Quantifying the accuracy of dense surface modeling within PhotoModeler Scanner

Eos Systems Inc.
www.photomodeler.com

September 25, 2012

Abstract

In this report, we analyze the accuracy of dense surface modeling within the PhotoModeler Scanner (PMS) software package. Using techniques from photogrammetry and computer vision, PhotoModeler Scanner is able to reconstruct surfaces present in a scene. The quality of the reconstructed surface depends on a) the estimation of the camera parameters and b) the matching accuracy between corresponding points on the scene photographs. The impact of both stages is examined in this report. Accuracy studies are conducted using simulated 3-D scenes, as well as using real-world images of scenes containing benchmark data. The simulated scenes allow testing individual parameters affecting the accuracy of dense surface model creation. The real world scenes are used to emulate a typical user scenario and to provide an accuracy analysis under general (non-ideal) conditions. A maximum accuracy of approximately 1 part in $44,000$ is achieved using perfectly known camera parameters and an artificial planar scene. A drop in accuracy to 1 part in $18,000$ is observed when manual intervention (in the form of photogrammetric targets) is used for camera parameter estimation. The most flexible case of fully automatic camera parameter estimation (with no special targets) results in an accuracy of 1 part in $10,000$. Finally, the accuracy of surfaces generated by PMS is found to have similar accuracy to a state-of-the-art laser scanner. More specifically, PMS outputs point clouds that are accurate to $\pm 0.9mm$ at a $3.5m$ range.

1 Introduction

The classical problem of surface reconstruction from multiple photographs finds varied applications in domains such as visual effects engineering (VFX), surveying, forensic sciences and digital preservation of architectural and archaeological objects. The wide spread availability of high resolution cameras has led to an explosion in various professional and hobbyist applications. The main scope of this article is to establish the usefulness of PhotoModeler Scanner (PMS) as a professional photogrammetry tool with defined accuracy bounds. The algorithms used in PhotoModeler Scanner are based on passive stereo techniques which help in recovering the 3-D structure of a surface. Alternative technologies include active stereo and time of flight sensors[1]. Active stereo can result in highly accurate reconstructions of the scene but they require controlled indoor environments and a higher degree of user intervention. Time of flight sensors have a high cost and can be difficult to transport and setup. These alternative methods have conventionally delivered higher accuracy in obtaining surface reconstructions. However, the availability of very high-resolution imaging sensors and the development of sophisticated algorithms from the computer vision community have pushed the accuracy envelope of passive stereo methods. An important aim of this paper is to re-examine the hierarchy of accuracy claims between the different alternatives to 3-D surface reconstruction.

2 Methodology

The quality of a 3-D reconstruction depends primarily upon (a) the accuracy of the recovered camera parameters involved in a photogrammetric project and (b) the accuracy of the matching between corresponding points. The camera parameters refer to intrinsic parameters (related to the imaging system) and extrinsic parameters (pertaining to the relative configuration of different viewpoints). In particular, intrinsic parameters of a camera refer mainly to the focal length of the camera, the location of the principal point and the radial distortion introduced by the lens (if any). The extrinsic parameters of a camera refer to the pose of the camera in relation to a particular coordinate system. These involve a translation of the camera center and rotational parameters describing the orientation of the camera with respect to the reference coordinate frame. The effects of the camera parameters can be isolated by assuming them to be accurately known. This can be achieved through artificial scenes rendered by a graphics package in which the camera parameters (both internal and external) are manually defined. Such a technique provides a tightly controlled environment to evaluate the effect of various factors on the accuracy of the 3-D reconstruction. Alternatively, if the intrinsic parameters of a camera are not known, they can be estimated to a high degree of accuracy using a calibration procedure. Similarly, the pose of the camera can also be obtained to a high degree of accuracy by establishing correspondences between points in multiple images of the scene. In subsequent discussion, we assume that the intrinsic parameters have been estimated to a high degree of accuracy. We will then use the term "camera

[1] http://en.wikipedia.org/wiki/Time-of-flight_camera

parameters" to refer exclusively to the extrinsic parameters of the camera. These extrinsic parameters are also referred to as the *orientation* or *pose* of the camera.

