Integration of Tracked and Recognized Features for Locally and Globally Robust Structure from Motion

Chris Engels[1], Friedrich Fraundorfer[2] and David Nistér[3,4]

[1] ESAT-PSI/VISICS, K.U. Leuven, Kasteelpark Arenberg 10
B-3001 Leuven-Heverlee, Belgium

[2] Computer Vision and Geometry Group
Swiss Federal Institute of Technology
Zurich (ETH),CH-8092, Zurich, Switzerland

[3] Center for Visualization and Virtual Environments
University of Kentucky, 1 Quality Street, Suite 800
Lexington, KY 40507-1464, USA

Abstract. We present a novel approach to structure from motion that integrates wide baseline local features with tracked features to rapidly and robustly reconstruct scenes from image sequences. Rather than assume that we can create and maintain a consistent and drift-free reconstructed map over an arbitrarily long sequence, we instead create small, independent submaps generated over short periods of time and attempt to link the submaps together via recognized features. The tracked features provide accurate pose estimates frame to frame, while the recognizable local features stabilize the estimate over larger baselines and provide a context for linking submaps together. As each frame in the submap is inserted, we apply real-time bundle adjustment to maintain a high accuracy for the submaps. Recent advances in feature-based object recognition enable us to efficiently localize and link new submaps into a reconstructed map within a localization and mapping context. Because our recognition system can operate efficiently on many more features than previous systems, our approach easily scales to larger maps. We provide results that show that accurate structure and motion estimates can be produced from a handheld camera under shaky camera motion.

1 Introduction

Recent advances in object and scene recognition based on local region descriptors has shown the possibility of highly efficient image matching and retrieval from a database. In this paper, we examine the applicability of this recognition not just for retrieval, but for the construction of 3D structure as well. Within a Structure from Motion (SfM) framework, we show how to use scene recognition to maintain consistency not just on a global scale, as is often exploited in SLAM applications, but also on a local scale when otherwise faced with failure of egomotion estimation, feature tracking, etc. The

[4] David Nistér is currently with Microsoft Live Labs.

Engels C., Fraundorfer F. and Nistér D. (2008).
Integration of Tracked and Recognized Features for Locally and Globally Robust Structure from Motion.
In *VISAPP-Robotic Perception*, pages 13-22
DOI: 10.5220/0002341800130022

SLAM community has understood the utility of structural recognition for loop closure for some time now, and research on vision-based SLAM has been especially fruitful in the past few years with the introduction of effective region descriptors, and SIFT [1] in particular.

Our approach uses a recognition system similar to the one presented by Nistér and Stewénius [2]. Rather than matching against descriptors in single images, we match against descriptors of submaps generated by short image sequences. The submaps consist of camera poses and 3D points associated with descriptors seen over multiple views. This approach offers several advantages: we reduce the redundancy inherent in matching against every descriptor in every image required to match new observations to the existing reconstruction, and we reduce outlier descriptors by requiring that each descriptor is initially seen in multiple views. Finally, by considering descriptors over a submap (and therefore in world space) rather than an individual image, we provide a more intuitive and natural connection between the descriptors and the true features in the world that a camera is observing.

To stabilize the structure estimate, we use additional features tracked frame-to-frame, and refine all points and cameras every time a pose is added using real-time bundle adjustment. As each new submap is being created, we simultaneously attempt to match descriptors from images in the submap to previously created submaps. This process produces hypotheses for linking the new submap to another set of submaps.

Our approach can be used for creating reliable handheld SfM reconstructions and provides a beneficial front-end and data association modules for a vision-based SLAM system. Many relevant approaches assume either a small number of recognizable features that can quickly reproduce the global coordinate frame, or smooth image motion produced by a steadily moving camera. Our model assumes constant failure in the system without any explicit requirement on seeing previously recognizable landmarks again. In order to create a long path, new sequences must be continuously recognized and attached to older ones.

We review related work in the next section. Section 3 describes submap construction in detail. Submap linking is discussed in Section 4. We provide several experiments in Section 5, and Section 6 concludes.

