TEXTURING LONG PLANAR SURFACES WITH IMPRECISE CAMERA POSES FOR INDOOR 3D MODELING

Michael Anderson, Kurt Keutzer and Avideh Zakhor

Department of Electrical Engineering and Computer Science, University of California at Berkeley
{mjanders,keutzer,avz}@eecs.berkeley.edu

ABSTRACT

Automated 3D modeling of building interiors is useful in applications such as virtual reality and environment mapping. Texture mapping walls is an important step in visualizing the results of an indoor 3D modeling system. Methods to localize the camera in the 3D scene often are not pixel accurate, meaning that when multiple images are used for texture mapping there are seams and discontinuities between these images. Several approaches to this problem have been proposed but each suffer from a distinct problem of error accumulation for long chains of images, such as those from a long corridor. We propose a new approach to texture mapping planar surfaces that eliminates discontinuities between images but does not suffer from error accumulation for long chains. We validate this approach using images from several long hallways with data generated by a human operated backpack 3D indoor modeling system.

Index Terms— 3D modeling, mosaicing, texture mapping

1. INTRODUCTION

Three-dimensional modeling of indoor environments has a variety of applications such as training and simulation for disaster management, virtual heritage conservation, and mapping of hazardous sites. Manual construction of these models can be time consuming, and as such, automated 3D site modeling has garnered much interest in recent years.

An indoor modeling system must first be able to simultaneously estimate the camera's location within an environment and the 3D structure of the environment. This problem is studied by the robotics and computer vision communities as the simultaneous localization and mapping (SLAM) problem. It is usually solved with the aid of laser range scanners, cameras, and inertial measurement units (IMUs) that survey the environment in a vehicle or human-operated backpack [1, 3]. The devices, along with various localization algorithms, can generate a point cloud which is then processed by a surface reconstruction algorithm to infer structure such as walls and ceilings. Finally, the reconstructed surfaces are textured using captured images and the estimated camera position for each photo.

Though the localization errors resulting from these laser based algorithms are quite low even in these complex environments, when the resulting recovered pose is used to texture map camera imagery onto the resulting 3D triangular mesh models, there is significant misalignment between successive images used to texture map neighboring triangles. This implies that the scan matching based localization algorithms are not pixel accurate. The misalignment problem is the focus of this paper.

Previous work on this problem proposed an image based approach in which the pose from laser scan matching based localization is refined using camera imagery [1]. Traditional image stitching approaches using image correspondences have also been tried [2]. However both these approaches suffer from error accumulation when used for long chains, for example 20 or 40 images. These long chains of images are common in indoor modeling when buildings have long corridors.

The approach presented here gracefully handles long chains of images without error accumulation, which we demonstrate on data from our human operated backpack 3D indoor modeling system [3]. This paper is organized as follows. Section 2 outlines the problem of texture mapping for indoor 3D modeling and demonstrate existing approaches. Section 3 describes our approach. Section 4 shows results using data generated from our backpack system and presents conclusions.

2. TEXTURE MAPPING CORRIDORS

The geometry of the the texture mapping problem for indoor 3D modeling is shown in Figure 1. We are given a set of M images. Each image has a camera matrix P_i for $i = 1..M$, which translates a point in the world coordinate system to a point in image i's coordinates. A camera matrix P_i is composed of the camera's intrinsic parameters, such as focal length and image center, as well as the extrinsic parameters which specify the rotation and translation of the camera center's position with respect to the world coordinates at the time

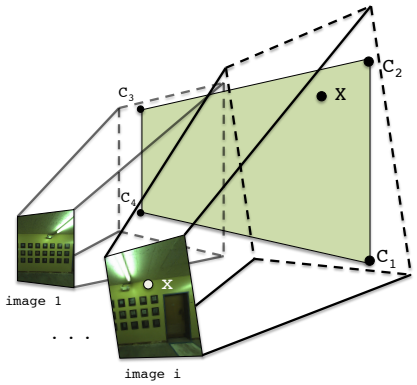


Fig. 1. The plane is specified in 3D space by the four corners C_1 to C_4. Images are related to the plane through the camera matrices $P_{1..M}$.

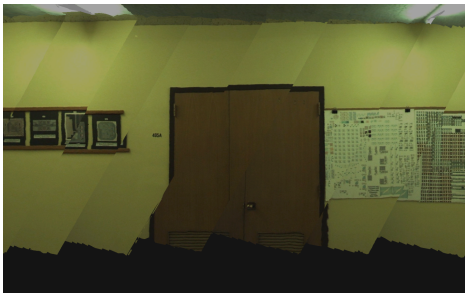


Fig. 2. The result of naive texture mapping based on the imprecise camera matrices estimated by the localization system.

that image i was taken. These extrinsic parameters are determined by the localization hardware and algorithms as part of the indoor modeling system. A point X on the plane can be related to its corresponding pixel x in image i through the following equation:

$$x = project(P_i X)$$

where $X = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ and $project(X) = \begin{pmatrix} x/z \\ y/z \end{pmatrix}$

We are also given a plane generated by the surface reconstruction system, which is to be texture mapped by these images. The plane is defined by four corner points C_1 to C_4 in world coordinates and a normal vector indicating the front facing side of the plane. The challenge is to texture the plane using these images, while eliminating any discontinuities or seams that would suggest that the plane was not composed of a single continuous image.