We propose a series of experiments gradually relaxing constraints on the known parameters required in a full 3-D reconstruction of a scene. The different parts of the experiment are broken down graphically in Fig. 1. The left-hand branch of the methodology diagram in Fig. 1 isolates the effects of the camera parameters from those of the dense matching stage. The experiments in the left-hand branch expose the accuracy of the matching algorithm while factoring out the accuracy of the camera parameters. The camera parameters can be factored out by manually setting them to be known parameters. We use an external graphics rendering package to create an artificial scene with known camera parameters. By using such an approach, a higher degree of control can be imposed on the structure of the scene.

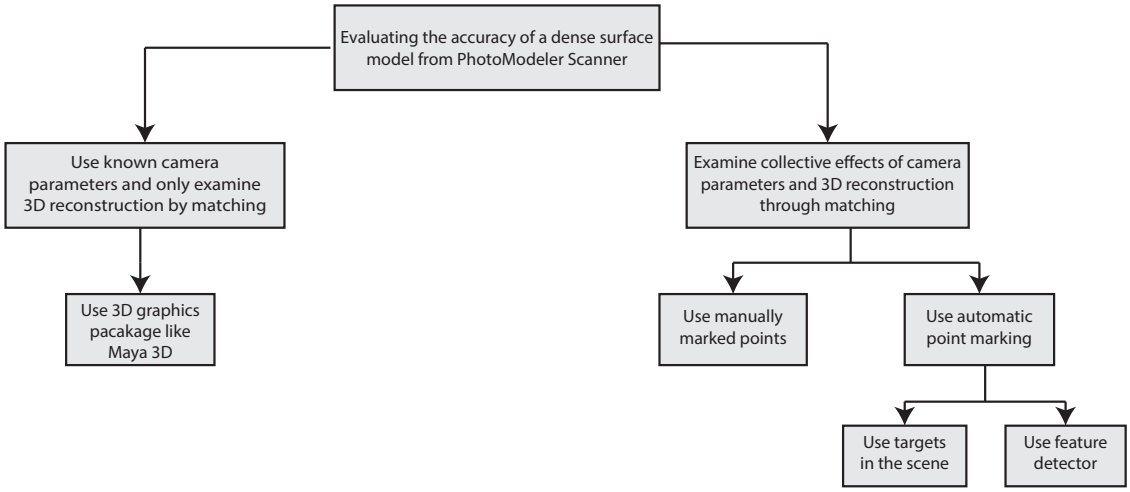


Figure 1: A graphical representation of the experimental methodology followed in this study

The right-hand branch of Fig. 1 shows the different paths that examine the accuracy of a dense surface model (DSM) as a function of both unknown camera parameters and the dense matching required in a multi-view stereo setup. In these experiments, PhotoModeler does not assume the camera parameters to be known. A calibration procedure is run to estimate the intrinsic parameters of the cameras used in the scene. Moreover, a pose estimation procedure is also used to obtain the extrinsic camera parameters. The pose estimation is reliant on finding corresponding points between images of the scene. PhotoModeler can estimate the pose by

1. using manually referenced points on corresponding images (requires user intervention in PhotoModeler Scanner),
2. automatically detecting manually placed targets in the scene (requires user intervention while setting up scene),
3. automatically detecting salient features from the scene (fully automatic).

Each of these situations is investigated in Sec. 3. A subtle assumption in various multi-view stereo methods is the availability of the intrinsic and extrinsic camera parameters. The most general case would be to consider both the intrinsic and extrinsic parameters of the camera as unknown and to estimate these parameters together with the 3-D reconstruction. However, PhotoModeler takes the middle ground and only requires knowledge of the intrinsic parameters of the camera. A convenient calibration capability is included within PhotoModeler in which the user completes a quick, largely automated procedure. This consists of taking photos of a calibration sheet from multiple viewpoints. PhotoModeler is then able to compute the intrinsic camera parameters like the focal length, principal point and radial distortion parameters. In the absence of such a calibration procedure, PhotoModeler also allows a user to use approximate values for the intrinsic parameters (for example, using focal length settings from the EXIF data of a photograph). These values can then be refined using a field calibration procedure in PhotoModeler.

