ENHANCED IMAGE-BASED 3D RECONSTRUCTION
WITH 2D-TO-3D CONSTRAINTS

BY

MICHAEL SITTIG

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2015

Urbana, Illinois

Adviser:

Professor Mani Golparvar Fard
ABSTRACT

This paper discusses an improvement to the Mesh-Assisted Structure from Motion algorithm. Mesh-Assisted Structure from Motion is an improvement over the original Structure from Motion algorithm, which generates a 3D reconstruction of a scene based on a set of 2D images depicting the scene. Mesh-Assisted Structure from Motion improves on it by incorporating a mesh. A user registers one of the images in the image set to the mesh and provides this registration as a constraint to the Structure from Motion pipeline. This solves some difficulties the default Structure from Motion algorithm has, and results in higher reconstruction quality.

The improvement discussed in this paper allows additional anchor images to be registered to the mesh and provided as constraints to the pipeline. Using additional anchor images results in fewer images needing to be registered during pipeline execution. This means that more of the pipeline may be executed automatically, with no additional user input.

Depending on the available images and mesh, using additional anchor cameras may or may not improve reconstruction quality. For most data sets, additional anchor images result in a slightly worse reconstruction. However, for some data sets, the reconstruction quality may be improved by using more anchor images. The most important criteria for determining if using more anchor images will improve or harm the reconstruction accuracy is the accuracy of the input mesh. However, even when the reconstruction becomes less accurate, using more anchor images will still decrease the amount of input necessary during execution, and thus allow more of the pipeline to be executed without needing to wait for user input.

Finally, several possible improvements to the modified Mesh-Assisted Structure from Motion algorithm are discussed. These improvements mostly relate to improving the anchor image selection process, to choose images which will result in a more accurate reconstruction. These improvements would increase the number of situations in which using the modified algorithm with multiple anchor images is superior to the algorithm with only a single anchor image.
# TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION……………………………………………………………….. 1

CHAPTER 2 RELATED WORK………………………………………………………… 3

CHAPTER 3 RESEARCH OBJECTIVE……………………………………………… 7

CHAPTER 4 METHODOLOGY……………………………………………………… 13

CHAPTER 5 RESULTS………………………………………………………………… 29

CHAPTER 6 CONCLUSION AND FUTURE WORK…………………………….. 40

REFERENCES……………………………………………………………………………… 44
CHAPTER 1

INTRODUCTION

Structure from Motion (SfM) is a technique which converts a set of images of a scene into a three-dimensional reconstruction of that scene. It functions by matching features across images and triangulating those points to generate a point cloud. Mesh-Assisted Structure from Motion (mesh-assisted SfM) is an algorithm which improves upon the SfM reconstruction process by incorporating a mesh representing the scene and some amount of user input. By aligning several images with the model and using those alignments as a constraint, it is possible to improve the accuracy of the reconstruction.

In this paper, I discuss a potential improvement to the mesh-assisted SfM algorithm. The traditional mesh-assisted SfM selects a single anchor image to be registered with respect to the input mesh, and registers additional images when doing so becomes necessary. This means that the mesh-assisted SfM pipeline will halt multiple times during execution to accept additional user input. In contrast, my improvement picks several anchor images. By registering multiple images prior to the reconstruction process, it is possible to reduce the number of images that must be registered during the process. This may result in a superior pipeline. In this paper, I discuss the process by which the new anchor images are selected, other changes that are made to the pipeline to support these images, and how the new pipeline’s performance differs from the default mesh-assisted SfM pipeline’s performance. I also discusses the reasons behind the differences in performance and in which situations the
new algorithm is preferable to the old algorithm. Additionally, I discuss how to identify and create datasets which may be more accurately reconstructed using additional anchor images.
CHAPTER 2

RELATED WORK

The process of turning a series of images into a 3D model is an area which has already received a considerable amount of focus in the fields of civil engineering, computer vision, and computer graphics. D4AR modeling\(^1\) is a method closely related to the methods used in this paper. It uses an unordered collection of photographs of a site to generate a point cloud using Structure from Motion (SfM)\(^2\) and Multi-View Stereo\(^3\). This point cloud may then be aligned with a CAD model, allowing the CAD model to be superimposed on top of the input images.

New methods improving upon SfM have also been developed. This paper uses an incremental SfM algorithm, which iteratively registers new images to improve the reconstructed point cloud. However, there also exists the approach of global SfM, which works on all images and data points simultaneously. Recent developments show that the global methodology may be more efficient than an incremental approach\(^4\).

There has also been considerable focus in the area of aligning models with images. Success has been reported with the usage of semi-automated systems for registering 3D models with time-lapsed videos\(^5\). There has also been success with the usage of tags for visualizing 3D models in augmented reality\(^6\), which is another way which a model may be aligned with a camera.

