Reduced Dimensionality Extended Kalman Filter for SLAM

Dinesh Gamage
dinesh.gamage@monash.edu
Tom Drummond
tom.drummond@monash.edu

Monash University
Australia.

Abstract

Computational complexity of the Kalman filter grows at least quadratically with the number of dimensions in the filter. This is a particular problem for applications like monocular simultaneous localization and mapping (SLAM) where it is not possible to run a single filter on a large map with many thousands of landmarks.

This paper presents a method for dramatically reducing the computational complexity of the Kalman filters by reducing the dimensionality as information is acquired. We prove the validity of our method by applying it to monocular SLAM, where there is a large number of dimensions in the filter that are not subject to process noise (the landmark locations). This has the effect of reducing the cost of running a filter or allowing a single filter to process a much larger set of landmarks.

Our approach also has a role to play within modern efficient sparse matrix approaches to SLAM where local information is coalesced into keyframes using Kalman filters. It also has general applicability to filtered measurement of static quantities where there are large numbers of dimensions that are not subject to process noise.

1 Introduction

A simultaneous localization and mapping (SLAM) system continuously explores the environment to causally estimate the ego-motion of a robot and to map the environment. Many successful SLAM algorithms represent the stochastic nature of the robot motion and the measurement models, together with noisy sensor data in a probabilistic manner, tracking the joint posterior over the vehicle pose and the map.

The filtering approach for SLAM, maintains only the current camera pose with all landmarks of interest as the state [7][1][12][3]. This representation marginalizes previous camera poses, which intern connecting all state elements with each other making the covariance matrix dense, leading to a fully connected graph. With new observations the filter complexity grows at least quadratically, quickly making the problem intractable. This quadratic growth can be handled by dividing the whole problem into small tractable maps while separately maintaining the global consistency [9]. Developments like parallel tracking and mapping (PTAM) [15] and dense tracking and mapping (DTAM) [16] take this further by noticing the possibility of working with an active set of landmarks by separating tracking from mapping. In these systems the tracker works with a local set of landmarks while the map maintains global consistency through bundle adjustment. Full bundle adjustment in PTAM adjusts the pose with respect to all key frames. It exploits the sparseness inherent in the structure-from-motion problem to reduce the complexity. Without maintaining all landmark descriptors in this manner one could even use a more efficient sparse matrix system [19] [20][8],[13],[14] as the back end to build a globally consistent map. As efficient as sparse matrix methods are, they still have limitations and aren't used to process all frames of video as this would generate much denser graphs with high connectivity which would overwhelm the approaches. Hence they are restricted to using sparsely sampled keyframes.

On the other hand coalescing observations into independent local keyframes by building a graph of local nodes, and optimizing the resulting graph [9] effectively overcomes the problem with keyframes in order to maximize the amount of information available. Because the information is acquired locally, the problem is nearly linear and so a Kalman filter was used to combine the information from multiple frames to give rich keyframes that know something about the inverse depth of landmarks as well as their image location. Hence the Kalman filter still has a role to play.

In this paper we present a method to reduce the dimensionality of the extended Kalman filter (EKF) for SLAM by identifying dominant modes of the filter. The method proposed can be used in general to reduce the dimensionality of the EKF irrespectively of its application, without being limited to SLAM. We are particularly interested in reducing the dimensionality of each node of the SLAM system mentioned in the previous paragraph [9], as this graph optimization problem is nearly linear with the number of nodes and as it combines the information from multiple frames to give richer key frames. The most complex operation in such system is the Kalman filtering within each node, which imposes an upper limit on the maximum number of landmarks each node can handle. Our approach can reduce the dimensionality of Kalman filters used in each node, increasing the number of features a node can handle leading to a more accurate estimation [5]. As we suggest in the discussion such a dimensionality reduced graph even can be used in a multi-camera setup more efficiently than a key frame based sparse matrix method.

