Vision-based Mapping with Backward Correction*

Stephen Se
MD Robotics
9445 Airport Road
Brampton, Ontario L6S 4J3, Canada
sse@mdrobotics.ca

David Lowe, Jim Little
Department of Computer Science
University of British Columbia
Vancouver, B.C. V6T 1Z4, Canada
{lowe,little}@cs.ubc.ca

Abstract

We consider the problem of creating a consistent alignment of multiple 3D submaps containing distinctive visual landmarks in an unmodified environment. An efficient map alignment algorithm based on landmark specificity is proposed to align submaps. This is followed by a global minimization using the close-the-loop constraint. Landmark uncertainty is taken into account in the pair-wise alignment and the global minimization process. Experiments show that pair-wise alignment of submaps with backward correction produces a consistent global 3D map. Our vision-based mapping approach using sparse 3D data is different from other existing approaches which use dense 2D range data from laser or sonar rangefinders.

1 Introduction

We have proposed a vision-based localization and mapping algorithm [10] by tracking Scale-Invariant Feature Transform (SIFT) natural landmarks and building a 3D map simultaneously on our mobile robot equipped with Triclops, a trinocular stereo system.

However, our algorithm builds a 3D map continuously without maintaining the local image data, and hence does not allow backward correction. Therefore, it may have problems when large slippages or long-term drifts occur, or when the robot closes the loop, i.e., returns to a previously mapped area.

In this paper, we consider the map building problem as alignment of multiple submaps, to avoid drift accumulation. Moreover, we use the close-the-loop constraint to backward correct all the submap alignments and obtain a consistent global map. The novelty of this work is the 3D mapping using sparse distinctive visual landmarks in an unmodified environment with an efficient map alignment algorithm and constrained optimization.

2 Previous Approaches

The general approach of map building is to incrementally integrate new data to the map. When each new frame is obtained, it is aligned to a cumulative map [1]. The resulting map may become inconsistent as different parts of the map are updated independently. Since the data frames have been integrated but not maintained, inconsistency is difficult to resolve.

Smith *et al.* [11] developed an uncertain spatial relationship representation called the stochastic map, which contains estimates of the spatial relationships, their uncertainties and their inter-dependencies. The Kalman Filter is applied with the state vector consisting of the robot position as well as all the features in the map and the covariance matrix containing all the cross-covariances between the features.

This approach is similar to bundle adjustment [14] in photogrammetry and computer vision literature, which refines a visual reconstruction to produce jointly optimal structure and viewing parameters.

However, the computational complexity is $\mathbf{O}(n^2)$ where n is the number of features in the environment. Leonard and Feder [6] proposed decoupled stochastic mapping by representing the environment in terms of multiple globally-referenced submaps, as a computationally efficient approach to large-scale concurrent mapping and localization. A real-time implementation has been proposed in [4] using a compressed filter which can reduce the computation requirement when working in local areas.

Thrun *et al.* [12] proposed a probabilistic approach to concurrent mapping and localization using the Expectation-Maximization (EM) algorithm. They further developed a real-time algorithm combining the strengths of EM algorithms and incremental algorithms [13]. Their approach computes the full posterior over robot poses to determine the most likely pose, instead of just using the most recent laser scan as in incremental mapping. When closing cycles, backwards correction is computed from the difference between the incremental guess and the full posterior guess.

Lu and Milios [8] presented a 2D laser scan alignment algorithm which aligns all frames of sensor data to obtain a consistent map. Spatial relationships between local frames are obtained by matching pairwise laser scans and then a procedure based on the maximum likelihood criterion is applied to optimally com-

bine all the spatial relations.

Gutmann and Konolige [5] proposed a real-time method to reconstruct consistent global maps from dense laser range data. The techniques of scan matching, consistent pose estimation and map correlation are integrated for incrementally building maps, finding topological relations and closing loops.

3 Simultaneous Localization and Map Building

Our mobile robot localization and mapping system uses SIFT visual landmarks in unmodified environments. By keeping the SIFT landmarks in a database map, we track the landmarks over time and build a 3D map of the environment, and use these 3D landmarks for localization at the same time.