---

\(^1\) Golparvar-Fard et al. 2011
\(^2\) Snavely et al. 2008
\(^3\) Furukawa and Ponce 2010
\(^4\) Crandall et al. 2011
\(^5\) Golparvar-Fard et al. 2009, Kahkonen et al. 2007
There have been several methods developed to automatically align models with photographs or paintings. However, these methods are difficult to use in a construction context for several reasons - construction sites rarely match exactly with the architectural model, the architectural model may not have sufficient detail, and unexpected features such as people or equipment may be present in the provided images.

2.1 ConstructAide

The system most closely related to this work is the ConstructAide system, which is a semi-automated structure-from-motion pipeline⁷. The pipeline takes in a series of unordered images and uses them to construct a point cloud using a modified structure from motion algorithm. A traditional structure from motion algorithm is entirely automatic, and registers individual images by utilizing the features present in each image. These features may be matched to tracks which are already present in the reconstructed scene, allowing the new image to be registered by solving the Pick-N-Points (PnP) problem by using the correspondences between 2D points in the image and 3D points for each matched track. The ConstructAide system additionally incorporates the ability to ask for user input by having a user manually align an image with a 3D mesh. This allows images to be registered even if they do not have enough tracks to be registered automatically.

⁷ Karsch et al. 2014
2.2 Current challenges

There exist several challenges with state-of-the-art systems for Structure from Motion. Firstly, there is no guarantee of the completeness of the scene. If there are no images available of part of the scene, which may happen if an area is obscured or otherwise ignored, then it is not possible to reconstruct that area solely from the input images.

The second challenge is that reconstruction is projective. This means that the projections of each angle into the images in which they are visible will be accurate. However, there are many possible angles in 3D space that could generate a given 2D angle - the same angle could be acute, obtuse, or right depending on the viewing angle. Thus, angles, and especially right angles, are not guaranteed to be preserved in reconstruction.

The third challenge is that reconstructions are unitless. The reconstruction will be accurate up to some scale, but it is not possible to know what that scale would be. If one wished to compare the size of two reconstructions, it would not be possible to do so without knowing the scale of each model, which cannot be calculated as an intrinsic value.

One option to solve these challenges is to utilize a model to assist the registration process. If a model is available, then it is possible to detect which portions of a scene may be missing in a reconstruction, and what ought to be in the missing area. Additionally, having a model makes it possible to add another constraint to a reconstruction, enforcing that angles are preserved by constraining the reconstruction to
approximate the input model. Finally, using a model as input makes it possible to give a scale to the reconstruction because the reconstruction and the model will have the same scale.
CHAPTER 3

RESEARCH OBJECTIVE

Structure from motion is a good algorithm for reconstructing scenes and objects from images. However, it also possesses several weaknesses which my research focused upon addressing.

One weakness inherent in SfM is that it is unable to guarantee scene completeness. This is due to SfM relying on inputted images. If there are no images which show what is present in some portion of the scene, then SfM will not be able to reconstruct that portion of the scene. This is because SfM is only able to reconstruct things based on the points present in its input images. It is unable to infer any data for portions of the scene not present in the images. However, this does not mean that it is inherently impossible reconstruct scenes without visual samples of their content. For example, the Patchmatch algorithm is capable of filling holes in two-dimensional images by reusing other portions of the scene\(^8\). In theory, it would be possible to apply this to three-dimensional images as well, by filling in repetitive but missing structures. This would allow for a more robust SfM algorithm which would be able to have more complete scenes.

A corollary of SfM’s inability to guarantee scene completeness is its difficulty incorporating irregular images. If most of the images are of some portion of the scene and have no overlap with another set of images, it will be difficult to combine both sets of images. This means that although there may be sufficient data to reconstruct an

\(^8\) Barnes et al. 2009
entire scene using SfM, the algorithm is not capable of combining every part of the scene into a single dataset. This challenge is one problem which is addressed by mesh-assisted SfM. Although unrelated segments of the scene may not have enough overlap to combine into a single scene using the images alone, it is possible to combine them using an additional data source. Disjoint scene segments may be combined by using a model's coordinate system, and representing each individual subset in that coordinate system. Although there may not be enough overlap between the different scene segments, the mesh provides a common base which every individual segment may be aligned with. However, combining disjoint sets of images using a mesh introduces new difficulties. If one wishes to combine multiple disjoint subsets of images, it must be determined which images are actually present in each subset. The mesh-assisted SfM process used by Constructaide solves this problem by adding disjoint images individually as they are encountered. However, it would be better if disjoint subsets could be identified earlier, so that entire disjoint scenes could be combined instead of just disjoint images.