Here we consider the particular class of Kalman filters where the process noise is zero for a large number of dimensions. To handle large fixed sized data sets those which change dynamically, a reduced Kalman filter [11] is being used in weather forecasting. Localization and mapping differs from this, as the state vector size grows with time. Also in SLAM there is a large number of dimensions with zero process noise. If the process noise is zero, the filter's n dimensional state vector can be regarded as a point in a nD space, with a hyper ellipse representing its uncertainty, given by the covariance matrix. After several observations, the uncertainty in some directions in this space will reduce, making further information along the same directions obsolete. This implies that information has to be fused only along other directions where enough evidence isn't yet available. In other words, observations contain useful information only along directions we are lot uncertain about. Using these principal modes we reduce the dimensionality of the EKF and maintain a reduced state It should be noted that, our method is fundamentally *different* from *sub-mapping* approaches as we are removing only less uncertain directions of the filter, indirectly maintaining all features of the filter in contrast to removing inactive landmarks done in sub-mapping.

We re-parameterize the filter in terms of uncertainty and work with a reduced number of uncertain dimensions, maintaining a covariance matrix only in this reduced space (see Section 3). This re-parameterization takes place repeatedly whenever the filter has enough information about an degree of freedom (it has a sufficiently small singular value) that it can freeze that dimension and remove it from the filter (see Section 2.2). This process frees up computational capability which in turn we can use by adding more state variables (corresponding to the locations of additional landmarks) to the filter (see section 2.1). EKF filters for SLAM problems also typically represent the camera pose (and possibly velocity) in the filter. These variable are subject to process noise and hence are not suitable for dimension reduction because the process noise introduces uncertainty in those variables. In the section 3.1, we also show how these variables can be handled in our reduced dimensional framework. The section 4 gives a complexity analysis of the system and the section 5 provides experimental results. Then we conclude (see section 6) by explaining how our method will be used for multiple camera SLAM.

2 Reduced Dimensionality Kalman Filter

If the filter starts at time $t = t_0$ with an unknown state and continues gathering information until $t = t_1$, the mean state of the filter, μ_{t_1} can be frozen and future changes to the mean can be represented as offsets. This observation enables us to decompose the covariance matrix into certain and uncertain dimensions through singular value decomposition. Let the covariance of the EKF be Σ_{t_1} at time $t = t_1$. Then:

$$\Sigma_{t_1} = UDU^T \tag{1}$$

where U is the set of singular vectors of Σ_{t_1} and D is the diagonal matrix of singular values. The singular vectors represent de-correlated uncertainty directions with variances proportional to their singular values. We partition D into D_s the significant (large) set and D_i the insignificant (small) set. U is also partitioned into U_s vectors corresponding to D_s and U_i vectors corresponding to D_i. With this Σ_t can now be written in block form:

$$\Sigma_{t_1} = \left[\begin{array}{c|c} U_s & U_i \end{array}\right] \left[\begin{array}{c|c} D_s & 0 \\ \hline 0 & D_i \end{array}\right] \left[\begin{array}{c} U_s^T \\ \hline U_i^T \end{array}\right] \tag{2}$$

So the original covariance matrix becomes:

$$\Sigma_{t_1} = U_s D_s U_s^T + U_i D_i U_i^T \tag{3}$$

As the second term of the above expression is comparatively small the column space of U_s can be used as the reduced space ignoring small singular vectors. Let the significant vectors U_s extracted at $t = t_1$ be U_{st_1} and x'_t be the reduced state which is Gaussian distributed in the reduced space with a mean μ'_t at $t(\geq t_1)$. The reduced mean μ'_t relates to the original state mean μ_t as:

$$\mu_t = \mu_{t_1} + U_{st_1} \mu'_t \tag{4}$$

This new state x'_t represents the variations of the original state around the point μ_{t_1} along the directions of the column space of U_{st_1}. Initially we start with μ_{t_1} being a zero vector, indicating our knowledge about uncertainty is zero along corresponding singular vectors in U_{st_1}. Subsequent observations can be projected onto the derived reduced space to gather information about the new state x'_t. The projected covariance matrix Σ_{t_1} is obtained as:

$$\Sigma'_{t_1} = U_{st_1}^T \left(\Sigma_{t_1} - U_i D_i U_i^T\right) U_{st_1} \tag{5}$$

