pi\sqrt{\lambda_1\lambda_2\lambda_3} = \frac{4}{3}\pi\sqrt{\det(\mathbf{\Sigma}_r)}\ . \tag{9}$$

Accumulated uncertainty is simply the sum of volumes over time, given N_s steps in a trial

$$AU = \sum_{k=1}^{N_s} V(\mathbf{\Sigma}_{k,r})\ . \tag{10}$$

This statistic is calculated over the Monte Carlo runs, and the average of accumulated uncertainty

$$\overline{AU} = \frac{1}{N} \sum_{i=1}^{N} AU_i \tag{11}$$

is used to rank feature extractors, where the highest-ranked extractors will have the lowest accumulated uncertainty while

remaining consistent. It should be noted that the accumulated uncertainty values obtained in this manner with different feature extractors are only directly comparable on the same dataset. Comparison between extractors on different length datasets could be accomplished by normalizing the accumulated uncertainty with respect to path length.

V. Evaluation Methodology

Using the metrics defined in the previous section, we now present a procedure for performance evaluation of visual SLAM with different feature extractors. Images and odometry data are captured from a mobile robot with a single camera while it is driven through an environment. A ground truth is captured simultaneously using an independent vision system to be explained shortly. Data is captured for later offline processing so that multiple SLAM runs can be made on the same dataset. After data acquisition, multiple Monte Carlo runs of the SLAM system are generated by randomly sampling the robot pose and innovation covariances at the prediction and observation steps, respectively. Evaluation is then carried out in two steps: the first to test estimate consistency, and the second to rank the feature extractors. The average NEES values are calculated in order to test consistency. Feature extractors that cause inconsistent estimates should be rejected. However, since EKF-SLAM has an inherent problem with inconsistency [8], it may become necessary to relax this rule. After rejecting based on consistency testing, the accumulated uncertainty metric is applied and used to rank feature extractors from the lowest to the highest uncertainty, using the same trajectory.

VI. Experimental Study

A. Setup

An ActivMedia Pioneer P3-AT mobile robot was used to record data for offline processing. A Dragonfly IEEE-1394 camera from Point Grey Research captures images at a resolution of 640×480 pixels using Bayer encoding. Images are Bayer decoded by downsampling, resulting in a final image size of 320×240 pixels. Odometry data from the robot is recorded at the same time an image is captured. Taking advantage of the structured nature of the environment, ground truth is generated with a ceiling tracker that uses the regular grid of ceiling tiles to calculate robot pose, as detailed [17]. The hardware setup is shown in Fig. 2.

The dataset is generated by driving the robot on the second floor of the Computing Science Centre at the University of Alberta. This environment represents a typical institutional foyer setting, shown in Fig. 3. Data is recorded after every 40 mm translation or 4° rotation, whichever is encountered first. A C-shaped path is taken by the robot, as shown in Fig. 4. It is designed such that during the last leg of the path the robot re-observes the initial part of the environment. Although this is not loop closure in the sense of returning to the starting position, it is a type of closure since the initial features are visible. This is important as loop closure is a key test of a SLAM system, since it is only through re-observing previous features that pose covariance is reduced.

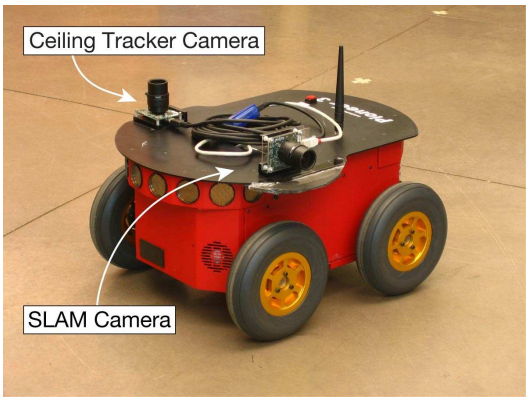

Fig. 2. ActivMedia Pioneer P3-AT with Point Grey Research Dragonfly IEEE-1394 cameras. One camera points to the left of the robot and is used for SLAM observations, while the other camera points upward and is used for tracking the ceiling to generate a ground truth robot path.

Fig. 3. Robot in experimental environment. The setting is an elevator foyer in the Computing Science Centre at the University of Alberta.

Additionally, this path configuration allows for the evaluation of feature extractors when a scale change occurs, as the robot observes the same space at varying depth.

The SLAM system described above is used to process the recorded data. The three feature extractors described earlier were used in the SLAM system: SIFT with a 128 element descriptor, Harris with an 11×11 image patch descriptor, and KLT with an 11×11 image patch descriptor.