In addition to the weaknesses of SfM, there are also several further limitations introduced by mesh-assisted SfM. Firstly, mesh-assisted SfM is dependent on human input for the initial alignment of the input mesh and the coordinate system of the cameras taking the images. In other words, the algorithm begins by having a user manually align and register a single anchor camera and a mesh. After the anchor image is registered, other images are incrementally registered and and aligned based on their similarities to already registered images. This introduces a new variable to the SfM
process, which now becomes partially dependent on the initial choice of the anchor camera. Depending on that first image, other images may be either easier or more difficult to register. The most optimal image would be whichever results in the most accurate scene reconstruction, but this may not be determined without actually going through the entire reconstruction process. Going through the entire process is obviously not a feasible strategy for determining the best anchor camera, so some heuristic must be used to determine the anchor camera. Improving this heuristic means improving the mesh-assisted SfM algorithm.

Another weakness of the mesh-assisted SfM algorithm is that it is difficult to determine exactly which images will need to be registered using the mesh. It is inconvenient for a human using the system to be required to manually register many images. This inconvenience is amplified if instead of aligning every required image prior to execution, the human needs to wait for the algorithm to partially execute and incrementally identify images to register. The mesh-assisted SfM algorithm used by the Constructaide system uses the latter methodology. Thus, there are two weaknesses present in the current mesh-assisted SfM image registration process. The process would be improved if every image which will require registration were to be identified prior to execution. Additionally, the process would be improved if the number of images requiring human intervention could be minimized.

One last weakness of the mesh-assisted SfM algorithm is its assumption of an accurate mesh. In practice, scenes do not perfectly correspond to the mesh they are based on. There may be additional obstacles present in the scene, such as vehicles or
people, which will not be present in a mesh. Alternatively, the mesh may have components which are not present in the actual scene, such as if a piece of construction has not been completed. Even when all components are present in both the scene and the mesh, it is still possible for there to be minor variations between the scene and mesh, which result in the registrations being less accurate. The algorithm is generally robust to minor variations, but major variations, such as extra obstacles or missing components, can make image registration significantly more difficult.

For my research, I focused primarily on improving anchor image selection, which also provided solutions to several of the other problems.

3.1 Research Question

The problem I wished to solve for my research was the problem of choosing optimal anchor images. Improving the anchor image selection will result in the mesh-assisted SfM algorithm having better performance. There are several reasons why this will be the case.

Firstly, the initial anchor camera has an effect on the quality of the reconstruction, and the amount of additional input required for the mesh-assisted SfM algorithm. If the chosen anchor image is different from every other image, then registering it will not provide any useful information, because the anchor image’s registration will not help register any other images. For the mesh-assisted SfM algorithm, it is possible to register multiple disjoint subsets of images using the mesh, so the algorithm will still function even if the initial anchor camera can not be used to register every image. However, it is
still preferable to register as many images as possible, so that the reconstruction will utilize the maximum amount of data and be more accurate.

Another reason why choosing a good anchor image is important is that it is possible for the initial anchor camera to be difficult for the user to register. For example, it may be the case that the pictured scene is different from the mesh, or it may lack distinctive features that may easily be matched to their corresponding points on the mesh. If the anchor image is difficult to register, then the registration will likely be less accurate, which in turn results in the final reconstruction being less accurate. Therefore, when the anchor image is poorly registered, it should be possible to mitigate the impact of that poor registration.

A final reason why choosing the optimal anchor image is important is because choosing the anchor image properly will decrease the number of images which must be registered by a user. If the number of images may not be decreased, then it is instead preferable to register as many of the necessary images prior to execution. When there are multiple disjoint subsets of images, one image from each subset will need to be registered. Therefore, if the subsets may be identified prior to execution, then an image from each subset may be registered prior to execution, which will allow all subsets to be registered and used in the reconstruction process. This will decrease the number of necessary manual registrations during execution of the mesh-assisted SfM algorithm.

The solution that I implemented to solve the anchor image problem was to add a new step to the mesh-assisted SfM pipeline to identify a minimal subset of images that should allow every other image in the dataset to be registered. This addresses all three
motivations for improving the anchor image selection. Firstly, if an image from each disjoint subset of images is used as an anchor image, then there will not be the problem of needing to combine subsets later in the pipeline. Every subset will already be incorporated into the reconstruction, and the reconstruction will be more accurate because images from multiple points of view are used to create it. Secondly, if multiple anchor images are used, the impact of any one of them being improperly registered will be diminished. When there are multiple registrations provided, any errors in those registrations will likely be canceled out by errors in another registration. This means that the reconstruction formed using multiple images should be more accurate than the reconstruction generated by using a single image. Finally, identifying the different subsets of images prior to execution means that combining them may be done without requiring additional input during execution. This means that less input is required for the mesh-assisted SfM pipeline, which makes it more convenient to use because more of it is done automatically. Even if some additional input is still necessary, using multiple unique anchor images will result in more other images being possible to register. For these reasons, identifying and registering multiple anchor images from disjoint subsets is a good solution to the anchor image problem.
CHAPTER 4

METHODOLOGY

There are multiple steps in the augmented Mesh Assisted Structure from Motion algorithm. Firstly, the input images are preprocessed to determine how similar each image is. Secondly, unique images are identified and pre-registered. Thirdly, the pre-registered images are provided as input to the remainder of the mesh-assisted SfM pipeline, which is then able to reconstruct the entire scene.

