Dense 3D Mapping with Monocular Vision

Kamil Wnuk
Computer Science Department, Harvey Mudd College
340 E. Foothill Blvd.
Claremont, CA 91711
kwnuk@cs.hmc.edu
ACM #: 8757965
Advisor: Zachary Dodds

Abstract

Currently, the vast majority of autonomous mapping in robotics relies on direct measurements from costly devices such as sonar, infrared, and laser range finders. Adapting established methodologies from the structure-from-motion (SFM) subfield of computer vision to data commonly available in robotics, we have created a unique toolkit, able to render visually dense 3d maps from odometrically annotated monocular vision. We then leverage the richness of our representation to augment the widely-used Monte Carlo Localization algorithm (MCL). The new version, coined Monte Carlo Correction, provides for pose updates and hypothesis merging on top of MCL, and demonstrates the potential of visually dense 3d maps as a basis for amplifying recent advances in robotics. We thus show that although computationally intensive, monocular vision provides a low-cost, yet highly capable alternative to currently popular mapping sensors.

Problem and Motivation

As humans, most of us take the computation involved in our visual perception of the world for granted. Despite efforts in the robotics and computer vision communities, a method of representing visual data to guide an agent around an environment, that is as efficient and effective as human vision, has not yet been developed.

Currently, the vast majority of robotic navigation relies on direct range-to-obstacle measurements from relatively costly devices such as sonar or infrared rings ($200-400) and laser range finders ($4000). For this work, the primary sensor is a $25 webcamera. In fact, most systems doing autonomous navigation typically would benefit from the additional information that a camera would provide.

The research we are conducting contributes to finding a cost effective solution to the representation problem and investigates how popular spatial reasoning algorithms can be adapted to take advantage of available visual information. Our newly developed toolkit allows for construction of visually dense 3d maps from data gathered by a platform equipped solely with monocular vision and odometry. In the present phase of our work, we are investigating the potential of this representation for improvement of popular spatial reasoning algorithms. We believe that dense 3d maps show strong potential for amplifying many existing navigational algorithms, and thus a tool for constructing such maps will be beneficial to the robotics community.

Background and Related Work

To put our research into context with other work, we have created a taxonomy of vision-based mapping, shown in Figure 1. The horizontal axis of the taxonomy corresponds to the density of image information represented in the map, while the vertical axis indicates how the relation of map primitives to each other is incorporated into the representation.

The most common type of data representation chosen for robotic systems using vision is one with sparse image information, depicted on the right side of our taxonomy. For example, Kosecka and Li, [1], used scale-invariant (SIFT) features and histograms of 2d images to localize in indoor environments. This sparse collection of 2d features characterizes the upper right-hand corner of Figure 1's taxonomy. Another such example is a monocular map building system developed by Neira et al., [2]. Their maps were also sparse,

based on vertical edges in the environment, but because these edges were mapped in a global set of coordinates their approach inhabits the lower right-hand area in Figure 1.

