On Combining Visual SLAM and Visual Odometry

Brian Williams and Ian Reid

Abstract—Sequential monocular SLAM systems perform drift free tracking of the pose of a camera relative to a jointly estimated map of landmarks. To allow real-time operation in moderately sized environments, the map is kept quite sparse with usually only tens of landmarks visible in each frame. In contrast, visual odometry techniques track hundreds of visual features per frame. This leads to a very accurate estimate of the relative camera motion, but without a persistent map, the estimate tends to drift over time. We demonstrate a new monocular SLAM system which combines the benefits of these two techniques. In addition to maintaining a sparse map of landmarks in the world, our system finds as many inter-frame point matches as possible. These point matches provide observations of the same features in the image, it is classed as “visual SLAM”. Because of this their system cannot benefit from revisiting a location; even if an old feature is re-observed, it re-enters the map, we improve the accuracy of ego-motion estimation in monoSLAM, by the effect of noise cancellation from many measurements, and also by overcoming failure-modes of monoSLAM, such as when there are too few map-to-image matches to constrain the ego-motion.

I. INTRODUCTION

As a camera moves through its environment, the motion of image features can be used to determine the trajectory of the camera and the three dimensional structure of the scene. Though the boundary is somewhat arbitrary, generally speaking, if the algorithm for estimating the trajectory works by matching features between image frames, it is classed as preforming “visual odometry”, while if the matching is between a live map of the scene structure and the current image, it is classed as “visual SLAM”.

A significant advantage of the latter is that repeated observation of the same features ensures that the trajectory estimate does not drift over time. Furthermore, though in monocular visual SLAM the scale is arbitrary, once set it is fixed by the map. The price of this, however, is the cost of building and maintaining the map. Current visual SLAM systems based on the EKF, say, (such as [1]) are limited in the size of the map by the computational complexity of maintaining the coupled pose and scene covariance. This in turn limits the number of feature matches available at any instant to those map features which project to the current view. This may be only a few, and occasionally too few to fully constrain the pose.

In contrast, a visual odometry system based on two-frame estimates of instantaneous relative motion [2] can work in constant time, but will inevitably exhibit drift because of accumulation of small errors in the inter-frame motion estimates. Furthermore, from two frames, only the direction of the inter-frame translation can be recovered, not the magnitude. To overcome this difficulty, sets of three or more views are used and the features are triangulated to maintain a consistent scale across the sequence [2], [3]. However, there are also occasional singularities where the epipolar geometry does not fully constrain the motion (e.g. when the camera undergoes a pure rotation).

In this work we aim to retain the advantages of a visual SLAM system, but to incorporate the additional information available from visual odometry style measurements into the filter. In the system described herein, map-to-image matches constrain the scale, as in “standard” monoSLAM. However by taking advantage of the apparent image motion of many features, rather than simply a select few from the map, we improve the accuracy of ego-motion estimation in monoSLAM, both by the effect of noise cancellation from many measurements, and also by overcoming failure-modes of monoSLAM, such as when there are too few map-to-image matches to constrain the ego-motion.

While map-to-image correspondences provide constraints on the absolute position of the camera in the map, two-frame point matches only provide constraints, via the epipolar geometry, on the relative motion of the camera between the two image locations. We show in this paper that such constraints are naturally incorporated into a filter recast from a world-centric frame into a camera-centric frame. To that end we derive the appropriate formulation of robo-centric SLAM [4] for a visual sensor, and show that (as expected) this also yields a more consistent estimate of the filter’s uncertainty.

Recently, Civera et al. [5] have also presented a monocular SLAM system which uses the robocentric framework and a visual odometry style observations. The observation of all point features is handled by including them as temporary landmarks in a transient map. Once a landmark passes out of view of the camera, it is removed from the map and forgotten. Because of this their system cannot benefit from revisiting a location; even if an old feature is re-observed, it re-enters the map as a new feature. Though they report accurate motion estimates over long sequences (via comparison with GPS data), they do not show a return to the same location,
when drift would be apparent. World-centric mapping is impractical in a system with a transient map, so their system is the first to report the use of robocentric mapping for a visual sensor. Nevertheless they do not report experiments to verify the expected benefits of this framework in terms of filter consistency. Their system is currently unable to achieve real-time operation, requiring about one second to process each frame.