The orientation of the cameras can be solved in an fully automatic manner by PhotoModeler using its *SmartMatch* (SM) feature. The SM capabilities of PhotoModeler include finding unique interest points in images of the scene and robustly matching them to corresponding points in images from different viewpoints. A more user-controlled approach can also be taken by placing photogrammetric targets in the scene. These targets can then either be automatically matched, or referenced manually by hand.

Several factors contribute to the quality of the reconstructed scene. While the algorithms used in the camera orientation and dense matching stage are obvious factors, we are also interested in other factors which can affect the quality of the dense surface model. A few such factors are enumerated in Sec. 2.1 and 2.2.

2.1 External parameters affecting DSM accuracy

This section enumerates several scene, camera and surface related parameters that affect the accuracy of a reconstructed surface using multi-view stereo techniques.

Base-to-Height Ratio: The base-to-height (B/H) ratio is a term that is primarily used in aerial photogrammetry. It is defined as the ratio of the separation between a camera pair and their height above the ground plane. The implicit assumption in computing a B/H ratio is that the camera pair are translated with respect to each other only in one dimension and that they are located at the same height above the ground. In more general multi-view configurations, the B/H ratio can be approximated through the use of various heuristics.

Camera Angle: The camera angle is defined as the angle between the look vectors of each camera in a camera pair. The look vector of a camera is depicted in Fig. 2. It is defined as the unit vector pointing in the direction of the vector between the camera center and the perspective center of the image plane.

Surface Angle: The camera angle does not consider the surface being modeled. The surface angle is defined as the angle between the camera-to-surface vectors at a given point on the surface. The camera-to-surface vector is defined as the vector between the camera center and a point on the surface. In this paper, we generalize the surface by its centroid and compute surface angle with respect to the centroid of the surface. A surface angle can, however, be calculated for every point on the surface. This is also shown in Fig. 2.

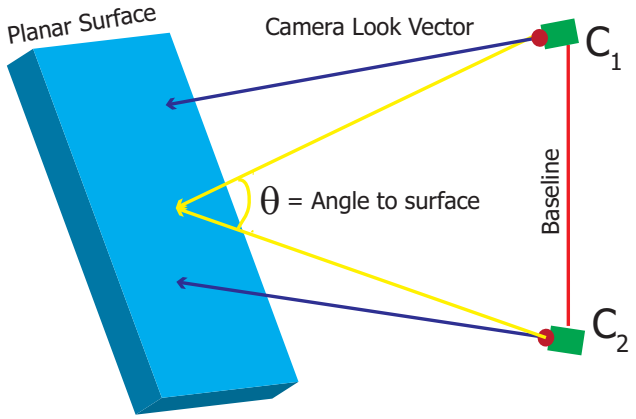


Figure 2: A depiction of the camera look vector and the surface angle

Texture: Correlation between neighborhoods of a pixel forms the basis of the dense matching used in PhotoModeler Scanner. A prerequisite for such correlative matching is the existence of surface texture that is unique enough to discriminate between different pixel neighborhoods. Moreover, the appearance of this texture must remain consistent between different image viewpoints. For example, specular highlights and lighting changes may change the appearance of the texture. As a result, the type of surface texture plays a vital role in the accuracy of the matching and subsequent 3-D reconstruction. We examine several common textures in the report and observe their effects in the accuracy of 3D reconstruction.

2.2 Internal PhotoModeler Parameters

This section describes internal parameters that PhotoModeler Scanner uses in creating a dense surface model. PhotoModeler Scanner may downsample images of the scene to improve processing time, which effectively decreases image resolution available to the correlation algorithm. We denote this factor as $\text{DSM}_{\text{sampling}}$. If the surface being modeled is an object with fine details on its surface, a smaller sampling factor should be used (higher resolution). However, if the object is featureless with flat sides, then a coarse sampling (lower resolution) would be a more prudent choice.