2 Previous Work

As stated in Section 1, our recognition engine is very similar to the one presented in [2], which is in turn largely based on work by Sivic and Zisserman [3]. Because of its compact image representation and its efficient matching process, it is well suited for mobile robot applications, including those operating in very large environments. The image search finds similar images by matching local features. First, the local features are detected in each image using the MSER detector [4]. Then a feature vector is computed over a local region using the SIFT descriptor [5]. Each SIFT feature vector is quantized into a vocabulary tree. A single integer value, called a visual word (VW), is assigned to a 128-dimensional SIFT feature vector. This results in a very compact image representation, where each image is represented by a set of visual words. Matching two images can be done by comparing the two lists of visual words. For image search applications

the image database is set up as an inverted file. For each VW, the inverted file maintains a list of image indices in which the visual word occurred. For an image query the indexed lists of all VWs that occur in the query image are processed. Weighted votes are collected for the images in the lists. The database image with the highest score is then selected as the best match. The query process is very efficient: tests show that a query in a 1 million image database (with an average number of 200 VWs per image) takes 0.02s. This results in a frame-rate of about 50Hz, well suited for real-time applications. Because of the compact image representation, a 1 million image database can be stored in less than 4GB, allowing it to be kept in RAM on current computers. Adding a new image to the inverted file database requires constant time only, since it is only necessary to add the image index to the corresponding VW lists. This time is almost negligible compared to the query. MSER detection and SIFT feature computation can be done at a frame-rate of 15Hz on 640×480 images.

Online SLAM and SfM systems very often use a Kalman Filter [6] or Extended Kalman Filter (EKF) for smoothing out estimation error. They use this method primarily because updates occur in constant time, which is necessary in the presence of increasing observations and data. Offline systems can use bundle adjustment to iteratively minimize error, but this process can be very slow for large numbers of observations. Because we have a limited number of frames within each submap, we use an online bundle adjuster for refinement of our submaps after each frame is added. Triggs et al. [7] provides a detailed survey of bundle adjustment methods, while Engels et al. [8] discusses specific efficiency requirements for running windowed bundle adjustment at frame rate.

The Atlas framework created by Bosse et al. [9] generates local submaps (mapframes) from laser range data and represents the world as a connectivity graph between the submap coordinate frames. The framework performs localization through a Kalman Filter. Later, Leonard and Newman [10] describe a SLAM algorithm that uses overlapping submaps in local coordinate frames. For efficiency, their system separates the estimation of the internal state of each submap from the global location estimate for that submap. While this approach precludes error correction within a submap after creation, the authors demonstrate that the error within the estimate is acceptably close to that of a full solution.

There are a number of vision-based SLAM approaches relevant to our system. Davison et al. [11] demonstrate a real-time handheld monocular SLAM approach that tracks a small number of points with high precision using an EKF. They focus on an enclosed environment that has a bound on landmarks and known initial points that allow a metric reconstruction, so scalability is not an issue, unlike the case in our paper. More interestingly, Sim and Little [12] present a system that performs real time vision-based SLAM using a stereo head and is designed with obstacle avoidance in mind. 36-dimensional SIFT features are extracted and quantized into identifiers using a *k-d* tree approximate nearest neighbor algorithm. A Rao-Blackwellised particle filter is used to solve the localization and mapping problem. Their landmark map uses an adapted version of the FastSLAM data structure [13]. As mentioned in [2], the hierarchical *k*-means approach has several orders of magnitude more descriptors than a *k-d* tree. Because their approach uses a stereo head, determining the 3D location of each SIFT descriptor is simple. In our

case, we need to observe the descriptor from multiple views whose computed position is greatly determined by the structure estimate.

3 Construction of the Local Submap

We construct each submap in a local coordinate frame by tracking local feature points and wide baseline features simultaneously. We generate pose estimates in a manner similar to the monocular visual odometry scheme proposed in [14]. In that approach, the authors track points over three views and compute initial pose estimates within a robust RANSAC framework. The tracked features are then triangulated into 3D world points; subsequent frames are computed via single-view pose using world-point to image-point correspondences. After a certain number of frames, the method performs another three-view pose estimation, stitches that estimate into the current coordinate system, and continues with single-view estimation. The idea is to prevent long term degradation of the triangulated points that eventually causes the odometry to fail. This paper proposes a different approach: instead of attempting to continuously compute relative camera motion in the presence of drift and degradation, we simply stop using the current coordinate system and its related structure and restart in a separate submap. We leave connecting the submaps to the linking step described in Section 4.

3.1 Local Feature Tracking

Our local feature tracker combines the advantages of tracking Harris corners [15] with the KLT tracker [16]. Briefly, the tracker initializes feature points in the image based on Harris corner strength, with the constraint that the points should be evenly-distributed throughout the image. The algorithm computes frame to frame tracking via normalized cross correlation on an image pyramid, which like the KLT tracker gives subpixel precision of the feature location, while the normalization provides robustness to varying camera gain.