A summary of their objective in that work would be to produce a visual odometry system using the monoSLAM framework. Our aim, in contrast, is to produce a visual SLAM system with a persistent map, but which benefits from visual odometry measurements. To that end, we build on our previous work [6], which was in turn an extension of [1]. Point landmarks are initialised at corner features using the inverse depth parameterisation [7] and are converted when they can be well estimated as a 3D point. At each frame, the system attempts observations of the map landmarks in the image using active search [1] after warping the patch descriptor to match the predicted camera viewing angle. False matches amongst the landmark observations are rejected using the joint compatibility branch and bound algorithm [8]. Our key novel contribution is to show how, via the robo-centric framework, we can elegantly incorporate additional measurements from pairwise point matches, as and when possible, and to demonstrate the improved accuracy and consistency that results.

The remainder of the paper is structured as follows. We begin (Section II) by describing our implementation of robot-centric mapping for a single camera SLAM system. The choice of sensor necessitates some differences from the original derivation in Castellanos et al [4] though our derivation is very close to Civera’s [5]. We then describe how pairwise point matches are expediently utilised to constrain the inter-frame ego-motion (Section III), and then (Section IV) give results on both simulated datasets and real video.

II. ROBOCENTRIC MAPPING

It has been shown that the Extended Kalman Filter suffers from inconsistency due to linearisation errors [9]. After the angular uncertainty grows beyond just a few degrees the filter becomes overconfident and underestimates the uncertainty in the estimates it produces. Castellanos et al. [4] have proposed a more consistent SLAM algorithm called robocentric SLAM. In their approach, the state is represented in a frame relative to the current position of the robot. In this frame, the position of the nearby landmarks being observed have lower uncertainty and so the linearisations made are more valid.

More pertinent to our own application is the fact that, because the current pose is always aligned at the origin of the coordinate frame, the new pose is given exactly by the incremental inter-frame motion. This incremental motion is precisely what visual odometry measures. We show in the Section III how these measurements can be incorporated naturally and elegantly, but begin by adapting robocentric mapping [4] to the particularities of a monocular handheld camera.

A. Robocentric State Representation

In the robocentric framework, the state, \( x \), at timestep, \( k \), is parameterised as a multi-dimensional Gaussian represented in the coordinate frame centred on the pose of the camera, \( C \).

\[
x_k \sim \mathcal{N}(\hat{x}_k^C, P_k^C)
\]

where a superscript indicates the reference frame for the estimate.

In the reference frame of the camera, the camera pose is known with certainty and so is not included in the state vector. An entry is created to estimate each map landmark relative to the camera, \( \hat{x}_L^C \), the linear, \( \hat{x}_v^C \), and angular velocities, \( \hat{x}_\omega^C \), of the camera, and the origin and orientation of the world reference frame, \( \hat{x}_W^C \). This final entry allows the estimate to be transformed into the world representation if required.

B. Prediction and Update Steps

Like the world-centric approach, the first step in the robocentric EKF is to predict the motion of the camera since the last timestep. Rather than using this motion to recentre the coordinate frame immediately, the incremental motion is instead added to the state vector so that the estimated motion is improved by the update. This helps to reduce the uncertainty and so decrease linearisation error. Our visual odometry observations will greatly improve this motion estimate.

We use a constant velocity motion model to predict this incremental motion. This motion prediction is then placed in the state vector to give the augmented predicted state, \( \hat{x}_{k|k-1}^C \). At the same time, the covariance is updated to reflect this prediction.

\[
P_{k|k-1}^C = F_k P_{k-1}^C F_k^\top + G_k Q_k G_k^\top
\]

where

\[
F = \frac{\partial \hat{x}_{k|k-1}^C}{\partial \hat{x}_{k-1}^C} \quad \text{and} \quad G = \frac{\partial \hat{x}_{k|k-1}^C}{\partial n}
\]

\( n \) is the process noise and \( Q_k \) is covariance.