2.1. Naive mapping

Ignoring the fact that the camera matrices $P_{1..M}$ are inaccurate, one can texture map the plane simply by discretizing the plane and projecting each point on the plane back to the images. Each point on the plane is projected separately into all the images in the set. For most images it lands outside of the range of valid pixels for that image. If it does land inside the valid range however, the pixel's color and intensity can be copied to this point on the plane. If the point lands in the valid region of multiple images, then either the pixel values can be mixed or one of the images may be chosen as a representative for that point based on a predetermined precedence ordering.

As Figure 2 demonstrates, this naive mapping leads to significant misalignment between successive images. This image was generated by choosing images from the set whose camera pose was nearby and roughly perpendicular to the plane. This suggests that while the errors in the localization system are quite low, they are not pixel accurate. For photo-realistic texture mapping, either the camera matrices need to be refined such that the localization is pixel accurate, or image processing techniques need to be applied to provide this illusion. The following are two existing techniques to solve this problem.

2.2. Image Mosaicing

When images are taken of a plane from arbitrary overlapping positions, they are related by homography [4]. Thus, existing homography-based image mosaicing algorithms are applicable [2]. However, errors can compound when long chains of images are mosaiced together using these approaches. For example, a pixel in the nth image in the chain must be translated into the first image's coordinates by multiplying by the 3×3 matrix $H_1 H_2 H_3 ... H_n$. Any error in one of these homography matrices is propagated to all further images until the chain is broken. For some chains of images this can happen almost immediately due to erroneous correspondence matches and the resulting image mosaic is grossly misshapen.

2.3. Graph-based optimization

Another approach is to used the graph-based nonlinear optimization framework developed in [1]. This approach aims to refine the camera matrices, each with 6 degrees of freedom, using image correspondences to guide the process. This process is carried out at the same time as the laser based backpack localization and is therefore specific to the system in [1, 3]. Unfortunately, this approach suffers from the same error propagation problem shown in Figure 3.

3. OUR APPROACH

3.1. Preprocessing

As a starting point, we naively texture map the plane using the given imprecise camera matrices for each image in the set. This is done by discretizing the plane into an arbitrary density of pixels and projecting each pixel into the image plane using the estimated camera matrix P_i. We later intend to shift

Fig. 3. Using the graph-based localization refinement algorithm from [11] suffers from the problem of compounding errors.

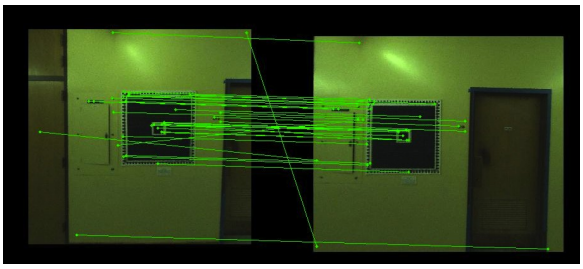


Fig. 4. SIFT feature matches are determined between neighboring images. The right image is then shifted on the plane so that the matches are aligned.

some of these projected images around on the plane and remove some others. Therefore, each image is stored intact in its own data structure so that it can later be merged with the other images on the plane. We use a Hough transform to rotate the projected images such that vertical lines are pointing directly upwards. This proves to be effective for indoor modeling, since many indoor scenes contain strong vertical lines from features such as doors, wall panels, or rectangular frames. Figure 5(a) shows the plane after basic projection and straightening.

3.2. Correcting Image Placements

Next, we proceed from left to right finding corresponding points between pairs of nearby images using SIFT matches [5]. An illustration of this is given in Figure 4. The SIFT matches allow us to determine the x and y distances between two images on the plane. These distances are later used to improve the image locations by aligning neighboring images.

Since SIFT matches frequently include outliers, the RANSAC framework [6] is used for a robust fit. The RANSAC framework requires two functions to be specified: the fitting function and the distance function. These functions are called for random subsets of the SIFT matches until the best set of inliers is found. The fitting function simply finds the average distance between matches. If the matches are exactly correct and the image is frontal and planar then the distances for various SIFT feature matches should be the same. The distance function for a pair of points is the difference between the actual SIFT match distance and the average distance computed by the fitting function. If the distance between a pair of SIFT matches is far away from the average distance of all SIFT matches, then this match is labeled as an outlier. This threshold can be somewhere around 10 pixels.

There are a total of M^2 possible pairs of images, however we can only measure distances between images that overlap at SIFT feature points. Given these distances and the original image location estimates, we solve a weighted least squares problem to estimate the correct location of the images on the plane. Observations that equate the unknown correct image locations to the original location estimates are given a very weight i.e. 0.01, while the observations that specify the distances between images are given a larger weight i.e. 1. The output of the weighted least squares routine is a set of image locations that can best honor all measured distances between the images and provide a more accurate looking result. Figure 5(b) shows the example plane after the image placements are corrected.