3.2 Tracking Features for Recognition and Wide Baseline Matching

The features tracked in Section 3.1 have several drawbacks. First, they are difficult to track outside of a frame-by-frame basis, especially over wide baselines. If a feature is lost in a single frame, it cannot be tracked further. Second, local features can not be matched efficiently in any way useful for recognition. In addition to tracking local features frame to frame, we also match wide baseline features as they occur. As discussed in Section 2, we use the hierarchically-quantized visual words defined in [2]. When such features are redetected over multiple—but not necessarily consecutive—frames, we triangulate them to associate a 3D position to the point. We use the center of mass of the MSER as our image point. While these features are highly redetectable, as shown in [17], the computed center of mass is more sensitive to noise than are our local features, which have subpixel precision. This sensitivity results in a higher degree of uncertainty of the 3D position. We compensate for this error by including the wide baseline features in a real-time bundle adjustment step.

3.3 Real-time Bundle Adjustment

We use Levenberg-Marquardt bundle adjustment implemented according to Engels et al.[8] to refine the structure of our submap. We refine our structure with several iterations after every pose computation, using the cost function stated in Section 3.4. In order to improve performance, we freeze cameras in place after a number of additional poses are added. Besides improving the camera poses, we also seek to improve the 3D locations of the visual words, which will be critical when linking submaps accurately.

3.4 Failure Detection

In an ideal environment, both wide baseline and tracked features would be redetected accurately in every image in a sequence. However, in practical situations such as estimating structure from a handheld camera, the camera motion may be shaky or rotate rapidly. Feature points are lost or mismatched, which may cause the current reconstruction to fail. If such a submap were placed into the world, it could propagate the error globally. We therefore need to detect when a reconstruction begins to fail and preempt computation of that submap. Because we can reconnect subsequent submaps to the global coordinate system at a later point, we can simply exclude the failed submap and start a new one.
Our strategy for detecting estimation failures is based on searching for significant increases in total reprojection error for each camera. To be robust to outliers, we use a robust cost function on the assumption that the reprojection error follows a Cauchy distribution:

$$C(P) = \sum_i \ln(1 + \frac{e_i^2}{\sigma^2}), \tag{1}$$

where P is a camera, e_i is the reprojection error of a world point-image point correspondence, and $sigma$ is a standard deviation. Because the bundle adjustment process can move the camera pose and world points, $C(P)$ can change as additional poses are added. An incorrect pose will have an effect on the structure and other poses after bundling. If

$$\frac{C(P)_{updated} - C(P)_{original}}{C(P)_{original}} > c \tag{2}$$

for some constant c, we judge the structure to be unstable and start a new submap. Note that $C(P)_{updated}$ may decrease freely without causing failure, which of course we would expect within bundle adjustment. We include additional failure conditions for the loss of large percentages of feature points and poor initial reprojection error.

4 Efficient Recognition and Linking of Submaps

Rather than creating a database of all individual frames we create a database of submaps, each built from a set of frames. This is possible as our submap representation allows us to match a frame directly to a submap. It is not necessary to match individual frames

and then figure out to which submap they belong. This is very beneficial as it also reduces the database size and time needed for matching. If storing individual frames in the database, corresponding VWs between subsequent frames would be stored multiple times, causing unnecessarily redundant data. In our case the submap contains only one instance of a VW tracked through several frames. In addition, the matching time is linear in the number of database entries (frames, submaps). When using submaps we have fewer entries and thus the matching is faster. We adopt the matching scheme from [2, 3] for efficient and scalable matching. The main reason for the high performance is in the use of quantized descriptors and the use of an inverted file. The inverted file stores the information in which submap each VW occurs. The inverted file consists of a list for each possible VW. For each VW in the submap, the submap's index is added to the corresponding list. When adding a submap to the database, we only need to insert the number of VWs seen in the submap. As each VW is represented as an integer, this update is very efficient. A database query is also very efficient, however it means to compute distance scores between the current frame and each stored submap. Scoring is performed as described in [2]. Although scoring is linear in the number of submaps the operations necessary for scoring a single frame depend on the number of VWs in the frame to score, which is rather low. Therefore a single scoring operation is extremely fast. Matching by score only gives a tentative match. This match is then verified by geometric means. The submaps contain 3D reconstructions of the VWs. For geometric verification, we run 3D-2D pose estimation between the submap and the current frame. The geometry score is the number of inliers in the robust pose estimation using RANSAC. The geometry score is a very strong criterion and makes the matching very reliable. When computing the geometry score, the pose of the current frame in the coordinate system of the submap is computed. This pose is stored and will be used to compute the coordinate transformation between submaps. The individual frame matches will be used as link hypotheses between submaps. If frames from a submap match to more than one other submap we create multiple links.
To maintain links between the different submaps an undirected graph is used. There are two different types of links: Topological links and metric link. A topological link can be seen as hypothesis from the recognition scheme. For each topological link, the corresponding coordinate transform is sought. If successfully estimated, we can upgrade the topological link to a metric link, if not, we just keep the topological link. The graph data structure allows us to transform all the submaps into a common coordinate system if a global map is requested. An arbitrary submap can be chosen as the common coordinate system. By graph search the necessary chains of transformations into the selected common coordinate system can be determined. A metric link is represented by a similarity transform T, consisting of a rotation R, translation t and a scale factor s. The relative camera positions within a submap are kept fixed.