The dimensionality of x'_t can be kept quite small, compared to the original state x_t. For all time steps $t(\geq t_1)$, information can be collected to update x'_t by changing its mean and the covariance. This makes the reduced state time dependent. To obtain the prediction equation in the reduced space, the linearized EKF states at time steps t and $t-1$ can be decomposed according to the equation 4. As we are assuming zeros process noise, the process model becomes the identity. Substituting the decomposed states into the process equation yields:

$$\mu_{t_1} + U_{st_1}\mu'_t = \left(\mu_{t_1} + U_{st_1}\mu'_{t-1}\right) \tag{6}$$

making the predicted state same as the previous state:

$$\mu'_t = \mu'_{t-1} \tag{7}$$

Similarly, if the measurement is z_t with the model Jacobian H_t and measurement noise v_t, update equation, after substitution becomes:

$$z_t = H_t\left(\mu_{t_1} + U_{st_1}\mu'_t\right) + v_t \tag{8}$$

which can be modified as:

$$z_t - H_t\mu_{t_1} = H_t U_{st_1}\mu'_t + v_t \tag{9}$$

Here $z_t - H_t\mu_{t_1}$ becomes the modified observation and $H_t U_{st_1}$ the projected Jacobian. The dimensionality reduction described so far works only when the state is static. In a SLAM setup, the camera pose keeps changing requiring a slightly different approach.

2.1 Adding New Landmarks

The computational savings achieved by reducing the dimensionality of the filter can be spent by admitting more variables into the filter to be measured (thus increasing its dimensionality again). A new variable l, can be directly added to the reduced state to obtain an augmented state. This requires increasing the size of μ'_t by one and adding a row and a column to U_{st} and Σ'_t. The augmented state estimate $\hat{\mu}'_t$ and set of reduced vector space $\hat{U}_{st_1}$ now become:

$$\hat{\mu}'_t = \begin{pmatrix} \mu'_t \\ l \end{pmatrix} \qquad \hat{U}_{st_1} = \begin{pmatrix} U_{st_1} & 0 \\ 0 & I \end{pmatrix} \tag{10}$$

with the covariance:

$$\hat{\Sigma}'_t = \begin{pmatrix} \Sigma'_t & 0 \\ 0 & \sigma \end{pmatrix} \tag{11}$$

where σ is the initializing uncertainty of the new variable. It has to be noted that this augmentation increases the dimensionality of the original space as well by the same number of dimensions. So the froze mean μ_{t_1} has to be augmented with zeros to get a modified mean $\hat{\mu}_{t_1}$. The relationship between the state estimate in the original space and the reduced space becomes:

$$\hat{\mu}_{t|t} = \begin{pmatrix} \mu_{t|t} \\ l \end{pmatrix} = \begin{pmatrix} \mu_{t_1} \\ 0 \end{pmatrix} + \begin{pmatrix} U_{st_1} & 0 \\ 0 & I \end{pmatrix}\begin{pmatrix} \mu'_{t|t} \\ l \end{pmatrix} \tag{12}$$

Here we have used the “hat” to denote augmented vectors and matrices. For clarity we can simply drop off the “hat” and use these variables to replace their un-augmented counterparts.

2.2 Iterative State Reduction

With continuous augmentations the reduced state $x'_{k|k}$ also will start growing. To keep the dimensionality of $x'_{k|k}$ manageable, we decompose it continuously by distributing newly learned knowledge over the froze mean μ_{t_1} and select a new reduced basis. Let us decompose Σ'_t as shown in equation 23 at a time step $t = t_n (\geq t_1)$ and denote the new basis by U'_{st_n}. Let x''_t be the new state, Gaussian distributed with a mean μ''_t. Thus μ'_t for any $t(\geq t_n)$:

$$\mu'_t = \mu'_{t_n} + U'_{st_n}\mu''_t \tag{13}$$

By substituting this into the equation 22 we can have:

$$\mu_t = \mu_{t_1} + U_{st_1}\left(\mu'_{t_n} + U'_{st_n}\mu''_t\right) \tag{14}$$

Above equation can be simplified as:

$$\mu_t = \mu_{t_1} + U_{st_1}\mu'_{t_n} + U_{st_1}U'_{st_n}\mu''_t \tag{15}$$

The reduction in equation 15, distributes the information learnt up to t_n over μ_{t_1} to get an updated mean μ_{t_n} such that:

$$\mu_{t_n} = \mu_{t_1} + U_{st_1}\mu'_{t_n} \tag{16}$$

With this the original reduced basis gets rotated towards the new basis to yield a new reduced basis U_{st_n} as:

$$U_{st_n} = U_{st_1}U'_{st_n} \tag{17}$$

This distribution can be done on frame to frame basis after each update, which makes the reduced basis U_{st_n} and the freeze mean μ_{t_n} time dependent.