Data from five trials was captured (Trials 1–5), where the trial number corresponds to the scale-change depth d in metres shown in Fig. 4. Fifty Monte Carlo runs were performed for every trial using each feature extractor, and the average NEES, uncertainty, and accumulated uncertainty at each time step were calculated. We show the results from Trials 1, 3, and 5 in Figs. 5, 6, and 7, respectively. Trials 2 and 4 exhibit the same trend and are described in [17].

B. Consistency Testing Results

Consistency is measured by examining the average NEES plots for each trial, as shown in Figs. 5–7. Given that the robot pose state is of size $n_x = 3$ and the number of runs in each trial is $N_r = 50$, the NEES consistency interval is $[2.36, 3.72]$ using a typical value of $\alpha = 0.05$. This interval is shown as dashed lines on the NEES plots.

Examining the three plots, it is seen that with all three feature extractors the robot pose average NEES remains within

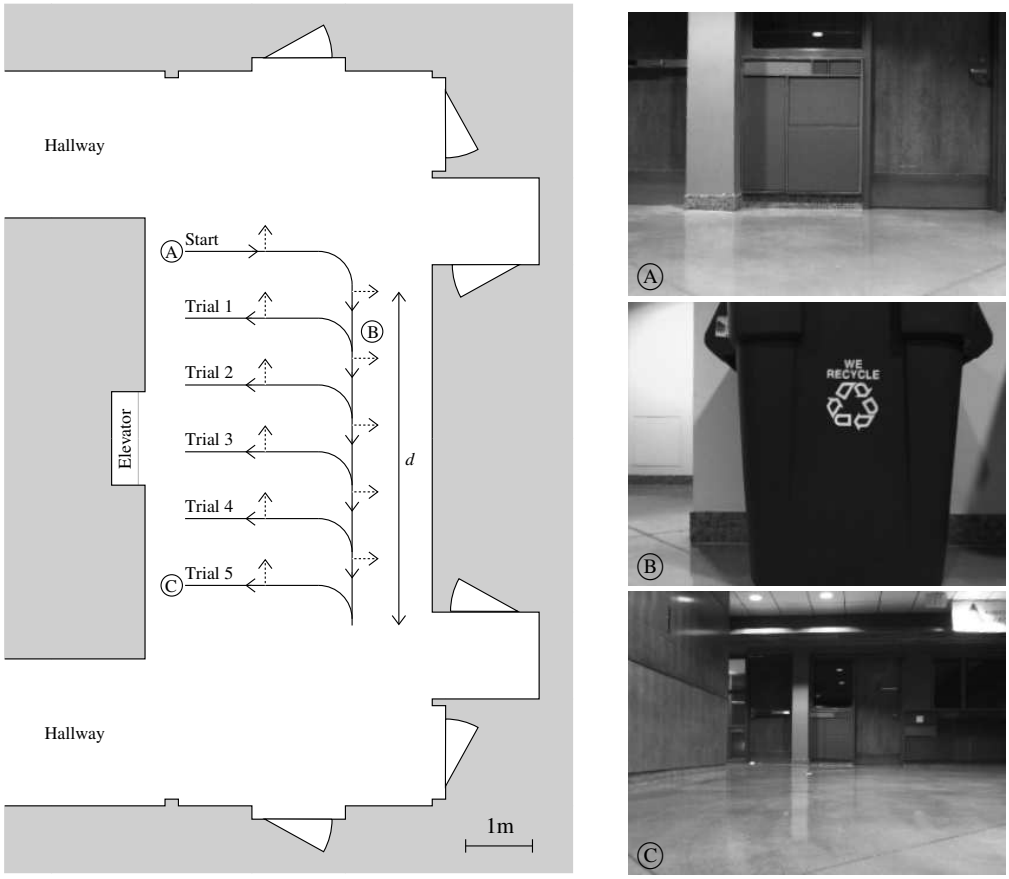

Fig. 4. Experimental environment consisting of an elevator foyer surrounded by laboratory rooms. Dimensions are approximate. Robot starts at point Ⓐ, drives forward, turns through a 90° arc to the right, drives downwards a distance d that varies from one to five metres, passing point Ⓑ. It then reverses through a 90° arc and drives in reverse to point Ⓒ. The front of the robot always faces to the right or down and the camera is mounted 90° counter-clockwise from the front of the robot, so it faces up or right. Camera direction is indicated by the dashed arrow, while direction of motion is indicated by the solid arrowheads. During the final section when reversing to point Ⓒ, the robot moves left while facing to the right.