When using the constant velocity motion model in monocular SLAM, the incremental motion estimate will be correlated with the rest of the state after the prediction stage. This is in contrast to the case of a robot with odometry measurements presented in [4]. The correlations appear in our case because the uncertain velocity estimates in the state vector are used to predict the incremental motion.

The update step in Robocentric Mapping is the same as that of the ordinary EKF, and so is omitted for brevity.

C. Composition Step

The final stage of robocentric mapping is to transform the entire stochastic map so that the new camera pose estimate, \( \hat{x}_k^C \), is centred at the origin. This is done using the (now refined) incremental motion estimate. The incremental motion and its uncertainty are effectively transferred to the landmark estimates as the motion is marginalised out of the state.
The estimate for each part of the state is calculated through composition with the refined motion

\[
\hat{x}_k^C = \begin{bmatrix}
\otimes \hat{x}_k^{C_{k-1}} \oplus \hat{x}_k^{C_k} \\
\otimes \hat{x}_k^{C_{k-1}} \oplus \hat{x}_k^{C_k} \\
\otimes \hat{x}_k^{C_{k-1}} \oplus \hat{x}_k^{C_k} \\
\otimes \hat{x}_k^{C_{k-1}} \oplus \hat{x}_k^{C_k}
\end{bmatrix}
\]

(4)

where \(\otimes\) and \(\oplus\) are coordinate frame inversion and composition as defined in [4]. The covariance is then transformed using the Jacobian of this transformation, \(J_{C_{k-1} \rightarrow C_k}\).

\[
P^C_k = J_{C_{k-1} \rightarrow C_k} P^{g,k}_{k} J_{C_{k-1} \rightarrow C_k}^T.
\]

(5)

### III. VISUAL ODOMETRY

The update time of the EKF algorithm scales quadratically with the number of entries in the state vector. For this reason our system [6], in common with Davison’s [1], keeps only a sparse map of landmarks, with typically 10 – 20 of these visible at any one time. However this neglects the information available from the image motion of other features. Even without knowledge of the 3D back-projection of an image feature, any pair of matched point features constrains the relative camera motion via the epipolar geometry. Such features are particularly useful in the case that very few map features project into the current frame.

One approach might be to find many matches, solve for the Essential Matrix [10] that encodes the instantaneous epipolar geometry, and then decompose this to yield a translation and rotation. Indeed this is the approach that early visual odometry systems took. We do not take this approach for a number of reasons. First, this method yields only the direction, not magnitude of the translation. Additional non-linear projections would be required to map the result to the state-space of the filter. Second, there exist singularities in which the epipolar geometry is defined, but the decomposition of the essential matrix is underconstrained (such as for a pure rotation of the camera). Third, a minimum of 8 points are required to compute E, but we would like to use additional points expediently, and this may mean using fewer than 8 points on occasion. Finally, in order to fuse the decomposition with the filter estimate would require suitable derivations of the uncertainty in the estimates (tedious, but not impossible).

Instead, we proceed as follows. We begin by determining the predicted Essential Matrix \(\hat{E}\) using the predicted inter-frame motion:

\[
\hat{E} = \begin{bmatrix}
\hat{E}_{C_{k-1} \rightarrow C_k} \\
\hat{E}_{C_{k-1} \rightarrow C_k} \\
\hat{E}_{C_{k-1} \rightarrow C_k} \\
\hat{E}_{C_{k-1} \rightarrow C_k}
\end{bmatrix}
\]

(6)

where \([\cdot]_x\) represents the skew symmetric matrix form of the translation.

