On Keyframe Positioning for Pose Graphs Applied to Visual SLAM

Andru Putra Twinanda, Maxime Meilland, Désiré Sidibé, Andrew I. Comport

On Keyframe Positioning for Pose Graphs Applied to Visual SLAM

Andru Putra Twinanda[1,2], Maxime Meilland[2], Désiré Sidibé[1], and Andrew I. Comport[2]
[1] Laboratoire Le2i UMR 5158 CNRS, Université de Bourgogne, Le Creusot, France.
[2] CNRS-I3S, Université de Nice Sophia Antipolis, Sophia Antipolis, France.

***Abstract*—In this work, a new method is introduced for localization and keyframe identification to solve a Simultaneous Localization and Mapping (SLAM) problem. The proposed approach is based on a dense spherical acquisition system that synthesizes spherical intensity and depth images at arbitrary locations. The images are related by a graph of 6 degrees-of-freedom (DOF) poses which are estimated through spherical registration. A direct image-based method is provided to estimate pose by using both depth and color information simultaneously. A new keyframe identification method is proposed to build the map of the environment by using the covariance matrix between raletive 6 DOF poses, which is basically the uncertainty of the estimated pose. This new approach is shown to be more robust than an error-based keyframe identification method. Navigation using the maps built from our method also gives less trajectory error than using maps from other methods.**

***Index Terms*—SLAM, spherical system, keyframe identification, covariance matrix.**

I. Introduction

SIMULTANEOUS Localization and Mapping (SLAM) has been one of the most discussed research topics in the domain of autonomous robotics. In the general visual SLAM problem, the camera pose and environment structure are estimated simultaneously and incrementally using a combination of sensors. A visual SLAM approach is interesting in a wide range of robotics applications where a precise map of the environment does not exist.

In the last decade, many methods have been explored to perform robust full translation and rotation (6DOF) localization and mapping. In particular, some of the visual SLAM approaches [1], [2] have used feature-based techniques combined with depth and pose estimation. Unfortunately, these methods are still based on error-prone feature extraction techniques. Furthermore, it is necessary to match the features between images over time which is also another source of error since feature mapping sometimes is not necessarily one-to-one.

One can also refer to appearance and optical flow based techniques to avoid the feature-based problems, by directly minimizing the errors between image measurements. Methods that have similar approach like this fall into the category of image-based or direct methods. One of the earlier works [3] uses a planar homography model, so that perspective effects or non-planar objects are not considered. Recent work [4], [5] uses a stereo rig and a quadrifocal warping function which closes a non-linear iterative estimation loop directly with images. Visual odometry methods are however incremental and prone to small drifts, which when integrated over time become increasingly significant over a large distance.

A solution to reduce drift in visual odometry is to use image-based keyframe techniques such as in [6], [7], where each pose is estimated with respect to a reference image (keyframe) that has been acquired from learning phase. This is one of the solutions for mapping problem in SLAM, where the environment is represented by a set of connected image keyframes. This approach is also referred to as graph-based SLAM. Most of the work in this domain focused on the back-end which optimizes the obtained graph, such as the method in [8] that performs pose graph optimization by exploiting the sparseness of the Jacobian of the system. However, such methods do not investigate the importance of a keyframe, subsequently do not reduce the number of keyframes. Traditionally, the choice of keyframes is solely based on the travelled distance by the robot or the passing time in between keyframes. This is, however, not the best way to select, from an image sequence, the best images to build the structure of the environment. In the earlier work [9], a statistical approach to identify keyframes using a direct-method was proposed, which is based on the median absolute deviation (MAD) of the residuals. The drawback of this method is that it depends on a threshold value that does not apply for all types of sequences, so that different values are given for different kind of environment, making the map learning process totally empirical.

In the last few years, dense techniques have started to become popular. In particular, an early work [10] performing dense 6DOF SLAM over large distances was based on warping and minimizing the intensity difference using omnidirectional spherical sensors. Alternatively, other approaches have focused only on the geometry of the structure [11]. However, these techniques limit themselves either to photometric optimization only or to geometric information only. Dropping one or the other information means that there are important characteristics from the complete information that are being overlooked which might degrade in terms of robustness, efficiency and precision.