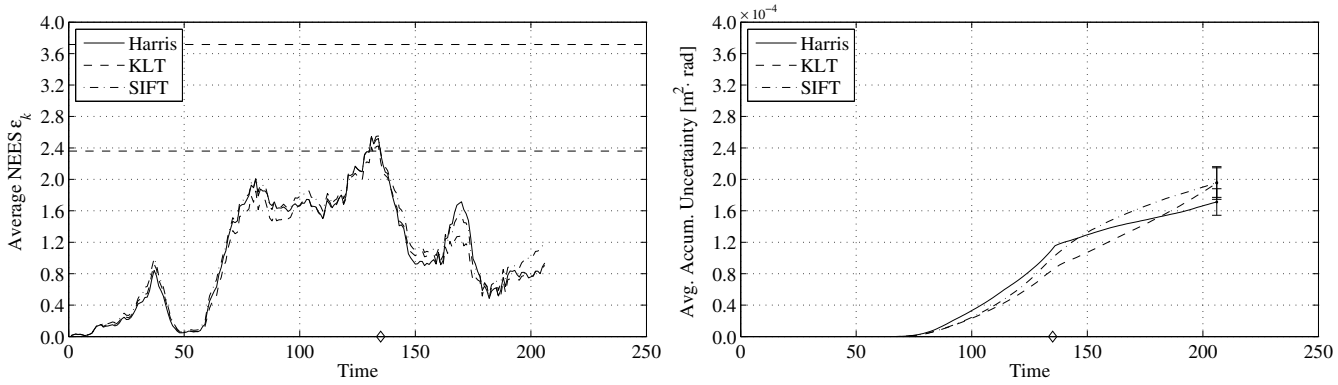

Fig. 5. Results from Trial 1. Left: Average NEES values using the three different feature extractors. Dashed lines indicate the 95% confidence bounds that demarcate the region in which the estimate is considered consistent, and the ⋄ on the time axis indicates when the robot was able to re-observe the initial section of the environment. Initially the system is conservatively inconsistent due to uncertainty added to account for the ground truth collection, and reaches the consistent region briefly before finishing conservatively inconsistent. Right: Average accumulated uncertainty with final 1σ standard deviation intervals. The standard deviation intervals overlap and little difference is seen between extractors. The change in slope around $t = 135$ corresponds to the robot re-observing the initial section of the environment.

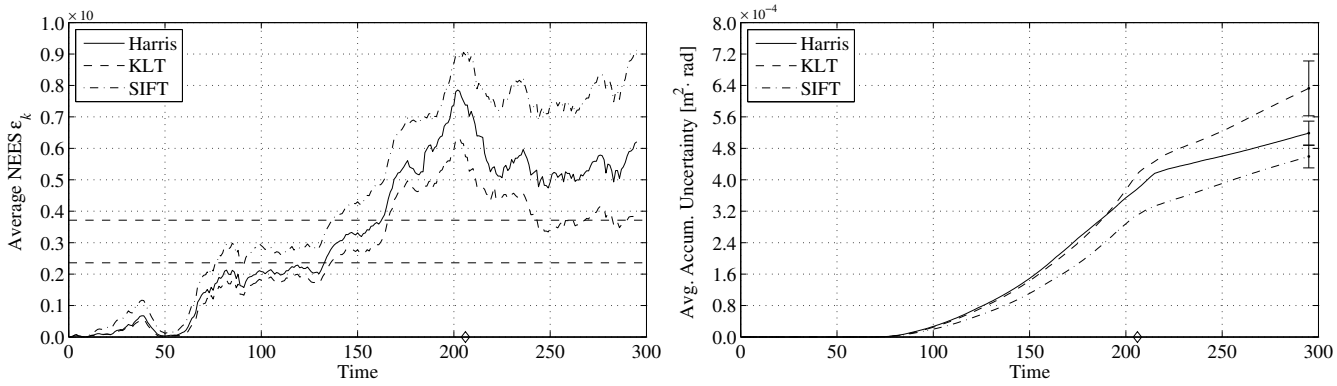

Fig. 6. Results from Trial 3. Left: Average NEES values using the three different feature extractors, with 95% confidence bounds and $\diamond$ marking the time of initial environment re-observation. Initially the system is conservatively inconsistent, then passes through the consistency region, with optimistic inconsistency beginning shortly after the first corner in the robot path between $t = 140$ and $t = 160$. Although the system recovers somewhat, it remains inconsistent at termination with all three extractors. Right: Average accumulated uncertainty with 1σ standard deviation intervals. The intervals either overlap or are very close, and little difference is seen between extractors. The change in slope around $t = 215$ corresponds to the robot re-observing the initial section of the environment.