3.1 SIFT Stereo

SIFT was developed by Lowe [7] for image feature generation in object recognition. The features are invariant to image translation, scaling, rotation, and partially invariant to illumination changes and affine or 3D projection. These characteristics make them suitable landmarks for robust mapping, since when mobile robots are moving around in an environment, landmarks are observed over time, but from different angles, distances or under different illumination.

SIFT key locations are selected at maxima and minima of a difference of Gaussian function applied in scale space. Each SIFT feature is annotated with a subpixel image location and its scale and orientation.

We compute 3D world coordinates for each feature by matching pairs of images from the Triclops trinocular cameras. Using the epipolar and disparity constraints, we match features by scale and orientation in the right-left and right-top image pairs. The SIFT features and their 3D coordinates serve as landmarks for map building and tracking.

3.2 Map Building

To build a map, we need to know how the robot has moved between frames. Robot odometry can give a rough estimate and it is prone to errors such as slipping. It allows us to predict the region to search for each match more efficiently.

Once the SIFT features are matched, we can use the matches in a least-squares procedure to compute a more accurate camera ego-motion and hence better localization. This will also help adjust the 3D coordinates of the SIFT landmarks for map building.

We build a 3D map when the robot moves around in our lab environment and a Kalman Filter is used to track each landmark with a 3x3 covariance matrix [10]. The system currently runs at 2Hz on a Pentium III 700MHz processor.

4 Map Alignment

We would like to build submaps of the environment and then align them afterwards to obtain a global map.

We consider the alignment of two maps based on the specificity of SIFT features. The algorithm is also applicable to multi-robot collaboration. When multiple robots build up their own maps individually, we need to combine them together afterwards.

4.1 Local Image Characteristics

Sufficiently distinctive features are required to match scenes in the map. In order to obtain a feature vector of high specificity, the local image region is described in the SIFT feature in a manner invariant to various image transformations [7], in addition to the scale and orientation described in Section 3.1.

This feature vector is formed by measuring the local image gradients at a number of orientations in coordinates relative to the location, scale and orientation of the feature. The gradient locations are further blurred to reduce sensitivity to small local image deformations, such as result from 3D viewpoint change.

The local and multi-scale nature of the features makes them insensitive to noise, clutter and occlusion, while the detailed local image properties represented by the features makes them highly selective for matching to large databases of previously viewed features.

4.2 RANSAC

Given two sets of SIFT landmarks, we would like to find the coordinate frame translation and rotation that will result in the most matches between SIFT landmarks in the first map and SIFT landmarks in the second map. Since our robot is limited to planar motion, there are only 3 parameters (2 for translation and 1 for rotation) for this alignment.

This can be formulated as a hypothesis testing problem, where multiple alignment hypotheses are considered and the best one corresponds to the alignment which can match the most landmarks from the first map to the second map.

RANSAC [3] has been used in many applications for model fitting, hypothesis testing and outlier removal. We employ RANSAC to test the alignment hypotheses and find the inlier landmarks.

4.3 Algorithm

Firstly, we create a list of tentative matches from landmarks in the first database map to the landmarks in the second database map. For each landmark in the second database, we find the landmark in the first database which is closest in terms of the local image characteristics, and has similar height.

Then, we randomly select 2 tentative matches from the list, and compute the alignment parameters (X, Z, θ) from them. Two tentative matches are required in this case, since for each match, we can obtain 2 equations with 3 unknowns:

$$X = X_i - X_i' \cos \theta - Z_i' \sin \theta \qquad (1)$$
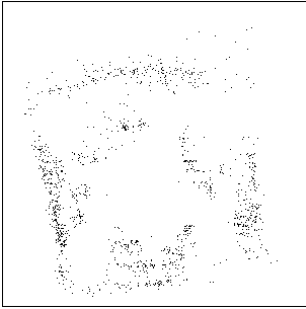$$Z = Z_i - Z_i' \cos \theta - X_i' \sin \theta \qquad (2)$$

Figure 1: *Map built without taking into account slippage occurrences.*

where (X_i, Y_i, Z_i) is the landmark position in the first database and (X'_i, Y'_i, Z'_i) is the landmark position in the second database. Equating two of these, we have:

$$A\cos\theta + B\sin\theta = C \qquad (3)$$
$$B\cos\theta - A\sin\theta = D \qquad (4)$$

where $A = X'_i - X'_j$, $B = Z'_i - Z'_j$, $C = X_i - X_j$, $D = Z_i - Z_j$. If the two tentative matches are correct, the distance between two landmarks is invariant for this Euclidean transformation, so the following constraint is applied to each sample selection: $A^2 + B^2 \approx C^2 + D^2$. This efficiently eliminates many samples containing wrong matches from further consideration.