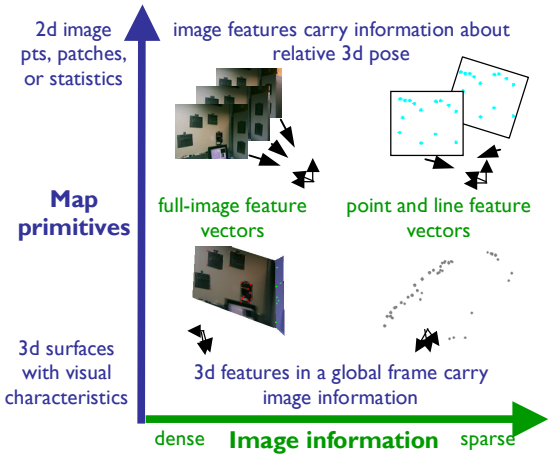

Figure 1: Taxonomy of vision based mapping

We are particularly interested in creating and navigating based on maps that classify in the lower left corner of our taxonomy, since this type of map representation incorporates the most information from acquired visual data. Previous vision based mapping work in this region of our taxonomy has been done by Iocchi et al., [3], who used a stereo vision system to create 3d maps of planar indoor environments. A number of other mapping techniques have been developed using different permutations of stereo cameras, laser range finders, and panoramic vision. Our focus, however, is on the more modest sensing capability of monocular vision. To accomplish this we draw upon the structure-from-motion (SFM) subfield of the computer vision community, which has experienced great success creating dense 3d reconstructions from single camera data, as demonstrated by Pollefeys et al.'s method of modeling structure using a hand-held video camera, [4].

Our approach bridges the gap between computer vision and robotics by adapting established SFM techniques, [5], for use in traditional robotics domains. We take advantage of robot odometry and the planarity of indoor environments to create visually dense 3d environmental reconstructions using a single camera. Just as our robotic focus differentiates our work from mainstream computer vision, the limited nature of our sensor suite also distinguishes us from sensor-rich world acquisition methods more commonly found on high-end robotics platforms.

Approach and Uniqueness

The map building process consists of five principal stages: feature identification and tracking, feature triangulation, surface fitting, estimation of homographies between images and the 3d surface, and mosaicking. This section will give an overview of each of these stages, highlighting places where our approach departs from other work.

The feature identification and tracking (Figure 2a) on the input image sequence relies on Stan Birchfield's KLT tracker [6]. This first processing stage outputs the coordinates of every tracked feature, for each image that it appears in. Each feature is assigned a unique ID upon its first tracking between images, and maintains that ID as long as it continues to be tracked through the sequence of images.

These tracked features' coordinates are then passed to our Triangulator. An earlier, offline calibration provides the internal camera parameters, such as focal length. With the odometrically annotated data, the Triangulator computes the camera rotations and translations for each image. A least-squares estimation follows, providing the 3d coordinates of each of the features (Figure 2b). A critical component of the Triangulator is that it allows for feature IDs to be propagated to the 3d points, thus maintaining a direct correspondence with the original images. This correspondence enables the later visual mosaicking of the estimated surfaces.

To estimate planar surfaces from the 3d point coordinates, the RANSAC [7] algorithm is used to find the optimal set of planes to fit the points (Figure 2c). For each plane the algorithm generates 2000 hypotheses and the optimal fit is decided based on a set of heuristics. This subsystem outputs a set of plane normals, a list of feature points to which each plane is fit, and the 2d coordinates of each feature when projected onto its plane. It is during this stage that most noisy points resulting from mistracking of features or

tracking of non static features, such as specularities, are classified as outliers and eliminated from the system, as shown in Figure 3.