PhotoModeler Scanner uses window-based correlation to match corresponding pixels over image rows (using the epi-polar constraint). The size of the square window is obtained from $\text{DSM}_{\text{radius}}$ and is given by

$$\text{Window Size} = 2 \times \text{DSM}_{\text{radius}} + 1$$

The size of the window is a critical factor in the resolution of the depth obtained. A large window size may lead to overly smoothed depth estimates while too small a window size may lead to noisy depth estimates. A final parameter worthy of mention is $\text{DSM}_{\text{texture}}$ which controls the quality of the reconstructed points. This parameter is used to control the strictness of the matching based on a user-specified setting. The texture parameter $\text{DSM}_{\text{texture}} \in \{1, 2, \ldots, 10\}$, where 1 indicates an ideal texture setting of a random surface texture while 10 indicates a worst-case texture setting of a repeating pattern or a texture-less surface. The net effect of varying $\text{DSM}_{\text{texture}}$ is to increase or decrease the density of the reconstructed point cloud by only allowing points which are stably matched.

3 Results

This section presents the results of the various accuracy studies performed. In Sec. 3.1, the results of simulations using artificial 3-D scenes are detailed (assuming perfectly known camera parameters). This assumption of fully known camera parameters is relaxed in Sec. 3.2 and experiments are carried out to determine the relative accuracy when the camera pose is jointly estimated with the scene reconstruction. In Sec. 3.3, real images are used with ground truth obtained from laser scanners to evaluate the surfaces generated by PhotoModeler Scanner .

3.1 Simulation using artificial scenes

The artificial scenes in this report are generated using Autodesk Maya 2012. The images are rendered under the assumption of ideal imaging conditions using the pinhole camera model. In this section, various internal and external parameters related to DSM accuracy are examined. The primary error metric used is the root mean square (RMS) error which is given by,

$$\text{RMS error} = \sqrt{\frac{\sum_{i=1}^{N} (z_i - z_{\text{true}})^2}{N}}, \tag{1}$$

where N is the total number of points in the point cloud generated from each image pair, z_i is the depth of a particular point in the scene with reference to a global coordinate system and z_{true} is the true depth of the point. We also use a 1 part in N accuracy measure which is expressed as a ratio of the error with respect to the size of the surface being examined. Since photogrammetric techniques are not scale-dependent, a $1:N$ measure captures the accuracy for both microscopic and macroscopic scales. The only inherent limitation is the resolution of the imaging device. We define the $1:N$ term as the *accuracy measure* and it is given by

$$\text{Accuracy Measure} = \text{Error Measure} : \text{Surface Size} = 1 : \text{Surface Size} \times \text{Error Measure}^{-1}. \tag{2}$$

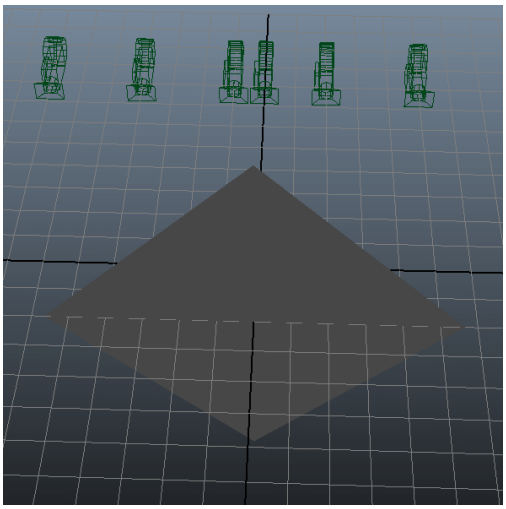
All the artificial scenes referred to in this section consider a plane parallel to the $X-Y$ plane, placed at $Z=-2$. A random texture is used as a default surface material. Since PhotoModeler Scanner uses a correlation-based method in matching image patches, a random texture provides maximum discrimination between image patches. The random texture is generated as a texture map in which each pixel is sampled from a uniformly distributed random variable over the range $[0,255]$.