In practice the significant set of vectors will become less than 20, as all other singular values are very small or zero. Irrespective of the size of the original state, this makes it possible to keep the reduced space dimensionality limited to a some number around 15 excluding the camera state.

3 Dimensionality Reduction in SLAM

Camera state and new features have to be handled separately when reducing the dimensionality in SLAM. A state vector x, composed of the camera and all the map features can be written as:

$$x = \left(x_c^T, y_1^T, y_2^T, \ldots, y_n^T\right)^T \tag{18}$$

where a scene 3D point i is defined by the dimension 6 state vector y_i and the camera state by x_c.

3.1 Reduction With the Camera

To model the desired variation while retaining the ability of doing standard operations on the camera, here we perform reduction only upon landmark states. As the first step, landmarks observed up to time $t = t_1$ are decomposed, keeping the camera state intact by directly transferring it into the reduced state. Transfer stacks the camera on top of the reduced state, which initially represented only the variation of landmarks.

Let us represent the state at time t_1, which is composed of camera parameters $x_{c_{t_1}}$ and the landmarks y_{t_1} as a Gaussian:

$$\mu_{t_1} = \begin{pmatrix} \mu_{c_{t_1}} \\ \mu_{y_{t_1}} \end{pmatrix} \qquad \Sigma_{t_1} = \begin{pmatrix} \Sigma_{c_{t_1}} & \Sigma_{c_{t_1} y_{t_1}} \\ \Sigma_{y_{t_1} c_{t_1}} & \Sigma_{y_{t_1}} \end{pmatrix} \tag{19}$$

For dimensionality reduction, first the covariance matrix of landmarks $\Sigma_{y_{t_1}}$, can be decomposed according to the equation 3 and then significant dimensions $U_{sy_{t_1}}$ can be extracted as the reduced space. As described earlier we fix the current landmarks state estimate $\mu_{y_{t_1}}$. In addition to the steps described in the previous section, to directly transfer the camera, $x'_{y_{t_1}}$ into the reduced space we have to extend its dimensions. So the reduced vector space $U_{sy_{t_1}}$ has to be extended by adding a set of orthogonal axes. Let the augmented reduced state to be x'_{t_1} and the vector space to be U_{st_1}, then:

$$\mu'_{t_1} = \begin{pmatrix} \mu_{c_{t_1}} \\ \mu'_{y_{t_1}} \end{pmatrix} \qquad U_{st_1} = \begin{pmatrix} I & 0 \\ 0 & U_{sy_{t_1}} \end{pmatrix} \tag{20}$$

The fixed point also has to be augmented with a set of zeros as we are transferring camera parameters into the reduced space. The modified fixed point then becomes:

$$\mu_{t_1} = \begin{pmatrix} 0 \\ \mu_{y_{t_1}} \end{pmatrix} \tag{21}$$

The original state x_t relates to the reduced state at $t \geq t1$ through:

$$\mu_t = \begin{pmatrix} 0 \\ \mu_{y_{t_1}} \end{pmatrix} + \begin{pmatrix} I & 0 \\ 0 & U_{sy_{t_1}} \end{pmatrix} \begin{pmatrix} \mu_{c_t} \\ \mu'_{y_t} \end{pmatrix} \tag{22}$$

The covariance in the reduced space can be easily obtained by projecting the original covariance onto the reduced space U_{st_1}:

$$\Sigma'_{t_1} = U^T_{st_1} \left(P^W_k - \begin{pmatrix} 0 & 0 \\ 0 & U_{i_y} D_{i_y} U^T_{i_y} \end{pmatrix} \right) U_{st_1} \tag{23}$$

Camera dynamics can be incorporated by working on this space where we have augmented the full camera state instead of its variations.