More recently, some techniques have considered to include both photometric and geometric information in the pose estimation process. In [12], a direct ICP technique was proposed which minimizes the error of both information simultaneously. Unfortunately, the approach is not well constrained in the technique because the minimization of the geometric error is only performed on the Z-component of the scene, not the whole 3D component. In this paper, it is argued that the

error minimization should incorporate all information provided from an omnidirectional spherical camera system, i.e. the photometric and depth (thus, 3D geometric) information, as also proposed in [10], [9]. By using all data, it is ensured that nothing will be overlooked while performing localization. The main contribution of this paper is to investigate a new keyframe identification method for graph-based SLAM that can be applied to general visual SLAM problem. However, in this case, a model of the environment is built by incrementally selecting a subset of the images from the learning sequence to be our reference spheres.

II. Spherical Tracking and Mapping

An environment will be represented as a graph containing nodes that correspond to robot poses and to all information obtained from those poses, as laid out in [10]. Every edge between two nodes corresponds to the spatial constraints between them. The 3D model of the environment is defined by a graph $\mathcal{G} = \{\mathcal{S}_1, .., \mathcal{S}_k; \mathbf{x}_1, .., \mathbf{x}_m\}$ where $\mathcal{S}_i$ are augmented spheres that are connected by a minimal parameterisation $\mathbf{x}_i$ which is the 6 degree of freedom (DOF) velocity twist between two spheres, expressed in exponentional map. For every sphere $\mathcal{S}$, it is defined by a set of $\{\mathcal{I}, \mathcal{Q}, \mathcal{Z}\}$ where

- $\mathcal{I} = \{i_1, \ldots, i_n\}$ is the spherical photometric image.
- $\mathcal{Z} = \{z_1, \ldots, z_n\}$ is the depth image.
- $\mathcal{Q} = \{\mathbf{q}_1, \ldots, \mathbf{q}_n\}$ is a set of equally spaced and uniformly sampled points on unit sphere where $\mathbf{q} \in S^2$ is expressed in spherical coordinate system (θ, ϕ, ρ) and belongs to a unit sphere $(\rho = 1)$

A. Localization

Robot motion can be represented by a transformation $\mathbf{T}(\mathbf{x})$ that takes the parameter $\mathbf{x}$ that consists of two vectors representing: translation velocity $\boldsymbol{v} = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}^T$ and rotation velocity $\boldsymbol{\omega} = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T$. The parameter $\mathbf{x} \in \mathbb{R}^6$ is defined by the Lie algebra as $\mathbf{x} = \int_0^1 (\boldsymbol{\omega}, \boldsymbol{v})\, dt \in \mathbb{SE}(3)$ which is the integral of a constant velocity twist which produces a transformation $\mathbf{T}$. The transformation and twist are related via the exponential map as $\mathbf{T}(\mathbf{x}) = e^{[\mathbf{x}]_\wedge}$, where the operator $[.]_\wedge$ is defined as follows:

$$[\mathbf{x}]_\wedge = \begin{bmatrix} [\boldsymbol{\omega}]_\times & \boldsymbol{v} \\ 0 & 0 \end{bmatrix} \tag{1}$$

where $[.]_\times$ represents the skew symmetric matrix operator.

For localization of a sphere $\mathcal{S}$, an initial guess $\hat{\mathbf{T}} = (\hat{\mathbf{R}}, \hat{\mathbf{t}}) \in \mathbb{SE}(3)$ of the current vehicle position with respect to a reference sphere $\mathcal{S}^* = \{\mathcal{I}^*, \mathcal{Q}^*, \mathcal{Z}^*\}$ is available, where $\hat{\mathbf{R}} \in \mathbb{SO}(3)$ is a rotation matrix and $\hat{\mathbf{t}} \in \mathbb{R}^3$ is a translational vector. Since it is assumed that the initial guess $\hat{\mathbf{T}}$ is available, the tracking problem boils down to the estimation of an incremental pose $\mathbf{T}(\mathbf{x})$ such that $\bar{\mathbf{T}} = \mathbf{T}(\mathbf{x})\hat{\mathbf{T}}$, where $\bar{\mathbf{T}}$ is the estimated pose of the current sphere.

The pose and the trajectory of the camera can be estimated by minimizing a non-linear least squares cost function[13]:

$$\mathcal{C}(\mathbf{x}) = \mathbf{e}_\mathcal{I}^\mathrm{T} \mathbf{e}_\mathcal{I} + \lambda_\mathrm{P}^2 \mathbf{e}_\mathrm{P}^\mathrm{T} \mathbf{e}_P \tag{2}$$

where, for every pair of spherical point and depth $\{\mathbf{q}_i^*, z_i^*\} \in \mathcal{S}^*$:

$$\mathbf{e}_\mathcal{I} = \begin{bmatrix} \mathcal{I}\left(w\left(\bar{\mathbf{T}}; \mathbf{q}_1^*, z_1^*\right)\right) - \mathcal{I}^*\left(\mathbf{q}_1^*\right) \\ \vdots \\ \mathcal{I}\left(w\left(\bar{\mathbf{T}}; \mathbf{q}_n^*, z_n^*\right)\right) - \mathcal{I}^*\left(\mathbf{q}_n^*\right) \end{bmatrix} \tag{3}$$

$$\mathbf{e}_P = \begin{bmatrix} \left(\bar{\mathbf{R}}\mathbf{n}_1^*\right)^T \left(\bar{\mathbf{P}}\left(w\left(\bar{\mathbf{T}}; \mathbf{q}_1^*, z_1^*\right)\right) - \bar{\mathbf{T}}\bar{\mathbf{P}}_1^*\right) \\ \vdots \\ \left(\bar{\mathbf{R}}\mathbf{n}_n^*\right)^T \left(\bar{\mathbf{P}}\left(w\left(\bar{\mathbf{T}}; \mathbf{q}_n^*, z_n^*\right)\right) - \bar{\mathbf{T}}\bar{\mathbf{P}}_n^*\right) \end{bmatrix} \tag{4}$$

where $\mathbf{e}_\mathcal{I}$ is a vector containing the intensity errors, $\mathbf{e}_P$ is a vector containing the structural errors, $\mathbf{P}_i$ is the i-th 3D point on the current sphere, $\bar{\mathbf{P}}_i^*$ is the homogeneous coordinate of $\mathbf{P}_i^*$ on the reference sphere, $\mathbf{n}_i^*$ is the surface normal at point $\mathbf{P}_i^*$, $\bar{\mathbf{R}}\mathbf{n}_i^*$ is the normal at point $\bar{\mathbf{T}}\bar{\mathbf{P}}_i^*$, and $w(.)$ represents the warping of a 3D point from a sphere to another, as shown in Figure 1.

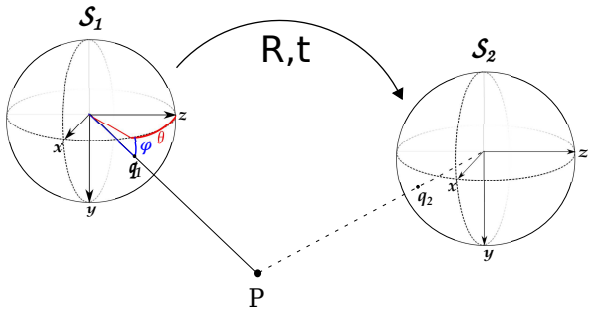


Figure 1. Illustration of spherical warping where the warping goes from $\mathcal{S}_1$ to $\mathcal{S}_2$ and P is a 3D point in the coordinate frame of $\mathcal{S}_1$.

The localization can now be considered as a minimization problem. The aim is to minimize simultaneously the cost function in an accurate, robust and efficient manner. Using an iterative approach, the estimate is updated at each step by a homogeneous transformation $\hat{\mathbf{T}} \leftarrow \mathbf{T}(\mathbf{x})\hat{\mathbf{T}}$. Using Gauss-Newton algorithm, the pose update $\mathbf{x}$ can be obtained iteratively from:

$$\mathbf{x} = -(\mathbf{J}^T\mathbf{J})^{-1}\mathbf{J}^T \begin{bmatrix} \mathbf{e}_\mathcal{I} \\ \lambda_\mathrm{P}\mathbf{e}_\mathrm{P} \end{bmatrix} \tag{5}$$

where $\mathbf{J}$ is the Jacobian of the cost function which is its derivative with respect to the 6DOF twist, and $(\mathbf{J}^T\mathbf{J})^{-1}\mathbf{J}^T$ is the pseudo-inverse of the Jacobian. The Jacobian can be expressed as $\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \mathbf{J}_{\mathrm{e}_\mathcal{I}} \\ \lambda_\mathrm{P}\mathbf{J}_{\mathrm{e}_\mathrm{P}} \end{bmatrix}$. By using chain rule, one can rewrite the Jacobian into more modular parts:

$$\mathbf{J}_{\mathrm{e}_\mathcal{I}} = \mathbf{J}_\mathcal{I}\mathbf{J}_w\mathbf{J}_{\mathbf{T}\mathbf{P}^*} \tag{6}$$

$$\mathbf{J}_{\mathrm{e}_\mathrm{P}} = \triangle\mathbf{P}^T\mathbf{J}_{\mathbf{R}\mathbf{n}^*} + \left(\bar{\mathbf{R}}\mathbf{n}^*\right)^T\left(\mathbf{J}_\mathbf{P}\mathbf{J}_w\mathbf{J}_{\mathbf{T}\mathbf{P}^*} - \mathbf{J}_{\mathbf{T}\mathbf{P}^*}\right) \tag{7}$$

where:

- $\mathbf{J}_\mathcal{I}$ is the intensity gradient with respect to its spherical coordinate position (θ, ϕ). It is of dimension $n \times 2n$.