3.3. Image Subsampling

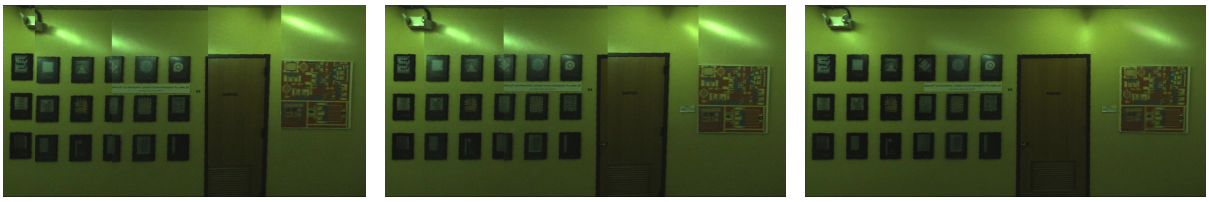
The backpack system produces images at a much faster rate than is necessary to fully texture map all surfaces. Many of these images can be discarded because the range they cover is also covered by other images in the set. Furthermore, it can be advantageous to throw away redundant images at this stage because fewer images generate fewer seams and therefore fewer potential visual artifacts from misalignment. We discard as many images as possible while still covering the entire plane. Figure 5(c) shows an example of a textured plane after location correction and subsampling.

3.4. Blending

The steps above eliminate visual artifacts from misalignment, but there still are obvious discontinuities in brightness between images resulting from differences in lighting. This problem has been successfully addressed in previous work using an alpha blending technique in which the pixel intensities in overlapping images are blended proportional to the pixel's location in the image [1]. We use a similar technique to improve the visual quality of our results. In the small area where images overlap, there is a linear weighting function in the horizontal direction that interpolates between the left image and the right, providing a gradual transition between images. The final location corrected, subsampled, and blended image is shown in Figure 5(c).

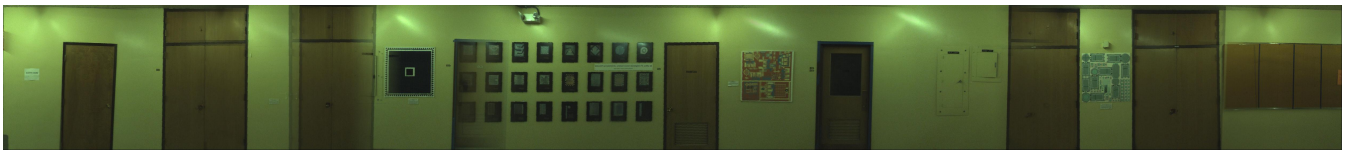
4. RESULTS AND CONCLUSIONS

Figure 6 shows the result of our approach on several hallways consisting of 21, 47, and 37 images respectively. The images were generated using our human operated backpack system for 3D indoor modeling [1]. Even after we processed these



(a) Naive mapping (b) After straightening and correcting locations (c) Final result with blending and subsampling

Fig. 5. These images demonstrate the effect of each step in our approach on a small portion of a long hallway.



(a)

(b)

(c)

Fig. 6. The final results of three planes texture mapped using this approach

images some errors remained. For example, there is a large blurry area on the left side of Figure 6(a) where a door is mistakenly covered by a portion of the nearby wall. Very few SIFT features could be automatically matched in this region so the algorithm left the images in their default incorrect locations determined by the localization algorithms. In Figure 6(b) there are obvious errors near the doors that were recessed slightly from the wall. These errors happen because the wall is not completely planar, which is an assumption we make in our approach. In all the images there are glares that move depending on the location of the camera. This change in illumination leads to inconsistencies across images.

In conclusion, the power of this approach seems to be its simplicity. After the images are projected to the plane it is just a matter of rotating them and shifting them in two dimensions. In contrast, the graph-based approaches refine the camera matrices, which have six degrees of freedom and are susceptible to errors in these dimensions as well. In traditional image stitching, the images are transformed repeatedly using homography matrices which can slowly degrade the quality of the images, especially over long chains. In contrast, we modify each image only once when it is projected to the plane.

5. REFERENCES

[1] T. Liu, M. Carlberg, G. Chen, J. Chen, J. Kua, and A. Zakhor, "Indoor localization and visualization using a human-operated backpack system," in *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*. IEEE, 2010, pp. 1–10.

[2] M. Brown and D.G. Lowe, "Automatic panoramic image stitching using invariant features," *International Journal of Computer Vision*, vol. 74, no. 1, pp. 59–73, 2007.

[3] G. Chen, J. Kua, S. Shum, N. Naikal, M. Carlberg, and A. Zakhor, "Indoor localization algorithms for a human-operated backpack system," in *Int. Symp. on 3D Data, Processing, Visualization and Transmission (3DPVT)*. Citeseer, 2010.

[4] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*, vol. 2, Cambridge Univ Press, 2000.

[5] D.G. Lowe, "Object recognition from local scale-invariant features," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*. Ieee, 1999, vol. 2, pp. 1150–1157.

[6] M.A. Fischler and R.C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.