Solving Equations 3 and 4, we obtain:

$$\theta = \tan^{-1} \frac{BC - AD}{AC + BD}$$

and substituting this into Equations 1 and 2 gives an alignment. We then check all the tentative matches which support this particular alignment (X, Z, θ).

This random selection, alignment computation and support seeking process is repeated m times. Assuming a contamination ratio of 0.70, to achieve 99% probability of a good sample, the required m is 50 [3].

The alignment with the most support is our hypothesis. We then proceed with least-squares minimization for the inliers which support this hypothesis, and obtain a better estimate for the alignment.

5   Building Submaps

The database map has been built using SIFT landmarks to correct odometry locally, which has been shown to be effective [10]. However, when there are large errors due to long-term drifts, the local correction is not sufficient. Therefore, we would like to detect the occurrence of drifts.

Figure 1 shows the map built without taking into account the drifts, where some parts of the map are skewed. Three rotational slippages of around 5 degrees clockwise each are intentionally added at 90, 180 and 270 degrees robot orientation.

When a drift occurs, the number of matches at the current position will be low but significantly higher at
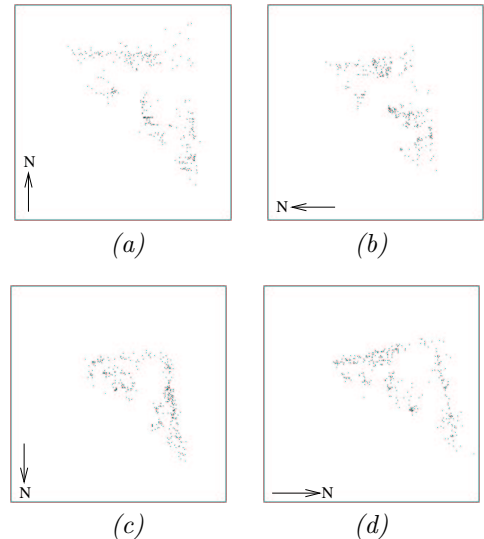

Figure 2: *Separate submaps are built due to slippage occurrences. (a) Submap 1. (b) Submap 2. (c) Submap 3. (d) Submap 4.*

a nearby position. This allows the detection of drift occurrences. To cater for the effect of drift in map building, we can estimate the actual robot pose based on the current frame, as in global localization [9], to correct the odometry for subsequent frames.

However, we employ an alternative method which starts building a new map whenever a drift is detected. Afterwards, all the submaps are aligned and combined to obtain a complete global map. This approach is more robust as the drift estimation is based on submap to submap alignment rather than frame to map alignment and hence more information can be utilized for the alignment process.

Using this method, we have obtained four submaps in this case, as shown in Figure 2, due to the three slippages. They are in different coordinates since the submap coordinates are the robot coordinates at the initial position for each submap.

This division can be done on a regular basis even if no drift is detected, by dividing into submaps every M frames, as described in Section 7.

5.1   Pair-wise Alignment

Using our map alignment algorithm, we can align two submaps together, provided there is some overlap. Since we terminate building the previous submap and then initiate building the current submap immediately, some overlapping landmarks do exist.