Each point in frame \(k - 1\) has a predicted epipolar line in frame \(k\)

\[
l = \hat{E}p_{k-1}
\]

(7)

If the prediction were correct, then \(p_k\), the correspondence for \(p_{k-1}\) would lie on this line, up to image noise displacement. In practice, of course, the prediction is wrong, and a
B. Example

The visual odometry update is illustrated in Fig. 2 using a simple situation for clarity. The camera begins at the origin looking down the z-axis at 200 points in the world. The camera is then moved backwards and to the right while rotating about the y-axis. Our method is used to correct an inaccurate prediction of the camera motion using the visual odometry measurements of these 200 features. When only visual odometry measurements are used, the estimate for the camera motion after the update matches the true epipolar geometry. However, the estimated motion is correct only up to scale since this is unobservable with only visual odometry measurements. By also including the observation of a single 3D map landmark, this scale is determined and the true motion is estimated correctly. This is our key innovation in this paper: combining these two observation types allows our system to accurately estimate the motion of the camera while retaining just a sparse map of landmarks to reduce computation time while preventing drift.

IV. RESULTS

To test the performance of the robo-centric monocular SLAM system with visual odometry we have run experiments on both real and simulated data. We first test the accuracy of the system by evaluating the performance in simulation. Simulations provide a good test since perfect ground truth is known, and enable us to verify the improved accuracy and consistency claims we make. However, a simulation cannot perfectly replicate realistic operating conditions so a further test of the estimation accuracy is performed using an aerial photo to provide ground truth of the camera position.

A. Estimation Quality in Simulation

The simulation consists of a 100 \times 20 metre courtyard that the camera moves around while facing the wall. The top down view of the map of landmarks and the camera trajectory is shown in Fig. 3. The simulation begins with the camera at the origin with a correct estimate of the initial linear and angular velocities in the state vector. The initial map also contains four known landmarks to fix the scale of the map created.

Twenty Monte Carlo runs were performed using this simulated trajectory. For each run, the monocular SLAM system automatically selected, initialised, and observed landmarks from the simulated environment. Observations of these landmarks were perturbed with random Gaussian noise with a standard deviation of 0.25 pixels. However, when testing each of the three algorithms on a particular run, the same noisy observations of landmarks were used. Correct data association for each observation was given to the SLAM system.

For visual odometry observations, the simulator randomly selects 200 features in the image plane to track the motion between each timestep. The depth of these features is initialised to be on wall of the courtyard plus a random offset of up to 2 metres. This offset is used to avoid all of the features lying on a single plane.

The results of these simulated runs can be seen in Fig. 3. Each of the three monocular SLAM techniques is able to track the true pose of the camera throughout the sequence with different degrees of accuracy. The largest part of the error in the estimate for all three techniques is due to scale drift. The perceived scale of the world begins to grow as the camera gets further from the initial known features. This is due to the differences between the assumed motion given by the constant velocity motion model and the true trajectory of the camera. Scale drift is also seen in monocular SLAM when working with real world data. The scale error is corrected when the camera comes around the loop and reobserves the initial features again, ‘closing the loop’.

With measurement noise and an imperfect motion model, errors in the estimate are inevitable. However, a good estimation algorithm should keep errors to a minimum and correctly estimate its uncertainty in the answer given. The
(a) **Before EKF Update:** The predicted motion is incorrect in both translation and rotation leading to an incorrect epipolar geometry prediction.

(b) **Updated Using Just Visual Odometry:** The estimated pose is corrected up to a projective ambiguity. With planar motion, the orientation and the direction of the translation is determined but the scale of the translation is not.

(c) **Updated Using a Landmark and Visual Odometry:** The single measurement of a 3D landmark ♦ removes the projective ambiguity allowing the pose determined.

Fig. 2. **EKF Update Process With Visual Odometry:** This figure illustrates the update process using our proposed visual odometry measurements. The predicted motion estimate (a) is updated using only the visual odometry observations (b) and these observations along with a single 3D map landmark observation (c). **Left Column:** The ideal image plane after motion showing the position of 200 visual odometry features (.), the true (∙) and estimated (+++) epipoles, and the true (⋯) and estimated (——) epipolar lines for two selected features (•). **Middle Column:** Perpendicular distance to the epipolar line for each of the visual odometry features given the estimated camera motion. **Right Column:** The 3D pose of camera (⊙) relative to the features (•). The camera starts at the origin and then translates and rotates about the y-axis to the true pose shown in black. The estimate for this pose (×) is in colour.

ideal uncertainty in the estimate is calculated by running the simulation with the same observations but with zero measurement noise. In Fig. 4, a translation and orientation component of the camera pose estimate are examined in detail showing than the robocentric framework provides both a better estimate and a more realistic estimate of the uncertainty.