4.1 Transformation between Submaps

Each submap has its own local coordinate system where the first camera is assumed to be located at $P = [I|0]$. For each linked pair of submaps the transformation from one coordinate system into the other is sought. In the matching step we already computed camera poses from the frames of the current submap in the coordinate system of the

matching submap. With two computed cameras we can establish the transform T. The scale factor s can be computed from the distance ratio of matching camera pairs. R and t can be computed from a single matching camera. This initial transformation allows us to put the cameras and the 3D points of the submaps into a common coordinate frame. A subsequent bundle adjustment on the two submaps is performed to increase accuracy. We then compute the transformation T again but now from the bundle adjusted cameras. To output a global map, an arbitrary submap can be chosen to define a global coordinate system, and all other submaps' coordinate frames can be transformed into that one. A final offline bundle adjustment step can optionally be performed to further align submaps that were not optimized pairwise before. One attractive alternative to the offline requirement would be to use the out-of-core bundle adjustment method proposed by Ni et al. [18], which offers the possibility of aligning a larger number of submaps within real-time constraints.

5 Experiments

5.1 Small Scale Turntable Sequence

To demonstrate the functionality of the method, we conduct a turntable experiment. A cup is placed on a turntable and a complete turn is captured. The image sequence consists of 158 frames. We apply our online SfM method to this image sequence. For this experiment, we set the number of frames per submap to 32. Fig. 1 shows the results of the successful structure and motion estimation. Five submaps are created and linked together to form the circular camera motion. Only a small gap remains between the first and last camera, which ideally should be at the same position. A final offline bundle adjustment step closes the loop in Fig. 1(c).

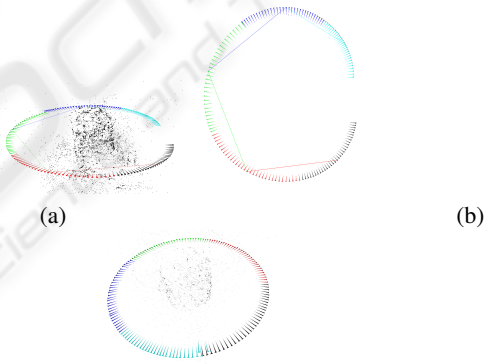

Fig. 1. Structure and motion for the turntable sequence. Each submap is shown in a different color. The lines illustrate the links between the submaps. A small gap remains between the first and the last camera, which we correct through bundle adjustment in (c).

5.2 Erratic Motion

Erratic motion, dropped frames and shaking cameras are critical conditions for most tracking based SLAM and structure from motion methods. In this experiment we demonstrate that our method successfully can overcome such critical conditions. A hand-held camera was moved in a sideway motion. At some point we started shaking the camera and pointing it upward. Fig. 2(a) shows example images from the critical part of the sequence. Fig. 2(b) shows the computed structure and motion from our method. The erratic movements happened between the red and green submap. When the erratic movements started, tracking started to fail and a new submap was initialized. With submap linking, the new submap is rapidly connected to the previous track. The only delays are the submap reinitialization and linking, which both occur within real-time constraints in failure and non-failure cases.

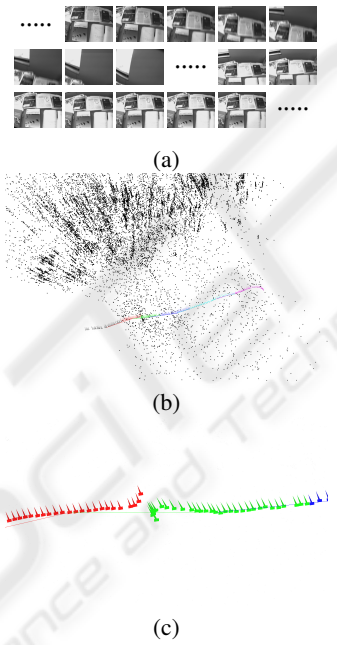

Fig. 2. Image sequence with shaking and erratic movements. (a) Images from the sequence around the critical part. (b) The computed structure and motion. The erratic movements occured between the red and green submap. When tracking failed a new submap was started and linked to the previous submap. (c) A zoom into the critical part.