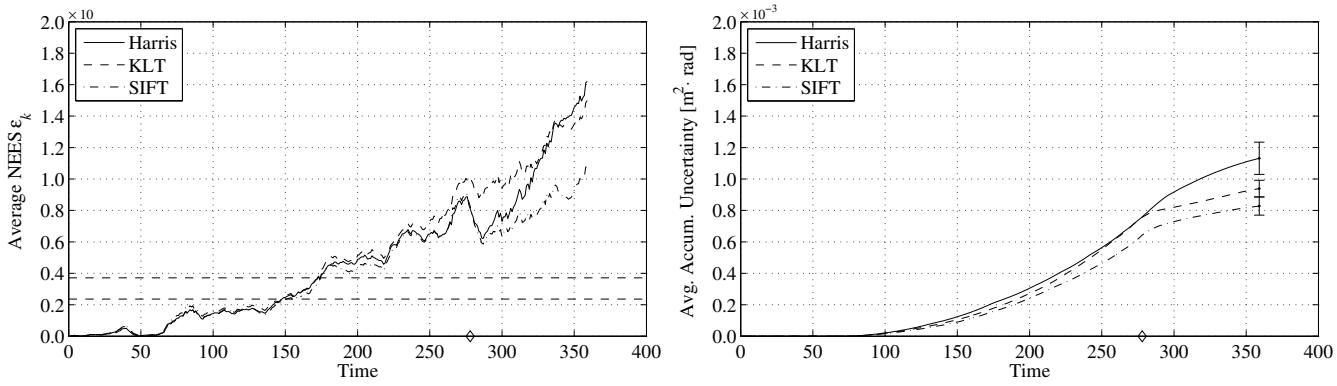

Fig. 7. Results from Trial 5. Left: Average NEES values using the three different feature extractors, with 95% confidence bounds and $\diamond$ marking the time of initial environment re-observation. Initially the system is conservatively inconsistent, then passes through the consistency region, with optimistic inconsistency again beginning shortly after the first corner in the robot path around $t = 160$. The system appears to diverge in all cases. Right: Average accumulated uncertainty with 1σ standard deviation intervals. The intervals either overlap or are very close, and little difference is seen between extractors. The change in slope around $t = 280$ corresponds to the robot re-observing the initial section of the environment.

or below the consistency region during Trial 1, rises above the consistency region to become optimistically inconsistent in Trial 3, and finishes well above the consistency region in Trial 5. Ideally, values should remain within the consistency bounds, although we will consider a conservatively inconsistent estimate acceptable in this paper since the error between robot pose and ground truth is still explained by the estimated covariance. All trials initially exhibit conservative inconsistency due in part to uncertainty added to the robot pose covariance to account for measurement error in the ground truth estimation. In all trials, there is little difference between performance with the three extractors, with the NEES in all cases following the same trend, so it seems there is no advantage to choosing a particular extractor in terms of consistency.

According to the methodology defined above, we should reject all the extractors in Trials 3 and 5. However, since all extractors are affected similarly by inconsistency, we will still compute and compare the accumulated uncertainty. Inconsistency in EKF-SLAM systems is a known problem that stems from, among other factors, the required linearization of non-linear models [8], [18] and is not easily avoidable. Finding useful solutions to EKF inconsistency is still an open problem. In light of this, it is reasonable that the longer Trials diverge further from the average NEES bounds.

C. Accumulated Uncertainty Results

It is seen from Figs. 5–7 that in many cases the average accumulated uncertainty with one extractor will lie within one standard deviation of another, and vice versa. For all other cases, the average accumulated uncertainty of all extractors lie within the same order of magnitude, often within a small multiple of the standard deviation of each other. While this does not imply statistical insignificance, it is a strong indication that SLAM performance does not vary much with different extractors.

If a ranking of performance with respect to extractor choice based on accumulated uncertainty is desired, SIFT

yields the lowest average accumulated uncertainty in four of five cases, although the standard deviation for SIFT often overlaps that of other extractors. In the four cases Harris is the second-lowest three times, with KLT second-lowest once.

In all cases, because average accumulated uncertainty results are not significantly different and the average NEES curves follow the same trend with different extractors we conclude that these results show that any of the three extractors may be used to perform visual SLAM in similar close-range indoor environments. This implies that other requirements such as speed or rotational invariance should be considered when choosing a feature extractor.