We employ a pair-wise alignment strategy, i.e., align each consecutive pair of submaps, and obtain the transformation from submap 1 to submap 2, from submap 2 to submap 3, and from submap 3 to submap 4. Figure 3(a) shows the pair-wise alignment results where the map is much better and unskewed. Submaps 1, 2, 3 and 4 occupy the top right, bottom right, bottom left and top left portions of the map respectively.
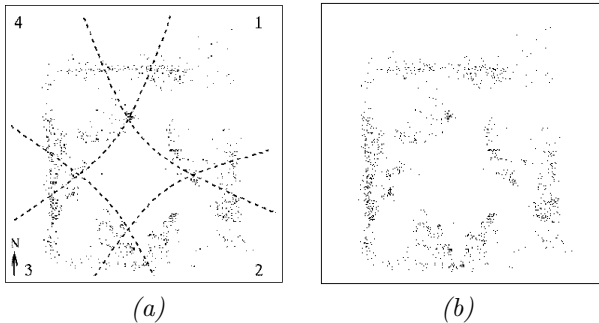
Figure 3: *(a) Pair-wise alignment map for submaps in Figure 2, with the submap composition indicated. (b) Incremental alignment map for submaps in Figure 2.*

5.2 Incremental Alignment

For incremental alignment, we align and combine submaps 1 and 2 to obtain a new map, and then align this new map with submap 3 to obtain another new map, and so on. Figure 3(b) shows the incremental alignment result and it looks very similar to the pair-wise alignment result.

In pair-wise alignment, the alignment of the current submap depends only on the single previous submap, but in incremental alignment, the alignment of the current submap depends on all the previous submaps covering that region.

When submap 4 is aligned in this case, its landmarks are matched with those in submap 3 as well as in submap 1, since we have rotated one revolution and back to the initial orientation again. On careful comparison of Figure 3(a) and (b), we can see that submap 4 has been pulled in a little bit towards submap 1 in Figure 3(b).

6 Closing the Loop

Pair-wise alignment and incremental alignment results are the same if each submap overlaps only with the previous submap, but different if we close the loop.

If we do not go back to a previously observed scene, we cannot obtain any correction. But when we do, we should spread out the correction throughout all alignments because errors have gathered over time, and not just attribute it to the last submap alignment.

We employ a global minimization strategy to do backward correction to all the submap alignments, when a close-the-loop is detected. Close-the-loop can be detected by checking if there is significant overlap of landmarks between the current submap and the initial submap, based on the SIFT specificity.

For submaps 1, 2, ..., n where submap n closes the loop, i.e., submap n goes back to the scene observed by submap 1 in the beginning, we firstly find the pair-wise alignment as before. We also do a pair-wise alignment between submap n and submap 1 too, and obtain n transformations in total. Let $\mathbf{T}_i$ denote the coordinates transformation for aligning submap i to submap i+1, or submap n to submap 1 when i equals n.

For a perfect alignment, we have the following constraint:

$$\mathbf{T}_1 \mathbf{T}_2 \dots \mathbf{T}_{n-1} \mathbf{T}_n = \mathbf{I} \tag{5}$$

where $\mathbf{I}$ is a 3x3 identity matrix.

During the pair-wise alignment, each $\mathbf{T}_i$ is obtained independently from the least-squares minimization of the inlier matches between submap i and submap i+1. To enforce the constraint given by Equation 5, we set up a matrix consisting of this constraint as well as all the local pair-wise alignments. We then minimize this to obtain alignments which can best satisfy this constraint globally but still conform to the local constraints due to pair-wise alignments.

Rather than solving directly for all the transformations, Newton's method computes a vector of corrections $\mathbf{c}$ to be subtracted from the current pair-wise alignment estimate $\mathbf{t}$: $\hat{\mathbf{t}} = \mathbf{t} - \mathbf{c}$.

Given a vector of error measurements $\mathbf{e}$ between the expected position of the SIFT landmarks and the matched position observed, and the deviation from our global constraint, we would like to solve for $\mathbf{c}$ that would eliminate this error: $\mathbf{J}\mathbf{c} = \mathbf{e}$ where $\mathbf{J}$ is the Jacobian matrix $J_{i,j} = \partial e_i / \partial x_j$.

If there are more error measurements than parameters (3n as there are 3 parameters for each alignment), this system of equations is overdetermined, and we will find an $\mathbf{c}$ that minimizes $|\mathbf{J}\mathbf{c} - \mathbf{e}|^2$. It can be shown that this minimization has the same solution as solving:

$$\mathbf{J}^\top \mathbf{J} \mathbf{c} = \mathbf{J}^\top \mathbf{e} \tag{6}$$

assuming the original nonlinear function is locally linear over the range of typical errors. Then $\mathbf{c}$ can be solved using any standard method for solving linear equation systems.