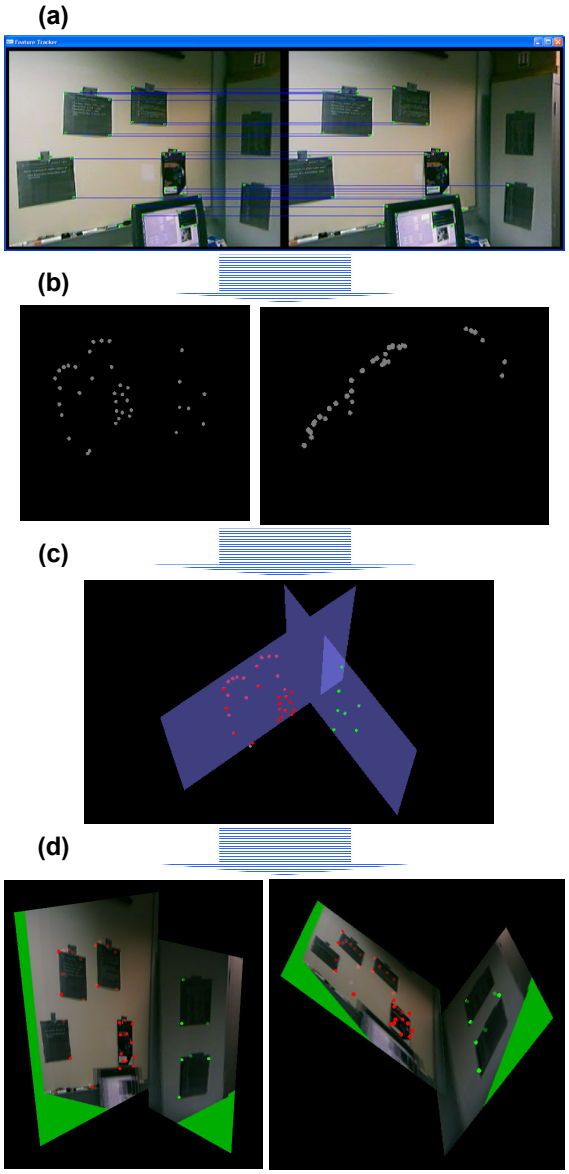

Figure 2: Various stages of map construction for a three image dataset of a lab corner: **(a)** feature tracking, **(b)** triangulation results, **(c)** best fit planes with features color coded according to which plane they belong to, and **(d)** two views of the final textured planar map.

The final two stages are responsible for mapping visual data onto the estimated structure. For each plane in the structure, a separate texture image is created. Homographies between sequential images and the environment's planes are found using the results from the tracking phase coupled with data from the plane fitting. These homographies are passed to the mosaicker, which creates a texture image for each plane (Figure 4). To accompany each texture is a file holding the coordinates of all the features appearing in the mosaic and their IDs. This allows the texture to be correctly aligned with the points in 3d space to create a visually dense map of the environment through which the robot has traveled.

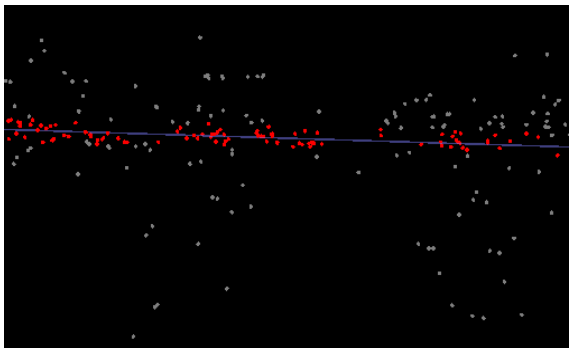

Figure 3: Plane fitting results for part of a hallway dataset consisting of 63 images. Here RANSAC has selected 113 inliers (red), from a total of 428 tracked points (gray).

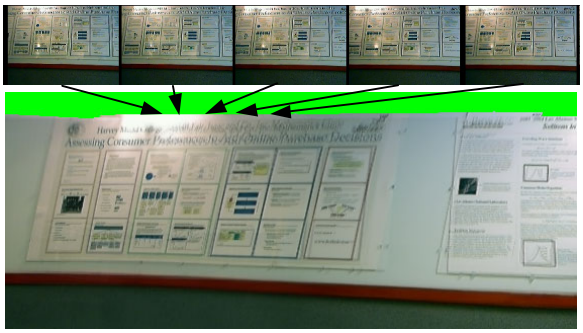

Figure 4: The initial mosaic created from a set of 63 sequential hallway images, 5 of which appear at the top. Before actually being textured onto a plane, a mosaic is morphed so that its features match up to the feature points in 3d.

Results and Contributions

Using our toolkit we were able to successfully map a corner of our lab (Figure 2d) as well as a nearby hallway (Figure 5), and we plan to continue testing with progressively larger and more complex datasets.