3.2 Prediction in the Reduced Space

When landmarks are static, the prediction involves estimating the camera and its covariance with cross-covariance [4]. As we transfer the camera state directly in to the reduced state x'_t, camera parameters can be predicted in the usual manner while keeping x'_{y_t} unchanged to get the predicted state $x'_{t|(t-1)}$. The covariance block in the reduce covariance Σ'_t corresponding to the camera state has to be added with process noise to get the predicted covariance $\Sigma'_{t|(t-1)}$.

3.3 Measurements in the Reduced Space

State update is done by projecting the measurement model onto the reduced space. Measurement model Jacobian H_t, can be projected onto reduced space by a right multiplication. For clarity if we drop the subscript t and denote the Jacobian as H:

$$H_s = HU_{st_1} \tag{24}$$

The innovation covariance then can be written as:

$$S_s = H_s \Sigma_{t|(t-1)} H_s^T + I \tag{25}$$

Reduced Kalman gain becomes:

$$K_s = x'_{t|(t-1)} H_s^T S_s^{-1} \tag{26}$$

If the predicted measurement is h, the reduced state can be updated as:

$$x_{t|t} = x_{t|t-1} + K_s(z-h) \tag{27}$$

4 Complexity Analysis

When all landmarks are static, prediction of a SLAM system simply becomes predicting camera parameters according to a dynamic model. As we directly transferred the camera state into the reduced space, the prediction complexity remains the same. The covariance of the camera has to be added with process noise covariance, which has the same dimensionality as well. The covariance of landmarks stays intact as landmarks are static. The matrix block that represents the cross-covariance between landmarks and the camera state need to get multiplied by the camera Jacobian. Let the dimensionality of the camera and the set of static landmarks to be n_c and n_l respectively. Let n_d be the dimensionality of the reduced space. In the reduced space the complexity of this multiplication becomes $\mathcal{O}(n_c^2 n_d)$ compared to the $\mathcal{O}(n_c^2 n_l)$ complexity in the original space.

Updating the state requires, inverting the innovation covariance of the filter. Let n_L and n_D be the dimensionality of full states including the camera, in the original space and in the reduced space respectively. The complexity of updating with a single landmark becomes $\mathcal{O}(n_L^2)$ in the original space. To obtain the Kalman gain, the innovation covariance has to be inverted. The complexity of inverting a n dimensional matrix is $\mathcal{O}(n^3)$. But, with the sequential innovation Kalman filter formulation [10], if the number of current measurements are limited to some number n_m which is smaller than the state dimensionality, the overall complexity reduces to $\mathcal{O}(n_m n^2)$. In a SLAM system, the number of observable feature n_m at a given time is smaller than the total number of landmarks. This reduces the complexity of the inversion step from $\mathcal{O}(n^3)$ to $\mathcal{O}(n_m n^2)$. When information is projected onto the reduced space, that information get distributed over all reduced states making a sequential approach impossible. The complexity has to remain cubic in the reduced space. But dimensionality of the reduces space is extremely small compared to the original space, where the complexity is $\mathcal{O}(n_D^3)$. Most importantly, the proposed method keeps n_D around 20 irrespectively of the original state dimensionality n_L.

5 Experimental Results

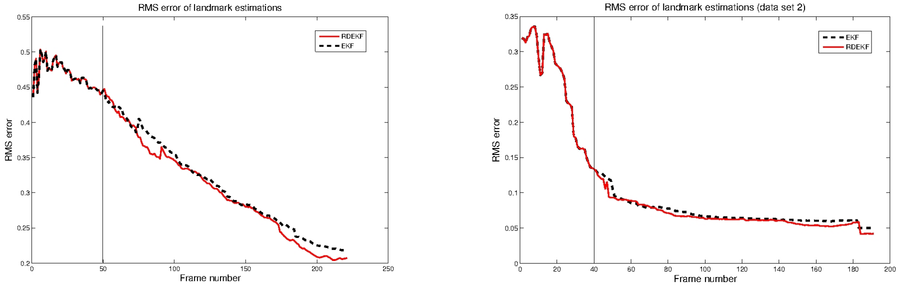
In order to prove the validity and the accuracy of the method proposed in this paper, first we used synthetically generated data. In the experimental setup we generated a set of $3D$ landmarks along with a synthetic camera sequence. All landmarks were inverse depth coded. At each time step t, we projected those $3D$ landmarks onto the corresponding camera plane C_t. Each projected feature was added with random Gaussian noise with a unit pixel standard deviation. To make the comparison a fair one, we used the same set of generated noisy data points to evaluate the accuracy of both scenarios. Those projected noisy feature points with their data associations in the previous image, were used to estimate camera parameters in a EKF setup. To evaluate the reduced dimensional EKF, we shifted from the standard filter to the reduced filter after learning the environment for some time. After each estimation step, inverse depth was extracted and normalized by the mean inverse depth to remove the scaling ambiguity. Synthetic ground truth data was also normalized in a similar manner.