To include the constraint in Equation 5 into the framework, we need to expand the matrix equation into several scalar equations first:

$$\begin{bmatrix} \cos\theta_1 & \sin\theta_1 & x_1 \\ -\sin\theta_1 & \cos\theta_1 & z_1 \\ 0 & 0 & 1 \end{bmatrix} \dots \begin{bmatrix} \cos\theta_n & \sin\theta_n & x_n \\ -\sin\theta_n & \cos\theta_n & z_n \\ 0 & 0 & 1 \end{bmatrix} = \mathbf{I}$$

where (x_i, z_i, θ_i) are the alignment parameters from submap i and submap i+1, or submap n to submap 1 when i equals n. We can then obtain three independent scalar constraints to minimize:

$$
\begin{aligned}
e_1 =\ & \sin(\theta_1 + \dots + \theta_n) \\
e_2 =\ & x_1 + x_2 \cos\theta_1 + z_2 \sin\theta_1 + \\
& x_3 \cos(\theta_1 + \theta_2) + z_3 \sin(\theta_1 + \theta_2) + \dots + \\
& x_n \cos(\theta_1 + \dots + \theta_{n-1}) + z_n \sin(\theta_1 + \dots + \theta_{n-1}) \\
e_3 =\ & z_1 - x_2 \sin\theta_1 + z_2 \cos\theta_1 - \\
& x_3 \sin(\theta_1 + \theta_2) + z_3 \cos(\theta_1 + \theta_2) + \dots - \\
& x_n \sin(\theta_1 + \dots + \theta_{n-1}) + z_n \cos(\theta_1 + \dots + \theta_{n-1})
\end{aligned}
$$

These three constraints will correspond to the RHS of the first three rows of our matrix. Let m_i be the

number of matches between submap i and submap i+1 for each of the local pair-wise alignment, we augment our matrix system with $2m_i$ rows as we need one row for the X error and one row for the Z error for each match. Let the jth landmark at (X_i, Z_i) of submap i be matched with (X_{i+1}, Z_{i+1}) of submap i+1, we have:

$$e_{g(i)+2j-1} = X_{i+1} \cos \theta_i + Z_{i+1} \sin \theta_i + x_i - X_i$$

$$e_{g(i)+2j} = Z_{i+1} \cos \theta_i - X_{i+1} \sin \theta_i + z_i - Z_i$$

where $g(i) = 3 + 2m_1 + 2m_2 + \ldots + 2m_{i-1}$.

$\mathbf{J}$ is a $(3 + 2\sum_{i=1}^{i=n} m_i)$ by $3n$ matrix whose i^{th} row is

$$[\frac{\partial e_i}{\partial x_1} \frac{\partial e_i}{\partial z_1} \frac{\partial e_i}{\partial \theta_1} \cdots \frac{\partial e_i}{\partial x_n} \frac{\partial e_i}{\partial z_n} \frac{\partial e_i}{\partial \theta_n}]$$

The computation of these partial derivatives is done analytically based on the $\mathbf{e}$ above. Once $\mathbf{e}$ and $\mathbf{J}$ are determined, we can compute $\mathbf{J}^\top \mathbf{J}$ and $\mathbf{J}^\top \mathbf{e}$ and then can solve Equation 6 for the correction terms. This correction can be repeated if necessary until it converges, by using the current corrected estimate for the next iteration.

7   Landmark Uncertainty

While the submaps are built, covariance matrices for 3D landmarks are kept [10]. Therefore, we can incorporate this information into the pair-wise alignment as well as into the backward correction procedure.

During pair-wise alignment, we take into account the covariances of the matching 3D landmarks and employ a weighed least-squares minimization instead. The weighed least-squares equation is given by:

$$\mathbf{WJ\,c} = \mathbf{We} \qquad (7)$$

where $\mathbf{W}$ is a diagonal matrix consisting of the inverse of the standard deviation of the measurements, assuming that landmarks are independent. The covariance of the alignment estimate is given by $(\mathbf{J}^\top \mathbf{W}^\top \mathbf{WJ})^{-1}$.