4.1 Preprocessing

The preprocessing step calculates a similarity score between every pair of images. By identifying which images are similar, it is possible to cluster images into subsets, with similar images being in the same subsets. These subsets will have the property that if one image in the subset is registered, it should be possible to register every other image in the subset.

The similarity score for a pair of images is a measure of how accurately one image may be transformed into the other using a homography transform. Because the SfM algorithm is capable of using homography transforms to incrementally register images, if two images have a high similarity score, then SfM will be able to register one image given the other’s registration.

To calculate the similarity score, the first step is to calculate the SIFT features for each image. These features correspond to identifiable key points, meaning that they may be tracked and matched between images. The next step is to perform this
matching. For every pair of images, it will be determined which features are present in both images, and which features are matched with which. If two images have many overlapping features, then they will likely contain similar image content. For the algorithm implemented in this paper, VisualSFM was used to calculate feature points and matches.

After the features are matched and identified, a homography transform is calculated. This transform is calculated in a RANSAC loop. For each step of the loop, a random subset of features are selected. Next, a transform matrix is calculated that will allow one image to be transformed into the image plane of the second image. This means that every feature may be transformed into the new image plane. Following this, every feature point is compared with its corresponding feature point in the other image. If the two points are sufficiently near to each other, they are declared to be an inlier. Otherwise, the points will be declared outliers. The best homography transform is thus the matrix which results in the highest percentage of inliers. The percentage of inliers is tracked, and is ultimately used as the similarity score between the two images. If two images are similar, then there will be many inliers because there will be very little distortion between the two. On the other hand, if the inlier score is very low, then that means there exists no homography transform matrix which will accurately transform one image into the other, which means that the two images will be members of different image clusters.

The similarity score algorithm may be found as algorithm 4.1.
4.2 Identification

After the similarity scores are calculated, the next step is to identify which images should be used as anchor images. The goal is to select one anchor image from each disjoint subset of images, so that every image may be registered using some anchor image. To calculate which images belong in which subset, and which image should be the representative of those subsets, the first step is to threshold all similarity scores to mark the image pairs as either ‘similar’ or ‘dissimilar’. A higher threshold results a more accurate reconstruction. On the other hand, using a low threshold will require fewer anchor cameras. For this paper, the threshold chosen was 80%, to match the threshold later used for registering images using a homography transform.

After similarity scores are thresholded, each image is ranked depending on how many other images they are similar to. The image which is similar to the largest number of other images is used to seed an image cluster. Images which are similar to the first image are added to the cluster, followed by the images similar to those images. This process repeats until no new images are added to the cluster, at which point one image is chosen from the cluster to serve as its representative anchor camera. For this paper, the initial seed image was chosen, because it results in the maximum number of images which may be registered using a single homography. However, it may be the case that some other criteria would be superior, such as identifying the image that results in the lowest average distortion across the entire image cluster. In practice, there did not seem to be any significant problems with using the seed camera.
After an image cluster is identified, the images in that cluster are removed from
the image pool and the process repeats until every image is assigned to a cluster, and
every cluster is assigned a representative to act as its anchor camera. The algorithm is
summarized as algorithm 4.2.

4.3 Mesh-Assisted Structure From Motion

After anchor images are selected, the anchor images are used as input to the
mesh-assisted SfM pipeline. The pipeline begins by having a user manually register
every anchor image. The user selects several 2D points in one of the images and their
corresponding 3D points on the model. Given a minimum of 4 corresponding 2D and 3D
points, it is possible to calculate the extrinsic parameters - translation and rotation - of
the camera that generated the image by finding the solution to the Perspective-n-Points
problem, or PnP. In addition to this, we may infer the intrinsic camera parameters by
extracting the focal length of the camera from the image’s EXIF tag, and assuming that
there is no camera distortion. This registration process allows all camera parameters to
be defined, in the form $C = P^*[R|t]$, where $P$ is the camera’s 3x3 intrinsic parameter
matrix, $R$ is a 3x3 rotation matrix, and $t$ is a 3x1 translation vector.

After camera parameters are calculated for all of the anchor images and the
images are marked as being registered, the remaining unregistered images are
iteratively registered. Images are registered in one of three ways. The first way is to
register using a homography. If an unregistered image is similar to a registered image,
defined as an inlier percentage of at least 80% for the homography transform, then the
image may be registered using a homography. The homography matrix is applied to 2D points in the registered image to obtain their locations in the unregistered image. Following this, PnP may be performed using the new 2D points and their matching 3D points to solve for the camera parameters of the unregistered image. Alternatively, the unregistered image may be registered using tracks. As cameras are registered, features may be triangulated and transformed into 3D coordinates. If an image has enough features which correspond to calculated 3D points, the PnP may be performed using the 2D and calculated 3D points to obtain the camera matrix. The third way to register a new image is to query the user for additional input, registering the camera as described above. Following each new camera registration, bundle adjustment is performed to calculate new tracks, update existing tracks, and update existing camera parameters.