In [10], an efficient way to compute $\mathbf{J}_{\mathcal{I}}$ using efficient second-order minimization is presented. The same technique is applied in this paper.

- $\mathbf{J}_w$ is the derivative of Cartesian to spherical conversion function. It is of dimension $2n \times 3n$.
- $\mathbf{J}_{\mathbf{T}_{\mathbf{P}^*}}$ is the derivative (velocity) of point transformation with a dimension $3n \times 6$.
- $\mathbf{J}_{\mathbf{P}}$ is the 3D point gradient with respect to its spherical coordinate position (θ, ϕ). It is of dimension $3n \times 2n$.
- $\triangle\mathbf{P}$ is the difference between the transformed points and the warped points $\left(\bar{\mathbf{P}}^w - \bar{\mathbf{T}}\bar{\mathbf{P}}^*\right)$
- $\mathbf{J}_{\mathbf{R}_{\mathbf{n}^*}}$ is the derivative with respect to the normal rotation. It is of dimension $3n \times 6$.

B. Robust Estimation

During the navigation, the environment can vary between the keyframe and the current images due to moving objects, illumination changes and occlusions. To deal with them, a robust M-estimator is used. The idea of M-estimator is to reduce the effect of outliers by replacing the residuals with another function of the residual. After applying the M-estimator to the residual, the pose update $\mathbf{x}$ can be obtained from:

$$\mathbf{x} = -(\mathbf{J}^T\mathbf{W}\mathbf{J})^{-1}\mathbf{J}^T\mathbf{W}\begin{bmatrix} \mathbf{e}_{\mathrm{I}} \\ \lambda_{\mathrm{P}}\mathbf{e}_{\mathrm{P}} \end{bmatrix} \tag{8}$$

where $\mathbf{W}$ is the weighting matrix where the diagonal corresponds to the weight computed by the weight function [14].

III. Keyframe Identification

In graph-based SLAM, selecting keyframes (i.e. reference spheres in this case) to be put as nodes in the final map is an important step. Taking too many references will cause the system to suffer from a high accumulated error because every time a new reference is taken, the residual error of the new reference will always be integrated in the following pose estimates, resulting in an accumulated drift. The error can be due to interpolations, occlusions, illumination change, and the dynamic of the environment (e.g moving cars). Yet needless to say, taking a new reference is also necessary to perform localization because the already mapped reference image goes out of view over large distances. Several strategies for keyframe selection will now be presented.

A. Median Absolute Deviation (MAD) [9]

One technique to achieve this goal locally is to observe the statistical dispersion of the residual error $\mathbf{e}$ obtained from the pose estimation process. The most common way to measure this is by computing the standard deviation (STD). However, the standard deviation is not a robust method because of its sensitivity to outliers. The MAD, on the other hand, is one of the simplest robust methods. It has a breakdown point of 50%, which means that the measurement still holds up close to 50% contamination of outliers, while STD has 0% breakdown point since a single large outlier can throw it off.

A new reference sphere is then placed according to the MAD of the weighted error:

$$\gamma < med(|\mathbf{W}\,\mathbf{e} - med(\mathbf{W}\,\mathbf{e})|) \tag{9}$$

where $med(.)$ is a function to extract the median of data and γ is the threshold for keyframe placement decision.

This approach is computationally cheap and optimized in many frameworks, resulting in a possibility to be applied for real-time applications. However, the criterion signifies that a new reference should be taken when the robust variance is too high, while 'too high' is an open statement. A drawback of this criterion is that we need to define a value to be the threshold. This process is totally empirical based on experiments and highly dependent on the characteristics of the sequence. Note that MAD can be applied to univariate data, hence the MAD is applied only on the intensity error since there isn't a good way to merge the two errors into the same scale and unit.