When calculating the average inverse depth from estimated landmarks, to get a meaningful average depth, selected features need to be sufficiently certain. In other words, if there are newly initialized landmarks with highly uncertain depth estimates the average will become erroneous. If the estimated depth for a landmark is ρ such that $\rho > 0$ with a covariance Σ_ρ to avoid such situations, we set the selection criteria of the landmark to be:

$$\rho - \sqrt{\Sigma_\rho} > 0 \tag{28}$$

In our experiment we took the difference between the estimated inverse depth ρ_e and the ground truth ρ_g for each selected landmark after the normalization. Then we divided this difference by the inverse depth covariance of the estimated landmark in order to give a lower weight to landmarks with a higher uncertainty. To remove the error induced by the scale of the covariance (due to multiplication), result has to be multiplied by the average inverse depth again. Let μ_e and μ_g be the averages for estimated and ground truth inverse depths respectively. If n is the number of landmarks, the final RMS error for each frame can be obtained as:

$$E_{RMS} = \sqrt{\sum\left[\left(\frac{\rho_e}{\mu_e} - \frac{\rho_g}{\mu_g}\right)\frac{\mu_e}{\sqrt{\Sigma_\rho}}\right]^2 \div n} \tag{29}$$



(a) Dataset 1

(b) Dataset 2

Figure 1: RMS error of landmark estimations with conventional EKF and proposed reduced dimensionality EKF

Graphs in figure 1 show the RMS error of estimated inverse depth values against the frame number. The vertical line indicates the instance we started reducing the dimensionality of the system. Until environment is learnt sufficiently for reduction, we have to rely on the standard EKF. The dotted lined graph shows the error of the EKF based landmark estimation and the full lined graph shows the error of the reduced dimensional EKF (RDEKF) based landmark estimation. The graphs clearly indicate that the proposed RDEKF could yield similar or better results compared to the standard EKF. Other than the dimensionality reduction, selecting dominant modes bundles the movement of individual landmarks along the global trend of the system. The error reduction can be attributed to this out-lier tolerance we get through dimensionality reduction.

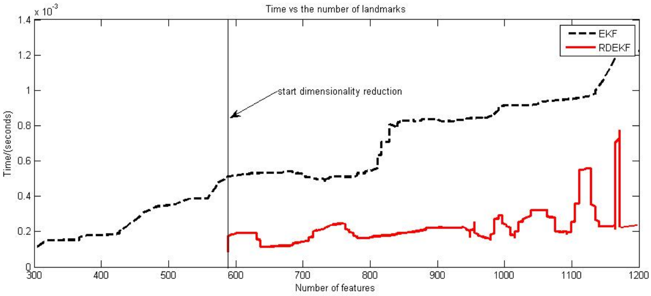


Figure 2: Time complexity of the update step for a real data sequence

.

The figure 2 shows the time taken by the update step of the Kalman filter for real data. Here we have considered only the update step as predictions are almost the same for both methods. In the figure, the values has been slightly median filtered to remove sudden spikes that occur in measuring the time complexity. The graph clearly shows that, we can keep the execution time of the RDEKF almost linear.

6 Discussion and Future Work

In this paper we introduced a dimensionality reduction technique to handle the complexity growth of the extended Kalman filter. Though the Kalman filter is not the state of the art any more with current sparse matrix methods, we believe still the Kalman filter has a considerable potential especially for collaborative SLAM.