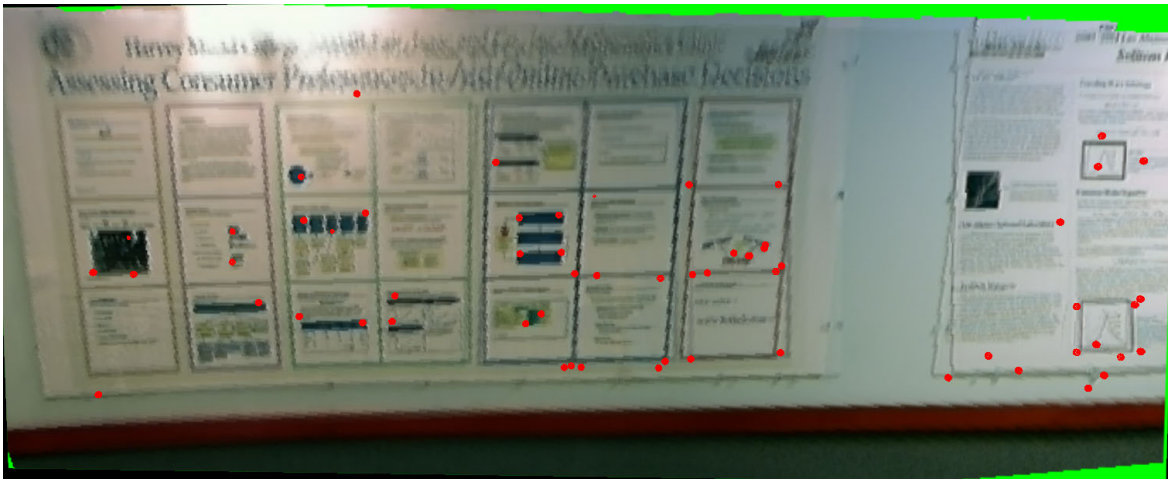

Figure 5: A straight-on view of the final hallway map with 3d feature points simultaneously displayed in red. An interesting note on this map is that a specularity that traveled across the bottom of the component images as the robot moved has been eliminated by pixel averaging.

Evaluation

To evaluate the accuracy of the maps constructed by our toolkit we measured the actual physical locations of distinct visual features and compared them to their locations in our 3d map for a single plane of our lab corner dataset. We found the average location error over 12 features on the whiteboard plane to be approximately 3.5 cm, or 2% of the distance of the plane from the camera. This result is shown in Figure 6.

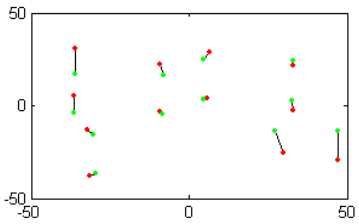

Figure 6: Feature location error on the whiteboard plane with real locations shown in green and map estimates in red.

Application to Robot Localization

An intrinsic ability provided by our maps is the capability to render views of the mapped environment from any conceivable pose the robot may be in, allowing for the generation of anticipatory visual data (Figure 6). This naturally supports the use of the Monte Carlo Localization algorithm (MCL), [8], a probabilistic method commonly used to enable mobile robots to locate themselves within a known map. During a single iteration of the MCL algorithm on our system, hypotheses (particles) whose anticipatory renderings match closely to what the robot is actually seeing are weighted with higher probability than those having more distant matches. A new generation of particles is then created by random resampling and used as the basis of the next iteration. After several iterations in an unambiguous environment, the particles converge to approximate the robot's actual position in the map. Figure 7 provides a visualization of a single iteration of MCL in one dimension.

Figure 6: Anticipatory renderings generated from the hallway map (top and bottom), compared with a raw image recorded by the robot (center).

We have devised a way to exploit the richness of our visually dense 3d maps and the

anticipatory rendering capability to improve MCL. Our augmented version, coined Monte Carlo Correction (MCC), identifies common features between the anticipatory renderings and what the robot is actually seeing using the KLT feature tracker. This tracking information is sufficient to calculate how a given particle must be shifted in order to best represent the robot's current position. This shift occurs before the resampling step, effectively collapsing particles within a certain range to the robot's actual position. Once the resampling occurs, duplicate particles representing the robot's actual position are merged into a single hypothesis. This approach promises to converge more rapidly on the robot's position and to be more efficient by eliminating duplicate hypotheses. A one dimensional visualization of a single MCC iteration can be found in Figure 8.