5.3 Indoor Experiment

The image sequence for the indoor experiment was acquired by walking around in a room. Our SfM method was applied with the same settings as in the previous experi-

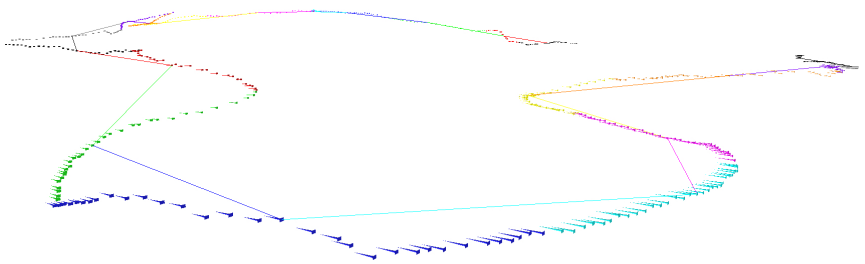

Fig. 3. Camera motion for a sequence of 1015 frames. It consists of 21 submaps.

ments. Fig. 3 shows the estimated camera poses for the 1015 frames sequence. A total of 21 submaps was created during reconstruction. It is notable that the scale between the different submaps is estimated accurately for the whole sequence showing only a marginally drift. As in 5.1 inaccuracies could be removed with a final off-line bundle adjustment.

6 Conclusions

We demonstrate a Structure from Motion scheme that successfully combines recognition and tracking. The computed structure consists of linked local 3D reconstructions. The local submaps combine wide baseline and locally tracked features and stabilize the estimate via real time bundle adjustment. Links between submaps are efficiently detected with a recognition engine. The experiments demonstrate the functionality of our approach and show the potential for building larger scale reconstructions.

Acknowledgements

This work was partially funded by K.U. Leuven GOA.

References

1. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60 (2004) 91–110
2. Nistér, D., Stewénius, H.: Scalable recognition with a vocabulary tree. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, New York City, New York. (2006)
3. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: Proc. 9th IEEE International Conference on Computer Vision, Nice, France. (2003) 1470–1477

4. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: Proc. 13th British Machine Vision Conference, Cardiff, UK. (2002) 384–393
5. Lowe, D.: Object recognition from local scale-invariant features. In: Proc. 7th International Conference on Computer Vision, Kerkyra, Greece. (1999) 1150–1157
6. Kalman, R.: A new approach to linear filtering and prediction problems. Transactions of the ASME: Journal of Basic Engineering (1960) 35–45
7. Triggs, B., McLauchlan, P., Hartley, R., Fitzgibbon, A.: Bundle adjustment: A modern synthesis. In: Vision Algorithms Workshop: Theory and Practice. (1999) 298–372
8. Engels, C., Stewénius, H., Nistér, D.: Bundle adjustment rules. In: Photogrammetric Computer Vision. (2006)
9. Bosse, M., Newman, P., Leonard, J., Teller, S.: An atlas framework for scalable mapping. In: IEEE International Conference on Robotics and Automation. (2003) 1234–1240
10. Leonard, J.J., Newman, P.M.: Consistent, convergent, and constant-time slam. In: International Joint Conference on Artificial Intelligence. (2003) 1143–1150
11. Davison, A., Reid, I., Molton, N., Stasse, O.: Monoslam: Real-time single camera slam. IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (2007) 1052–1067
12. Sim, R., Little, J.J.: Autonomous vision-based exploration and mapping using hybrid maps and Rao-Blackwellised particle filters. In: Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS), Beijing, IEEE/RSJ, IEEE Press (2006) 2082–2089
13. Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B.: Fastslam: A factored solution to the simultaneous localization and mapping problem. In: Proc. of the AAAI National Conference on Artificial Intelligence. (2002) 593–598
14. Nistér, D., Naroditsky, O., Bergen, J.: Visual odometry. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Washington, DC. (2004) I: 652–659
15. Harris, C., Stephens, M.: A combined corner and edge detector. In: Alvey Vision Conference. (1988)
16. Tomasi, C., Kanade, T.: Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University (1991)
17. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Van Gool, L.: A comparison of affine region detectors. International Journal of Computer Vision 65 (2005) 43–72
18. Ni, K., Steedly, D., Dellaert, F.: Out-of-core bundle adjustment for large-scale 3d reconstruction. In: Proc. 11th IEEE International Conference on Computer Vision, Rio de Jeneiro, Brazil, IEEE (2007)