The algorithm for mesh-assisted SfM is summarized as algorithm 4.3.
4.4 Algorithms and Figures

Algorithm 4.1: Similarity Score Calculating

Let a and b be a pair of images.
Let \( f_a \) and \( f_b \) be a list of features in image a and image b.
perform matching to obtain matched features \( f_{ma} \) and \( f_{mb} \).
compute homography matrix between a and b using RANSAC:
  randomly select 4 matched points, \( f_{Ra}(1...4) \) and \( f_{Rb}(1...4) \)
calculate homography matrix \( H \) such that \( f'_{Ra} = Hf_{Rb} \)
for each point \( f_{ma}(i) \)
  if \( ||f_{ma}'(i) - Hf_{mb}(i)|| < \) maximum allowed error
    mark i as an inlier
perform least squares fit on all inliers to obtain new matrix \( H' \)
  keep \( H' \) matrix which has highest percentage of inliers
set score \( S_{ab} \) to be the percentage of inliers for the best homography
Algorithm 4.2: Anchor Image Selection

Let $S_{ab}$ be the score between image $a$ and image $b$. Let $S_{\text{min}}$ be the minimum threshold for two images to be considered similar.
Let $R_i$ be whether an image has been marked (initially false).
Let $A_i$ be whether an image is an anchor image (initially false).

while there are images for which $R_i = \text{false}$:
    select the image $i$ for which $R_i = \text{false}$ and $\text{count}(S_{ij} > S_{\text{min}})$ is maximized
    set $A_i = \text{true}$
    set $R_i = \text{true}$

while exists image pair $j$, $k$ for which $R_j = \text{true}$, $R_k = \text{false}$, and $S_{jk} > S_{\text{min}}$:
    set $R_k = \text{true}$
Algorithm 4.3: Mesh-Assisted Structure From Motion

Let $R$ be a set of registered images.
Let $U$ be the set of unregistered images.
compute feature matches between all images.
while there are still unregistered images:
    for each registered image $r$ in $R$ and unregistered image $u$ in $U$:
        if there is a valid homography between $r$ and $u$:
            register $u$ using a homography
        if no images were registered using a homography:
            identify unregistered image $u'$ with the most tracks
            if $u'$ has enough tracks:
                register $u'$ using tracks
            else:
                identify unregistered image $u''$ with fewest tracks
                manually register $u''$
        perform bundle adjustment on all registered images
In this dataset, some image pairs (1/2, 1/4, 4/5, and 3/6) are marked as similar. The algorithm begins by marking image 1 as an anchor image, because it is similar to the most number of other images. Next, images 2 and 4 are added to its cluster. Next, image 5 is added to the cluster. No more images are similar to any of the images in the cluster, so image 3 is selected to be a new anchor image. It is similar to image 6, so the cluster is expanded. At this point, all images either are an anchor image, or are assigned an anchor image to represent them, so the algorithm halts and returns that images 1 and 3 are anchor images.
Figure 4.2: Threshold effects

Suppose we have some set of images to cluster. Each image will have some similarity score expressing how similar it is to other images (expressed here as proximity). If a good threshold is chosen, images will be properly clustered. If the threshold is too low, images will be clustered very easily, and fewer anchor images will be chosen than necessary. If the threshold is too high, more clusters will be created, and there will be too many anchor images. For this paper, a threshold of 80% was used.
Figure 4.3: Sample clusters

Four images from the Northwest dataset. Three of the images are from similar points of view, and thus will all be put in the same cluster, for which one of them will be an anchor image. The fourth image is different, and is thus is an anchor image for its own set.
Figure 4.4: Mesh-Assisted Structure From Motion, part 1

The mesh-assisted SfM algorithm begins by presenting the user with the model’s mesh over one of the selected anchor images. The user may rotate and move around the model to get a better view.
After the user identifies which features they wish to use, they may select features by clicking on the model.
Once features are selected, the user may switch to 2D mode and select the 2D features which correspond to the selected 3D features.
As soon as four features are matched, the model will be automatically registered. If the user is happy with the registration, they may save it. Otherwise, they may add additional features or remove inaccurate features to obtain a more accurate camera matrix.

This process is repeated with all of the chosen anchor cameras.
After all anchor cameras are registered, they are fed into the main SfM pipeline, which registers other cameras using homographies, tracks, or additional manual input. At the end, the entire set of images will be registered, and the user will be provided with camera matrices for each camera.