For our global minimization, we can compute the covariance of the three scalar constraints from the uncertainty of each pair-wise alignment based on first order error propagation [2]:

$$cov(e_1) = (\cos^2 \sum_{i=1}^{i=n} \theta_i)\,(\sum_{i=1}^{i=n} cov(\theta_i))$$

$$cov(e_2) = cov(x_1) + \cos^2 \theta_1 cov(x_2) + x_2^2 \sin^2 \theta_1 cov(\theta_1) + \\ \sin^2 \theta_1 cov(z_2) + z_2^2 \cos^2 \theta_1 cov(\theta_1) + \ldots$$

$$cov(e_3) = cov(z_1) + \sin^2 \theta_1 cov(x_2) + x_2^2 \cos^2 \theta_1 cov(\theta_1) + \\ \cos^2 \theta_1 cov(z_2) + z_2^2 \sin^2 \theta_1 cov(\theta_1) + \ldots$$

Each of these three rows is also multiplied by the total number of submaps we are aligning, so that they contribute the appropriate weights. We also have the covariance matrix information for each landmark for the rest of the matrix. We can then carry out a weighed least-squares minimization on the whole matrix, given by Equation 7.

For the experiment above, we compute the product of all the pair-wise alignments obtained originally, i.e.,

$$\mathbf{T_1 T_2 T_3 T_4} = \begin{bmatrix} 0.9988 & -0.0489 & 0.0545 \\ 0.0489 & 0.9988 & 0.0885 \\ 0 & 0 & 1 \end{bmatrix}$$

which corresponds to a (5.45cm,8.85cm) translational and 2.8 degrees rotational misalignment. The misalignment becomes (3.00cm,5.92cm,0.43deg) for the weighed least-squares pair-wise alignment.

This is better because, by taking into account the uncertainty of the matching landmarks, we can trust the more reliable landmarks more, whereas each landmark is trusted equally previously. The misalignment is further improved to (0.15cm,0.37cm,0.03deg) for the weighed least-squares alignment with backward correction. We can now trust the more reliable pair-wise alignment more since not all the pair-wise alignment estimates are equally reliable.

The whole process is fast and it only takes 0.12 second on our robot equipped with a Pentium III 700MHz processor, excluding file I/O time. Each RANSAC pair-wise alignment takes around 0.03 second to align submaps with several hundred landmarks each, and the global minimization takes less than 0.01 second.

Even if we do not have the pair-wise alignments as the initial estimate but start with a zero vector, it still converges to the same result after several iterations.

Instead of detecting drift, we can build a new submap every M frames (in this experiment M=30), and combine the submaps together afterwards to avoid long-term drifts. Figure 4 shows 4 submaps, each of them constructed from 30 frames. Figure 5(a) shows the weighed least-squares pair-wise alignment result with a misalignment of (0.40cm,7.48cm,7.35deg). Figure 5(b) shows the weighed least-squares backward correction result with a misalignment of (0.23cm,1.59cm,0.45deg). On comparison between the two maps, a slight skew in the top left portion of Figure 5(a) can be seen.

8   Conclusions

We have proposed a vision-based map building algorithm by efficiently aligning and combining multiple submaps using highly specific SIFT natural features, to avoid the effect of drifts. Constrained optimization is used to find the maximum likelihood registration for the submaps to produce a globally consistent map, by attributing errors to all the pair-wise alignments according to landmark uncertainty.