B. Incremental Ellipsoid

In the pose estimation process, one can compute the uncertainty of the estimation by using the covariance matrix. We propose a method that further observes the error ellipsoid. The orientation of the ellipse can be obtained by computing the eigenvector of the sub-covariance matrices. The orientation of the ellipsoid is, however, not used in the proposed criterion since the orientation of the error is invariant because it is based on the magnitude of the uncertainty. Instead, the semi axes $\mathbf{s} = \begin{bmatrix} s_x & s_y & s_z \end{bmatrix}^T$ of the error ellipsoid are more interesting to monitor since they are directly connected to the magnitude of the uncertainty. A new keyframe will be added to the map whenever:

$$\left\|\mathbf{s}_{t|t^*}\right\| > \left\|\mathbf{s}_{t|t-1}\right\| + \left\|\mathbf{s}_{t-1|t^*}\right\| \tag{10}$$

where $\mathbf{s}_{t|t^*}$ are the semi-axes resulting from warping the current sphere to the reference sphere, $\mathbf{s}_{t|t-1}$ are the semi-axes for warping the current sphere to the previous current sphere, $\mathbf{s}_{t-1|t^*}$ are the semi-axes for warping the previous current sphere to the reference sphere. The diagram of the comparison is shown in Figure 2.

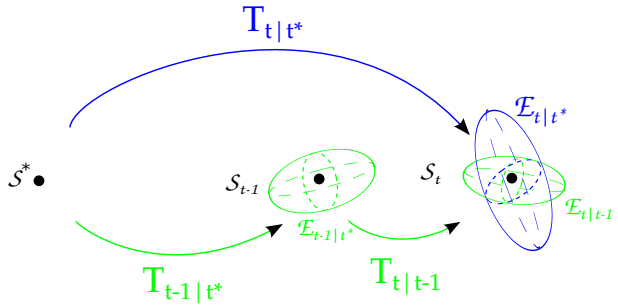


Figure 2. Illustration of incremental ellipsoid criterion

C. Symmetric Ellipsoid

The incremental error ellipsoid is, however, biased to the direction of computing the sequence. It is almost certain that if the direction of the exploration is inverted (i.e moving from

the end to the beginning of the sequence), the selected nodes will not be the same. This shows the method for selecting reference spheres is not based on the underlying information in the measurement, but depends on the computation order. If it is assumed that the complete sequence and its connectivity is already acquired (before the map learning is performed), a less biased method can be implemented. Instead of selecting the references incrementally, all the images in the sequence will initially be considered as references in the graph. In order to get the best nodes symmetrically, a symmetric comparison is added in the three-node groups. In this case, both forward and backward uncertainty is considered, as shown in Figure 3. The inequality in Equation 10 is now:

$$\|\mathbf{s}_{t|t^*}\|+\|\mathbf{s}_{t^*|t}\| > \|\mathbf{s}_{t|t-1}\|+\|\mathbf{s}_{t-1|t^*}\|+\|\mathbf{s}_{t^*|t-1}\|+\|\mathbf{s}_{t-1|t}\| \tag{11}$$

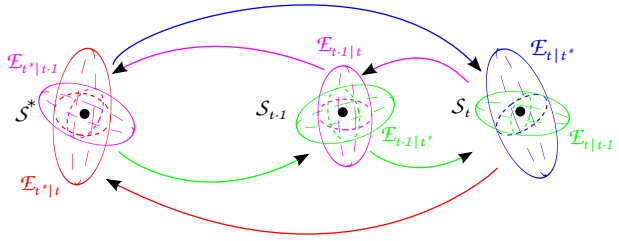


Figure 3. Illustration of symmetric ellipsoid criterion

IV. Experiments

A. Experimental Setup

To test the method, four synthetic sequences have been made. These sequences simulate indoor environment, however the system is designed to work in both outdoor and indoor environments. The detail of the sequences can be seen in Table I and some images are shown in Figure 4. The first two sequences are used to build the map and the last two are used during the map testing phase. We will compare the performance of our keyframe identification methods with the MAD criterion. In this experiment, two MAD thresholds γ are used: 8 and 12.

(a) (b)

Figure 4. Image with (a) spherical and (b) diffuse illumination

Our quantitative evaluation involves the number of references during the map building as well as the trajectory error with respect to the ground truth that can be computed from:

$$\triangle\mathbf{T} = \widetilde{\mathbf{T}}^{-1}\hat{\mathbf{T}} \tag{12}$$

where $\widetilde{\mathbf{T}}$ is the ground truth and $\hat{\mathbf{T}}$ is the estimated pose. The 6 DOF error between the estimated and the ground truth can be obtained by computing the logarithmic map of $\triangle\mathbf{T}$, such that $\triangle\mathbf{x} = log\,(\triangle\mathbf{T})$. The trajectory error $\triangle\mathbf{x}$ will be a 6-element vector that caintains the difference of translation velocity $\triangle\boldsymbol{v}$ and rotation velocity $\triangle\boldsymbol{\omega}$.