The cameras are visualized here as blue dots, with smaller dots showing the camera frustrum in the direction which the camera is pointing. For this dataset, the cameras are distributed along the bottom side of the building, facing towards the top.
CHAPTER 5

RESULTS

My hypothesis was that using the modified mesh-assisted SfM pipeline with improved anchor camera selection would improve the pipeline’s accuracy in reconstructing scenes. To test the modified pipeline, experiments were performed using four datasets of varying size. All datasets were of construction sites in varying stages of completion. Construction sites were chosen because they are a realistic case in which the pipeline would be used. Additionally, the construction sites all had mesh models available, and provided a diverse set of scene compositions to test.

To evaluate the pipeline, mesh-assisted SfM was performed on each dataset, once with a single arbitrarily chosen anchor camera and following the default mesh-assisted SfM algorithm, and once using multiple anchor cameras chosen as described above. The pipeline then generated a set of camera matrices for each image. These camera matrices were then compared to ground truth matrices to determine three error metrics: rotational error (difference between viewing directions), translational error (difference in camera location), and reprojection error (difference in location of the reprojection of several prechosen points). For context, the results from three additional algorithms used in the original mesh-assisted SfM paper are also shown: VisualSfM (VSfM), Photosynth (PS), and a results generated by matching Photosynth output to the input mesh using an iterative closest point algorithm (PS-ICP).
5.1 Discussion

Compared to the standard mesh-assisted SfM pipeline, the modified pipeline with additional anchor cameras performed slightly worse. For almost every image set, the modified pipeline had a higher rotational error, higher translational error, and higher reprojection error. This suggests that for most data sets, using additional anchor cameras will lead to worse performance. The modified pipeline did still generally have better performance than the algorithms that were not mesh-assisted SfM, which suggests that mesh-assisted SfM is still a good algorithm even when additional anchor images are used.

The one exception to the trend of the modified algorithm performing worse than the unmodified algorithm was the Northwest dataset. For this dataset, the translational and reprojection error actually went down. This suggests that for some datasets, using additional anchor cameras can lead to better performance. Identifying the criteria that lead to a superior reconstruction may allow the modified algorithm using an increased number of anchor cameras to have better performance.

Overall, when compared to the algorithm which used a single anchor image, using multiple anchor images raised the amount of rotational error, but had almost no change for reprojection error or translational error. This suggests that the cameras calculated by both algorithms are very similar, but using multiple anchor cameras will slightly increase rotational error.

One statistic which was not tracked was the number of additional cameras which needed to be registered by the user during execution. For the unmodified algorithm,
datasets were generally registered using two to four additional cameras. For the modified algorithm, there were generally zero to two additional. In every case, the modified algorithm required more images total to be registered, but fewer images were registered during execution. This suggests that providing additional anchor cameras will decrease the number of cameras which must be registered later, but not at a one-to-one ratio.
## 5.2 Figures and Tables

### Table 5.1: Results

<table>
<thead>
<tr>
<th>Name</th>
<th>Algorithm</th>
<th>Images</th>
<th>Anchor Images</th>
<th>Rotational Error</th>
<th>Translation Error</th>
<th>Reprojection Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Northwest</td>
<td>Single</td>
<td>15</td>
<td>1</td>
<td>0.41</td>
<td>0.92</td>
<td>3.93</td>
</tr>
<tr>
<td></td>
<td>Multi</td>
<td>12</td>
<td></td>
<td>0.83</td>
<td><strong>0.38</strong></td>
<td>1.45</td>
</tr>
<tr>
<td></td>
<td>VSfM</td>
<td></td>
<td></td>
<td>2.28</td>
<td>2.51</td>
<td>22.70</td>
</tr>
<tr>
<td></td>
<td>PS</td>
<td></td>
<td></td>
<td>8.79</td>
<td>6.99</td>
<td>23.26</td>
</tr>
<tr>
<td></td>
<td>PS-ICP</td>
<td></td>
<td></td>
<td>79.40</td>
<td>10.19</td>
<td>52.96</td>
</tr>
<tr>
<td>West</td>
<td>Single</td>
<td>26</td>
<td>1</td>
<td><strong>0.99</strong></td>
<td><strong>0.69</strong></td>
<td>3.27</td>
</tr>
<tr>
<td></td>
<td>Multi</td>
<td>7</td>
<td></td>
<td>3.94</td>
<td>0.76</td>
<td>3.33</td>
</tr>
<tr>
<td></td>
<td>VSfM</td>
<td></td>
<td></td>
<td>1.81</td>
<td>0.53</td>
<td><strong>1.96</strong></td>
</tr>
<tr>
<td></td>
<td>PS</td>
<td></td>
<td></td>
<td>1.67</td>
<td>1.16</td>
<td>3.32</td>
</tr>
<tr>
<td></td>
<td>PS-ICP</td>
<td></td>
<td></td>
<td>20.02</td>
<td>1.97</td>
<td>20.25</td>
</tr>
<tr>
<td>Northeast</td>
<td>Single</td>
<td>22</td>
<td>1</td>
<td><strong>0.49</strong></td>
<td><strong>0.68</strong></td>
<td><strong>2.11</strong></td>
</tr>
<tr>
<td></td>
<td>Multi</td>
<td>3</td>
<td></td>
<td>1.04</td>
<td>1.08</td>
<td>4.14</td>
</tr>
<tr>
<td></td>
<td>VSfM</td>
<td></td>
<td></td>
<td>1.23</td>
<td>1.34</td>
<td>3.54</td>
</tr>
<tr>
<td></td>
<td>PS</td>
<td></td>
<td></td>
<td>1.21</td>
<td>1.14</td>
<td>3.12</td>
</tr>
<tr>
<td></td>
<td>PS-ICP</td>
<td></td>
<td></td>
<td>6.22</td>
<td>9.08</td>
<td>17.65</td>
</tr>
<tr>
<td>Basement</td>
<td>Single</td>
<td>10</td>
<td>1</td>
<td><strong>0.00</strong></td>
<td><strong>0.00</strong></td>
<td><strong>0.53</strong></td>
</tr>
<tr>
<td></td>
<td>Multi</td>
<td>9</td>
<td></td>
<td>0.49</td>
<td>0.11</td>
<td>0.77</td>
</tr>
<tr>
<td></td>
<td>VSfM</td>
<td></td>
<td></td>
<td>137.90</td>
<td>8.15</td>
<td>45.61</td>
</tr>
<tr>
<td></td>
<td>PS</td>
<td></td>
<td></td>
<td>12.45</td>
<td>1.56</td>
<td>8.22</td>
</tr>
<tr>
<td></td>
<td>PS-ICP</td>
<td></td>
<td></td>
<td>3.29</td>
<td>1.55</td>
<td>9.67</td>
</tr>
</tbody>
</table>
### Table 5.2: Comparison of Results