The PTAM idea later has been extended to accommodate multiple trackers to map the environment through a sub-mapping approach [2]. It would be much efficient if it is possible to replace the back end of such a system with a new sparse matrix method. But such a back-end will impose some other limitations in a multi camera setup. When multiple cameras are updating the same sub-map, it is impossible to do a bundle adjustment within that sub-map in an incremental fashion, making methods like incremental smoothing and mapping (ISAM) [14] less appropriate as they cannot have multiple roots. The possible solution; merging maps together requires marginalizing all camera poses from each map to build a secondary map only with landmarks before fusing them together [6]. This marginalization will become tedious when the graphs are large.

We introduced our method aiming the representation of SLAM as a graph of coalesced observations [9]. Such a graph can be easily extended to accommodate multiple cameras, each coalescing its information into a selected node, where global consistency can be maitained separately. This method represents observations relative to an active node which has some similarities with relative bundle adjustment [17] where landmarks are represented relative to their observed coordinate system. But as camera is marginalized from each node when coalescing observations, multiple camera information can be fused independently into each node. Compared to existing methods which requires camera marginalization from a large bundle adjusted graph [6], fusing information into a node will be much more efficient. But fusing new information should be done only along uncertain directions. Our dimensionality reduction technique can be used to identify such directions, which intern increases the number of landmarks that can be handled by each node for improved accuracy [18].

References

[1] A. Azarbayejani and A.P. Pentland. Recursive estimation of motion, structure, and focal length. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(6): 562–575, 1995.

[2] Robert O Castle, Georg Klein, and David W Murray. Wide-area augmented reality using camera tracking and mapping in multiple regions. *Computer Vision and Image Understanding*, 115(6):854–867, 2011.

[3] A. Chiuso, P. Favaro, H. Jin, and S. Soatto. Structure from motion causally integrated over time. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(4): 523–535, 2002.

[4] J. Civera, A. Davison, and J. Martínez Montiel. 1-point ransac. *Structure from Motion using the Extended Kalman Filter*, pages 65–97, 2012.

[5] L.A. Clemente, A.J. Davison, I. Reid, J. Neira, and J.D. Tardós. Mapping large loops with a single hand-held camera. In *Robotics: Science and Systems*, 2007.

[6] Alexander Cunningham, Manohar Paluri, and Frank Dellaert. Ddf-sam: Fully distributed slam using constrained factor graphs. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 3025–3030. IEEE, 2010.

[7] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(6):1052–1067, 2007.

[8] Frank Dellaert and Michael Kaess. Square root sam: Simultaneous localization and mapping via square root information smoothing. *The International Journal of Robotics Research*, 25(12):1181–1203, 2006.

[9] E. Eade and T. Drummond. Monocular slam as a graph of coalesced observations. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[10] G. Evensen. The ensemble kalman filter: Theoretical formulation and practical implementation. *Ocean dynamics*, 53(4):343–367, 2003.

[11] Brian F Farrell and Petros J Ioannou. State estimation using a reduced-order kalman filter. *Journal of the Atmospheric Sciences*, 58(23):3666–3680, 2001.

[12] H. Jin, P. Favaro, and S. Soatto. A semi-direct approach to structure from motion. *The Visual Computer*, 19(6):377–394, 2003.

[13] Michael Kaess, Ananth Ranganathan, and Frank Dellaert. isam: Incremental smoothing and mapping. *Robotics, IEEE Transactions on*, 24(6):1365–1378, 2008.

[14] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John J Leonard, and Frank Dellaert. isam2: Incremental smoothing and mapping using the bayes tree. *The International Journal of Robotics Research*, 31(2):216–235, 2012.

[15] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234. IEEE, 2007.

[16] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2320–2327. IEEE, 2011.

[17] Gabe Sibley, Christopher Mei, Ian Reid, and Paul Newman. Vast-scale outdoor navigation using adaptive relative bundle adjustment. *The International Journal of Robotics Research*, 29(8):958–980, 2010.

[18] Hauke Strasdat, JMM Montiel, and Andrew J Davison. Real-time monocular slam: Why filter? In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2657–2664. IEEE, 2010.

[19] S. Thrun, Y. Liu, D. Koller, A.Y. Ng, Z. Ghahramani, and H. Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *The International Journal of Robotics Research*, 23(7-8):693–716, 2004.

[20] M.R. Walter, R.M. Eustice, and J.J. Leonard. Exactly sparse extended information filters for feature-based slam. *The International Journal of Robotics Research*, 26(4): 335–359, 2007.