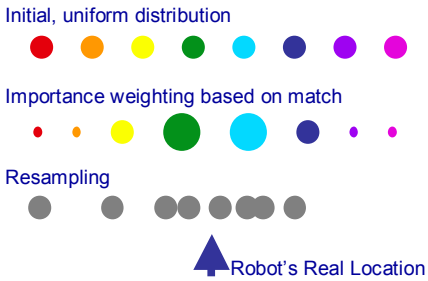

Figure 7: Single iteration of MCL in 1 dimension.

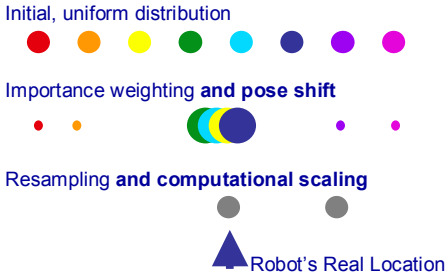

Figure 8: Single iteration of MCC in 1 dimension

Perspective

As our current work demonstrates, the environmental representation used for spatial reasoning tasks makes a significant difference in how an autonomous system is able to reason about the world. With our particular representation, we were able to leverage the richness of visual data to improve upon the standard MCL algorithm. Similarly, we believe that visually dense 3d maps can serve as a basis for amplifying other recent advances in spatial reasoning algorithms. Upon completion of fully implementing the MCC algorithm discussed above, our future work will thus focus on attempting to leverage our representation to improve spatial reasoning tasks such as FastSLAM, [9], a recently developed efficient solution to the Simultaneous Localization and Mapping problem (SLAM).

To promote continued work in applying SFM methodologies to robotics and further investigation of the applications of visually dense 3d maps, the full source code (C++ for Windows) of our toolkit is available online. We feel that further work in these areas will endow the robotics community with highly capable, very low cost robotic platforms, thus making vision-based robotics accessible far beyond research laboratories.

References

[1] Kosecka, J. and Li, F., "Vision based topological markov localization," *Proc., ICRA*, pp. 1481-1486 (Apr 2004).

[2] Neira, J., Ribeiro, M. I., and Tardós, J. D., "Mobile robot localization and map building using monocular vision," *Proc., 5th Int. Symp. on Intelligent Robotic Systems*, Stockholm, Sweden, pp 275-284 (July 1997).

[3] Iocchi, L., Konolige, K., and Bajracharya, M., "Visually Realistic Mapping of a Planar Environment with Stereo," Experimental Robotics VII LNCIS, D. Rus and S. Singh, Eds., Springer, pp 521 – 532 (June 2003)

[4] Pollefeys, M., Vergauwen, M., Verblest, F., Cornelis, K., Tops, J., and Koch, R. "Visual modeling with a hand-held camera," *Int. Journal of Computer Vision*, 59 (3), pp 207-232 (Oct 2004).

[5] Ma, Y., Soatto, S., Kosecka, J., and Sastry, S. *An Invitation to 3D Vision: From Images to Models* Springer Verlag, December 2003.

[6] Shi, J. and Tomasi, C. "Good features to track," *Proc., CVPR*, pp 593-600 (June 1994). Code available 8.1.04 at **www.ces.clemson.edu/~stb/klt**.

[7] Fischler, M. A. and Bolles, R. C., "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Comm. Of the ACM*, 24 (6), pp 381-395 (June 1981).

[8] Thrun, S., Fox, D., Burgard, W., and Dellaert, F., "Robust Monte Carlo Localization for Mobile Robots," *Artificial Intelligence*, 128(1-2), (2001)

[9] Montemerlo, M., and Thrun, S., "Simultaneous Localization and Mapping with Unknown Data Association Using FastSLAM," In *IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, Taiwan, 2003