<table>
<thead>
<tr>
<th>algorithm</th>
<th>mean rotational error</th>
<th>mean translational error</th>
<th>mean reprojection error</th>
</tr>
</thead>
<tbody>
<tr>
<td>single</td>
<td>0.47</td>
<td>0.57</td>
<td>2.46</td>
</tr>
<tr>
<td>multi</td>
<td>1.57</td>
<td>0.58</td>
<td>2.44</td>
</tr>
<tr>
<td>VSfM</td>
<td>35.80</td>
<td>3.13</td>
<td>18.45</td>
</tr>
<tr>
<td>PS</td>
<td>6.03</td>
<td>2.71</td>
<td>9.48</td>
</tr>
<tr>
<td>PS-ICP</td>
<td>27.23</td>
<td>5.70</td>
<td>25.13</td>
</tr>
<tr>
<td>delta</td>
<td>-1.10</td>
<td>-0.01</td>
<td>0.03</td>
</tr>
</tbody>
</table>

Comparison of results between the five different algorithms. Mean rotational error was measured in degrees, mean translational error is in meters, and mean reprojection error is measured in percentage of image width. Delta is the difference between the mesh-assisted SfM using a single anchor image vs multiple anchor images, which was small but slightly in favor of using a single anchor image.
Northwest was a construction site which had a filled in foundation, and columns were beginning to be put in. There were many easily identifiable pillars and corners, and few obstructions.
West features a building which had approximately two levels finished. There were many inconsistencies between the mesh and the scene, such as scaffolding, railings, and an entire room right in front of most of the cameras which was not yet finished.
Northeast was of a basement area still under construction. The model was sparse, and had many inconsistencies with the scene, while the images had few identifiable features that could easily be matched to the model.
Basement was more images of the basement area, after support structures began to be put in. At this point, the scene structure was more identifiable. The dataset itself was very small, containing only 10 images.
When a single image is used as an anchor camera, many additional images will need to be registered during execution.

Using the modified algorithm, multiple images are selected. Depending on how many anchor images are selected, some number of additional images will need to be manually registered. The more anchor cameras there are at the start, the fewer additional will need to be registered.

The ideal set of anchor images would be sufficient for registering every other image. However, identifying the minimal subset necessary is not doable without actually running the entire pipeline. As a result, the only way to guarantee no additional images need to be registered is to use every image as an anchor image.

Using more anchor cameras will generally increase the total number of cameras that need to be registered, instead of keeping it constant. One way to understand why is to consider the case in which there are some number of cameras which absolutely must
be registered, and some number of other cameras which may be registered based on the necessary cameras. If the anchor image selection process only chose the necessary images, the total number of registered cameras would be constant, because no unnecessary images were registered. However, every unnecessary image that is registered will raise the number of registered images by one. Thus, because we do not know which images are necessary without actually running the pipeline, we can not guarantee that no unnecessary images are registered, which means that every additional image will, on average, raise the number of images registered by some amount. The total number of registered images will be reduced if we can accurately predict necessary images, and minimized if we only pick no unnecessary images.