Table I
Sequence Data

Seq	#Images	Size	Illumination	Distance Traveled
1	1549	1024×512	Spherical	142 m
2	1549	1024×512	Diffuse	142 m
3	1400	512×256	Spherical	169 m
4	1400	512×256	Diffuse	169 m

B. Map Building Result

From Table II, it can be seen that there is a huge increase of number of references in the maps using MAD criteria on the sequence with spherical illumination (Sequence 1) compared to the sequence with diffuse illumination (Sequence 2). This is inevitable due to the higher intensity error introduced in the Sequence 1, meaning that the MAD threshold is easily reached after only a few images. The number of references using the incremental ellipsoid criterion, however, does not vary much with respect to the change in illumination: 32 and 30 for Sequence 1 and 2 respectively. In contrast, the number of keyframes in the maps using the MAD criteria varies with changes in lighting condition: 30 to 150 for MAD-8, and 19 to 77 for MAD-12.

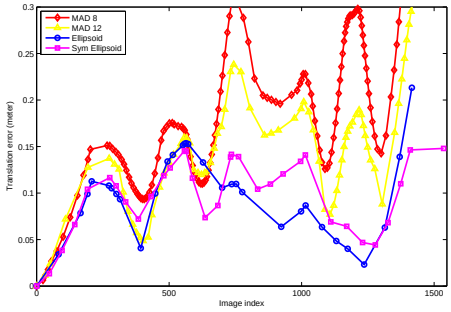
From this result, it can be seen that the ellipsoid criteria are better in terms of automatically choosing a consistent number of references for both types of sequences because it does not include a scalar threshold that has to be tuned before map learning process. In other words, the value 8 or 12 is not the best threshold value for the MAD criterion to select keyframes from Sequence 1. This verifies our argument that the MAD has a disadvantage due to its threshold that needs to be adjusted depending on the condition on the sequence, unlike the ellipsoid-based criteria that do not need any adjustments.

To observe the pose error, we can refer to Figure 5 that shows graphs of the chosen keyframes index against their pose error. If we look closely on the graphs, keyframes in the maps built by using the MAD are rather uniformly picked along the sequence. On the other hand, the ellipsoid criteria do not behave the same way and pick more keyframes at certain points along the sequence. These are the points where the robot is taking a turn and going to another corridor. By doing such turns, there will be a lot of new information introduced in the sequence and naturally it is favorable to take new keyframes when a lot of new information is introduced. The implication of this new information in the sequence is that higher error and higher uncertainty will be computed, resulting in more keyframes during the turns. However at some other points, the criteria pick less keyframes. This is the counter part of taking a turn which is going through a straight trajectory. Since we are working with a dense spherical system, going through such straight trajectory (in a corridor) does not introduce a lot of new information in the images. So, the criteria will only decide

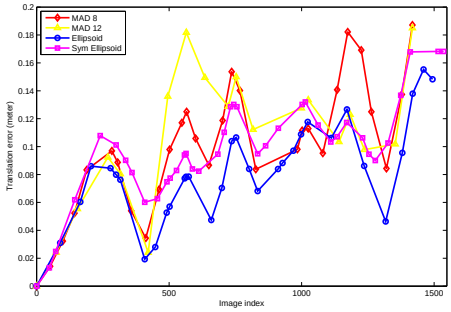
to take a new keyframe when the interpolation error starts to decrease the accuracy of the tracking system.

Table II
MAP BUILDING RESULT

Seq.	Criterion	#Ref	Avg. transl. err. (m)	Avg. rot. err.
1	MAD-8	150	0.1972	0.0183
	MAD-12	77	0.1484	0.0121
	Inc. ell.	32	0.0979	0.0065
	Sym. ell.	33	0.0982	0.0055
2	MAD-8	30	0.0999	0.0091
	MAD-12	19	0.1045	0.0086
	Inc. ell.	30	0.0814	0.0066
	Sym. ell.	34	0.0983	0.0061



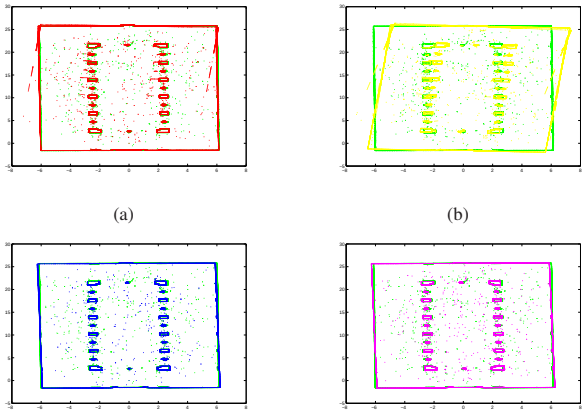
(a)



(b)

Figure 5. Keyframe's translational error for: (a) Sequence 1 and (b) Sequence 2. The rotational error is similar.