Our pairwise alignment and backward correction are similar to the scan alignment and maximum likelihood optimization in [8] and the scan matching and map correlation in [5]. However, their algorithms are developed mainly for dense 2D range data obtained from laser or sonar and are not applicable to sparse 3D data from vision. [6] also used submaps which are maintained separately but not combined together.
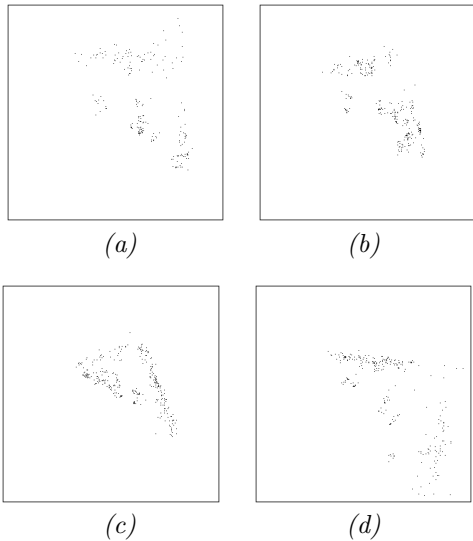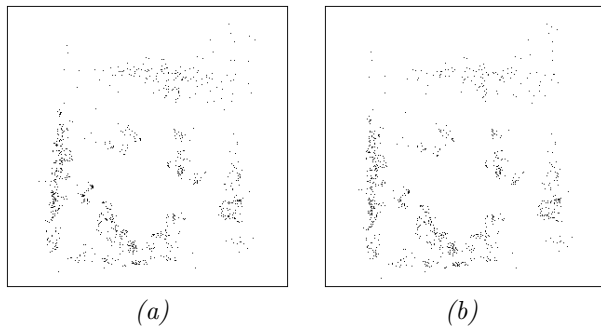
Figure 4: *Submaps built every 30 frames.*

Figure 5: *(a) Weighed least-squares pair-wise alignment for submaps in Figure 4. (b) Weighed least-squares backward correction for submaps in Figure 4.*

Integrating new data to the map incrementally and bundle adjustment using all image frames are two extremes of map building. Incremental map building does not require keeping any information from each frame and, as a result, it does not allow any backward correction when we close the loop. It has low storage and computational cost, but may lead to an inconsistent map. On the other hand, bundle adjustment requires keeping image information from each frame but it allows backward correction at each frame. It has high storage and computational cost.

The approach described above is a practical solution, like a trade-off between these two methods. It only requires information for each submap and allows backward correction between each submap. Backward correction within each submap is not necessary as, while building each submap, odometry has been corrected locally based on the SIFT landmarks.

The complexity of our approach increases by the square of the number of submaps, not by the squares of the number of landmarks, assuming a small number of overlapping landmarks between submaps. Future works include experiments in larger environments and with more complicated close-the-loop scenarios.

References

[1] N. Ayache and O.D. Faugeras. Maintaining representations of the environment of a mobile robot. *IEEE Transactions on Robotics and Automation*, 5(6):804–819, 1989.

[2] P.R. Bevington and D.K. Robinson. *Data Reduction and Error Analysis for the Physical Sciences*. McGraw-Hill, second edition, 1992.

[3] M.A. Fischler and R.C. Bolles. Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Commun. Assoc. Comp. Mach.*, 24:381–395, 1981.

[4] J. Guivant and E. Nebot. Optimization of the simultaneous localization and map building algorithm for real time implementation. *IEEE Transactions on Robotics and Automation*, 17(3):242–257, June 2001.

[5] J. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, California, November 1999.

[6] J.J. Leonard and H.J.S. Feder. A computational efficient method for large-scale concurrent mapping and localization. In *9th International Symposium of Robotics Research*, London, 1999. Springer-Verlag.

[7] D.G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh International Conference on Computer Vision (ICCV'99)*, pages 1150–1157, Kerkyra, Greece, September 1999.

[8] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, 1997.

[9] S. Se, D. Lowe, and J. Little. Local and global localization for mobile robots using visual landmarks. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 414–420, Maui, Hawaii, October 2001.

[10] S. Se, D. Lowe, and J. Little. Vision-based mobile robot localization and mapping using scale-invariant features. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2051–2058, Seoul, Korea, May 2001.

[11] R. Smith, M. Self, and P Cheeseman. A stochastic map for uncertain spatial relationships. In *4th International Symposium on Robotics Research*. MIT Press, 1987.

[12] S. Thrun, W. Burgard, and D. Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning and Autonomous Robots (joint issue)*, 31(5):1–25, 1998.

[13] S. Thrun, W. Burgard, and D. Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping. In *IEEE International Conference on Robotics and Automation (ICRA)*, San Francisco, CA, April 2000.

[14] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment - a modern synthesis. In A. Zisserman and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*. Springer-Verlag LNCS 1883, 2000.