The underestimation of the uncertainty when using the worldcentric approach is due to linearisation errors. These errors become significant when the orientation uncertainty grows above 2 degrees. Once this occurs, the uncertainty estimate becomes corrupted by linearisation errors and is lower that the ideal uncertainty. This result was also found by Bailey et al. [9]. The robocentric approach is able to maintain a better estimate of the uncertainty because in the reference frame of the camera, the angular uncertainty of observed landmarks is much lower.

An estimator is said to be consistent if its state estimation error is unbiased and the actual Mean Square Error matches the calculated covariances. The consistency of an estimation
algorithm can be investigated by examining the normalised estimation error squared (NEES), $\epsilon$.

$$
\epsilon = (x_k - \hat{x}_k)^\top (P_k)^{-1} (x_k - \hat{x}_k)
$$

If the filter is consistent and linear-Gaussian, $\epsilon$ is $\chi^2$ distributed with dimension equal to the size of $x_k$. Here we perform 20 monte-carlo runs and have calculated the NEES of the camera position estimate. The 95% acceptance region for the $\chi^2$ test is between 2.02 and 4.16. If the average NEES is outside these bounds, it shows the estimator is conservative if lower and optimistic if higher. The results are shown in Fig. 5. More detail on this standard consistency check can be found in [9] and [14].

The simulation also provides a way to test the benefit of visual odometry. The simulation was rerun with different numbers of visual odometry observations but identical noisy landmark observations. Fig. 6 shows that the estimation error generally decreases as the number of visual odometry observations per timestep is increased, as expected.

**B. Estimation Quality in the Real World**

To test the accuracy of our system on real world data, a sequence was recorded outdoors using a trajectory which can be aligned to an aerial photo. The handheld camera was pointed at a row of buildings while the experimenter walked down the white line painted in the road. This sequence was then used to test the benefit of the robocentric framework and visual odometry measurements. The same landmark observations at each frame were used in each test. The results are shown in Fig. 7. Alignment was performed manually using the trajectory and building facades visible near the start of the trajectory (on the right). This makes any scale drift during the sequence more apparent.

When the traditional worldcentric framework is used, the scale increases over the sequence and the trajectory and map begin to curve towards the top of the image. With the same observations, the robocentric approach gives a very similar final estimate, but with larger estimated uncertainty reflecting a more consistent estimate. When visual odometry measurements are used alongside landmark observations, the motion estimate is far more accurate which in turn leads to a more accurate map estimate. The trajectory can be aligned to match the true trajectory shown by the white line above the parked cars in the photo. We hypothesize that the main benefit here accrues in a few key frames in which only a few map features were observed, and which poorly constrained the motion in the absence of additional VO features.

In another experiment, the accuracy was tested by moving...
the trajectory estimated using both visual odometry and landmark observations closely matches the true trajectory when aligned with this aerial photo.

During all of our experiments here, we allowed the system (30 Hz) for small maps and small numbers of observations.

6ms, or about 20% of the usual per-frame budget.

V. CONCLUSION

We have presented a monocular SLAM system which provides a high quality estimate of the camera pose both in accuracy and consistency. The increase in accuracy is achieved through a novel method for including many more observations per frame without the need of increasing the size of the state estimated. As well as observing the map landmarks at each frame to prevent drift, our system also observes the inter-frame motion of every other corner feature in a visual odometry style method. These extra observations are used to constrain the estimate of the inter-frame motion of the camera leading to a less noisy pose estimate.

The consistency of the estimate is improved through the use of the robocentric mapping framework. We have adapted this technique for use with a handheld camera and have shown that it provides more consistent estimates in monocular SLAM than the traditional worldcentric EKF algorithm. The robocentric framework provides a natural method for handling the visual odometry observations since estimating the inter-frame camera motion at each frame is a key part of the robocentric approach.

REFERENCES