VII. Conclusion

This paper has presented two main contributions: a method for comparing the performance of a SLAM system using several different feature extractors, and an experimental study using this methodology with three extractors commonly used in visual SLAM. The methodology consists of two parts: testing consistency using the average NEES values to determine if estimates are compatible with ground truth, and creating a ranking of the feature extractors that result in consistent estimates using the novel average accumulated uncertainty metric.

From the results obtained, it was seen that over time the SLAM system becomes optimistically inconsistent. This agrees with the simulation carried out by others and is mostly a result of the EKF linearization step, rather than feature extractor choice. However, the average NEES curves using different extractors follow the same trend with only a small offset in magnitude, showing that the consistency of the system is largely independent of feature extractor choice. Although the inconsistency should cause the rejection of all the feature extractors, further analysis was performed since all feature extractors were affected equally without any advantage. No significant difference was found between different feature extractors in terms of accumulated uncertainty.

We speculate that a contributing factor was the gating performed on measured feature points before the observation step. Rejecting measurements that have large innovation with respect to innovation covariance greatly reduces the number of mismatches resulting from the feature matching step, which allows a poor feature extractor to perform as well as a good feature extractor, acting to equalize performance. It appears instead that the choice is "lost in the noise" among the multitude of parameters and choices involved in designing and calibrating the components of a visual SLAM system. As such, the system as a whole must be analyzed to determine optimal choices and settings to maximize performance.

These points lead to the final conclusion that the choice of feature extractor is not critical, so other criteria or constraints in a particular situation may dictate feature extractor choice. As with any experimental study, it was not possible to test every possible option and parameter set, and as such it is difficult to determine how well the results generalize to other situations. However, we believe similar systems operating in similar environments will yield the same results.

We conclude by suggesting that the metric of accumulated uncertainty could be an interesting tool for quantitatively comparing SLAM results. Future work may include using this metric to examine the effect of other parameter choices on SLAM performance.

References

[1] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I, the essential algorithms," *IEEE Robotics and Automation Magazine*, vol. 13, no. 2, pp. 99–110, June 2006.

[2] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Proc. of International Conference on Computer Vision*, Nice, France, October 2003.

[3] J. M. M. Montiel, J. Civera, and A. J. Davison, "Unified inverse depth parameterization for monocular SLAM," in *Proc. of Robotics: Science and Systems*, Philadelphia, USA, August 2006.

[4] J. Klippenstein and H. Zhang, "Quantitative evaluation of feature extractors for visual SLAM," in *Proc. of Fourth Canadian Conference on Computer and Robot Vision*, Montreal, Canada, May 2007.

[5] Ó. M. Mozos, A. Gil, M. Ballesta, and O. Reinoso, "Interest point detectors for visual SLAM," in *Proc. of the Conference of the Spanish Association for Artificial Intelligence*, Salamanca, Spain, November 2007.

[6] M. Ballesta, A. Gil, Ó. M. Mozos, and Ó. Reinoso, "Local descriptors for visual SLAM," in *Workshop on Robotics and Mathematics (RoboMat)*, Coimbra, Portugal, September 2007.

[7] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*, ser. Mathematics in Science and Engineering. Orlando, USA: Academic Press, 1988, vol. 179.

[8] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot, "Consistency of the EKF-SLAM algorithm," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, 2006.

[9] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. of Fourth Alvey Vision Conference*, Manchester, United Kingdom, 1988, pp. 147–151.

[10] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, "A comparison of affine region detectors," *International Journal of Computer Vision*, vol. 65, no. 1/2, pp. 43–72, 2005.

[11] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[12] S. Baker and I. Matthews, "Lucas-Kanade 20 years on: A unifying framework," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221–255, February 2004.

[13] J. Shi and C. Tomasi, "Good features to track," in *IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, 1994.

[14] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, June 2001.

[15] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Autonomous Robot Vehicles*, I. J. Cox and G. T. Wilfong, Eds. New York: Springer-Verlag, 1990, pp. 167–193.

[16] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[17] J. Klippenstein, "Performance evaluation of feature extractors for visual simultaneous localization and mapping," Master's thesis, Department of Computing Science, University of Alberta, 2008.

[18] S. J. Julier and J. K. Uhlmann, "A counter example to the theory of simultaneous localization and map building," in *Proc. of IEEE International Conference on Robotics and Automation*, Seoul, Korea, May 2001, pp. 4238–4234.