So far, we can conclude that in Sequence 1 the two ellipsoid criteria are superior compared to the MAD, in terms of number of references and pose error. Almost at every point in the maps obtained by ellipsoid criteria, the keyframes' pose error is less than the ones by the MAD. Although it is also the case for Sequence 2, we can not conclude yet whether the ellipsoid criteria are better than MAD criteria since the keyframes' pose error is not very different in the maps. However, we can see in Figure 6 that the reconstructed structures using ellipsoid criteria are slightly better, as the reconstructed structures of the second floor from MAD criteria are slightly slanted compared to the ground truth because the MAD criteria give more rotational error in the maps compared to the ellipsoid criteria. This can be the effect of reference placement choice by the criteria which has been mentioned previously, in which ellipsoid criteria select more keyframes on the turns than on straight trajectories. The reconstructed structures from Sequence 1 also have similar results, in which the structures reconstructed using MAD criterion are slanted compared to the ellipsoid criteria.



(a) (b) (c) (d)

Figure 6. Map quality on the second floor in the order of MAD-8, MAD-12, incremental, and symmetric ellipsoid ((a),(b),(c),(d)) on Sequence 2. The structure in green is the ground truth.

C. Map Testing Result

The first noticeable result from the trajectory error in Figure 7 is that there are a lot of spikes in the translational error graph. These spikes are caused by the changing of reference during navigation because the minimiziation process is still biased to the previous reference. This can be avoided by taking multiple keyframes simultaneously as references during navigation, as mentioned in [13], such that when a new reference is considered, change is not so radical since other references are kept during reference switching.

Referring to the trajectory error for environment with spherical illumination (Sequence 3) in Figure 7-a, it can be seen that tracking with the maps obtained by using MAD gives higher error. This drift is naturally caused by the reference pose error during the map learning. In addition to higher pose errors, other problems might appear in maps with high number of keyframes. Such maps make creating edges in the graph challenging, making it necessary to consider more sophisticated methods to build the connections between keyframes. With a high number of keyframes, they can be easily connected by false connections (false loop closures). These wrong connections can lead to wrong changes during navigation process, which will result in failure in tracking and higher trajectory errors.

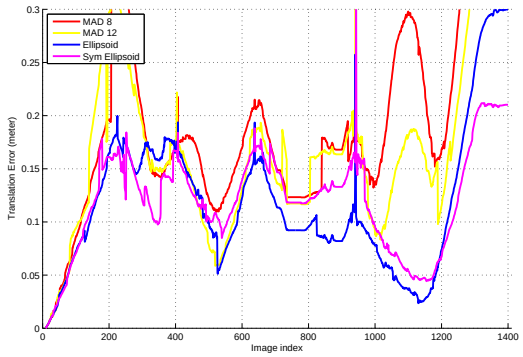
The incremental and symmetric ellipsoid methods seem to perform equally well, with slightly better performance from incremental ellipsoid, except at the end of the sequence. This might be the result of bias in direction. The incremental ellipsoid only considers one direction of the trajectory during learning which is the same direction as the testing sequence. So, the minimization scheme favors the incremental ellipsoid more than the symmetric ellipsoid.

If the case with diffuse illumination is considered, as shown in Figure 7-b, the performance of all four criteria pretty much

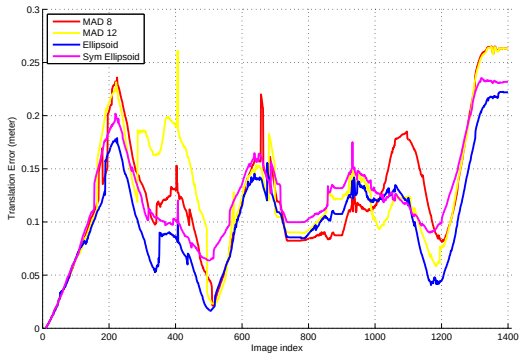
the same. Even so, at some points the ellipsoid criteria perform better than the MAD criteria and vice versa. This is highly related to the reference pose estimation error during the map building phase.