The reason why it is preferable to have fewer or no images be manually registered during execution is because it allows for more of the algorithm to be run automatically. Every time execution halts to accept user input, the algorithm needs to wait for the user to finish before continuing, which results in wasted execution time. Additionally, it is preferable from a user perspective to not need to attend to the pipeline while it executes. It is the best user experience when it is only necessary to perform input at the start of the pipeline, and to walk away entirely while the pipeline is executing.
CHAPTER 6

CONCLUSION AND FUTURE WORK

The modified mesh-assisted SfM algorithm did not perform very well. The reconstructions were slightly less accurate, and more images needed to be manually registered total. The only way in which using additional anchor images consistently improved performance was in reducing the number of cameras which needed to be registered during execution. Because the pipeline was identical for both algorithms other than using a different number of anchor images, this suggests that using more anchor images will have a small but negative effect on the pipeline’s accuracy.

However, another statistic to consider is that for the two datasets which were almost entirely registered manually by the modified pipeline (Northwest and Basement), the errors were very low - only one of the four pipelines run with a single anchor image performed better than the pipeline which had almost entirely manually registered. This suggests that using many anchor images may be a valid strategy, and that manual registrations can be very accurate. So, the question to answer is: why did using more registrations work in some cases, and not others? Similarly, what can be done to make using additional anchor cameras a superior strategy?

The most likely reason for the decreased performance of the modified pipeline compared to the unmodified pipeline is human error when registering the mesh to the anchor images. For people, registering 2D points to 3D points is not a particularly difficult task. Humans are capable of building a mental model of a scene and matching
the 2D points to 3D points without too much difficulty. However, this task becomes significantly more difficult when the model is inaccurate.

Several types of model inaccuracies made point registration more difficult. The most significant cause of error was lack of model detail. In general, the mesh that is being used will not have an exact correspondence with the scene presented in the images. Either the mesh will have features which are not yet constructed, or the images will have additional obstacles such as scaffolding or people which are not present in the mesh. Both of these errors lead to a difficulty in determining which features in the image correspond to which features in the model. This process is simplified when a model is accurate, but becomes significantly more difficult the less accurate the model is. For example, the dataset Northwest was very accurate, and thus had relatively low error rates for manual registration. On the other hand the mesh for dataset West had many image features which were not present or occluded in the mesh and vice-versa, and thus could not easily be registered.

Another source of error is localization errors. For a human, it is very difficult to estimate distances. As a result, corners are fairly simple to identify and match between the model and the image. However, straight lines are much more difficult to measure. As a result, if a line in a model is longer or shorter than its corresponding line in the image, it is difficult to determine where exactly the end of the line ought to be. This source of error was most pronounced for Northeast, which had relatively few identifiable features in the images.
If the primary cause of the increased error in the modified pipeline compared to the unmodified pipeline were human error, this would actually explain why the Northwest dataset had better performance. The Northwest dataset likely had better performance with more anchor images for two reasons. Firstly, it was highly detailed. There was a strong correspondence between the features present in the image and the features present in the mesh, and there were many easily identifiable points and corners to use as keypoints. For the other models, there was more noise. Either there were extraneous features in the mesh which were not present in the images, or there were additional occlusions in the image which were not present in the mesh, or there was a general lack of identifiable features which could be easily localized. The Basement dataset also had the property of having clearly identifiable features, which may be why it performed well when everything was manually registered.

The second possible reason why the Northwest dataset had better performance with more anchor images is because it was the dataset for which all testing of the pipeline was performed on. As a result, the person performing manual registrations on the anchor images had significantly more practice performing those registrations, and thus was able to register them more accurately. When manual registrations are more accurate than a registration calculated using a homography transform, using more manual registrations will result in better performance.

Overall, this suggests that using more anchor images may be a valid strategy if the provided mesh is very accurate, because when the mesh is an accurate representation of the pictured scene, manual registrations will be more accurate. When
the mesh is easy to register, it is possible to register more images accurately. On the other hand, if the mesh is a poor representation of the scene, it is preferable to register whichever single image is the easiest, and perform the remainder of the pipeline automatically. The ideal rule for choosing anchor cameras may be to register as many anchor images as possible as may be registered accurately, and register all difficult images using homographies.

For future work, it would be worthwhile experimenting with different heuristics for selecting anchor cameras. It is possible that some other rule for selecting anchor images would perform better, such as using a different similarity threshold, or using a different rule for selecting which image should represent an image cluster. Another feature which could be added would be an improved user interface. If there were additional tools to assist in camera alignment and feature localization, that would make it easier for users the manually register images, which would result in more accurate camera matrices. This is especially relevant when many images need to be registered, which can be the case for large datasets.
REFERENCES