Table III
MAP TESTING RESULTS

Seq.	Criterion	Avg. transl. err. (m)	Avg. rot. err.
3	MAD-8	0.2026	0.0161
	MAD-12	0.1706	00116
	Inc. ell.	0.1193	0.0077
	Sym. ell.	0.1207	0.0065
4	MAD-8	0.1247	0.0102
	MAD-12	0.1292	0.0088
	Inc. ell.	0.102	0.0069
	Sym. ell.	0.1241	0.0065



(a)



(b)

Figure 7. Translational error during navigation test on: (a) Sequence 3 and (b) Sequence 4. The rotational error is similar.

V. CONCLUSIONS

A new spherical localization method was proposed that uses all photometric and geometric information for dense visual SLAM. A novel keyframe identification method (incremental ellipsoid) was proposed that incorporates the covariance matrix and compares the uncertainty ellipsoid between spheres. We have also extended it to work on symmetric navigation paths within the pose graph (symmetric ellipsoid) to ensure best selection of the keyframes. Although the MAD has the advantage of computational efficiency, it has been shown that the MAD has a drawback due to its scalar threshold value that needs to be adjusted accordingly to the characteristics of the sequence. On the other hand, the proposed methods don't need this adjustment and have better statistical properties, in terms of number of references as well as the quality of the maps. It has been shown that the method is more robust to variations in the lighting condition of the map.

There are still several aspects that remain to be explored within the proposed model. All the criteria presented in this paper are still biased to the first image in the sequence since it has to be included in the final map. By combining the symmetric ellipsoid criterion and loop-closure detection during keyframe identification, this bias can be eliminated since the first keyframe can be pruned during the map building phase. It has been mentioned beforehand that the work presented here is just improving the front-end of the graph-based SLAM. Some testing should also be done after applying it to the back-end. By doing this, the graph optimization method that will adjust the position of the nodes in the graph accordingly to its constraints. However, no pruning is needed since the selected nodes are already optimized in the mapping process.

REFERENCES

[1] A. Davison and D. Murray, "Simultaneous localization and map-building using active vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 865–880, 2002.
[2] A. Howard, "Real-time stereo visual odometry for autonomous ground vehicles," in *IROS*, 2008, pp. 3946–3952.
[3] S. Benhimane and E. M, "Real-time image-based tracking of planes using efficient second-order minimization," 2004.
[4] A. Comport, E. Malis, and P. Rives, "Accurate Quadri-focal Tracking for Robust 3D Visual Odometry," in *IEEE International Conference on Robotics and Automation, ICRA'07*, Rome, Italy, April 2007.
[5] ——, "Real-time quadrifocal visual odometry," *International Journal of Robotics Research, Special issue on Robot Vision*, vol. 29, no. 2-3, pp. 245–266, 2010.
[6] E. Menegatti, T. Maeda, and H. Ishiguro, "Image-based memory for robot navigation using properties of omnidirectional images," 2004.
[7] A. Remazeilles, F. Chaumette, and P. Gros, "Robot motion control from a visual memory," in *IEEE Int. Conf. on Robotics and Automation, ICRA'04*, vol. 4, New Orleans, Louisiana, April 2004, pp. 4695–4700.
[8] M. Kaess, A. Ranganathan, and F. Dellaert, "isam : Fast incremental smoothing and mapping with efficient data association," in *In IEEE International Conference on Robotics and Automation*, 2007, pp. 1670–1677.
[9] M. Meilland, A. I. Comport, and P. Rives, "Dense visual mapping of large scale environments for real-time localisation," in *IEEE International Conference on Intelligent Robots and Systems*, sept. 2011, pp. 4242 –4248.
[10] ——, "A spherical robot-centered representation for urban navigation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'10. Taipei, Taiwan*, Paris, France, 8-9 Novembre 2010.
[11] D. Cole and P. Newman, "Using laser range data for 3d slam in outdoor environments," in *ICRA*, 2006, pp. 1556–1563.
[12] T. Tykkälä, C. Audras, and A. Comport, "Direct Iterative Closest Point for Real-time Visual Odometry," in *The Second international Workshop on Computer Vision in Vehicle Technology: From Earth to Mars in conjunction with the International Conference on Computer Vision*, Barcelona, Spain, November 6-13 2011.
[13] M. Meilland and A. Comport, "Super-resolution 3D Tracking and Mapping," in *IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany., May 6-10 2013.
[14] P. Huber, "Robust estimation of a location parameter," *Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73–101, Mar. 1964.