Effect of B/H Ratio: The first parameter we examine is the base-to-height ratio (B/H ratio) briefly explained in 2.1. The configuration of the scene is shown in Fig. 3a. The size of the plane is 8 cm $\times$ 8 cm which results in a diagonal length of 11.31 cm. The diagonal length is considered as the surface size to be used in the calculation of the accuracy measure defined by Eq. 2. The plane is rotated by 45° in the $X-Y$ plane so that three corner points of the plane are visible in all views. This is necessary to define an approximate surface in the scene to aid in setting the disparity search range.

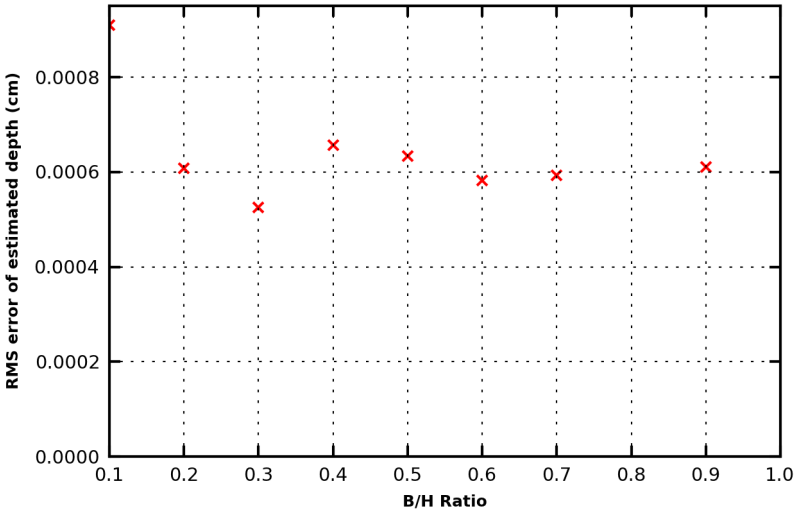
B/H Ratio	1 part in N Accuracy (from RMS error)
0.1	1 part in 17,700
0.3	1 part in 30,700
0.5	1 part in 25,300
0.9	1 part in 26,200

Table 1: Trends in accuracy as a function of B/H Ratio

In Fig. 3a, the camera angle is kept constant at 0° and the B/H ratio is increased from 0.1 to 1. The RMS error is plotted versus the B/H ratio in Fig. 3b. It can be observed that the error is high for a very small B/H ratio of 0.1 and then stabilizes to an equilibrium error between B/H ratios of 0.5 and 1.0. This pattern is observed in different scenarios using different textures. While it is difficult to draw conclusions from such a graph, the observed trend agrees with the observation made in [1] which indicates that an ideal B/H ratio for creating a digital elevation map is between 0.5-1.0. The trend in the RMS error shows oscillatory behavior around the value of 0.0006 which corresponds to a 1 part in 18,900 accuracy. The 1 part in N accuracy for certain B/H ratios is shown in Table 1.



(a)



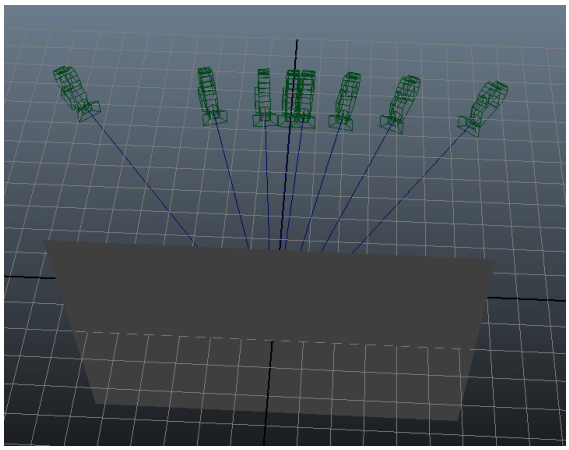
(b)

Figure 3: The artificial scene used to investigate B/H ratio effects is shown in (a) and the trend in RMS error as a function of B/H ratio is shown in (b)

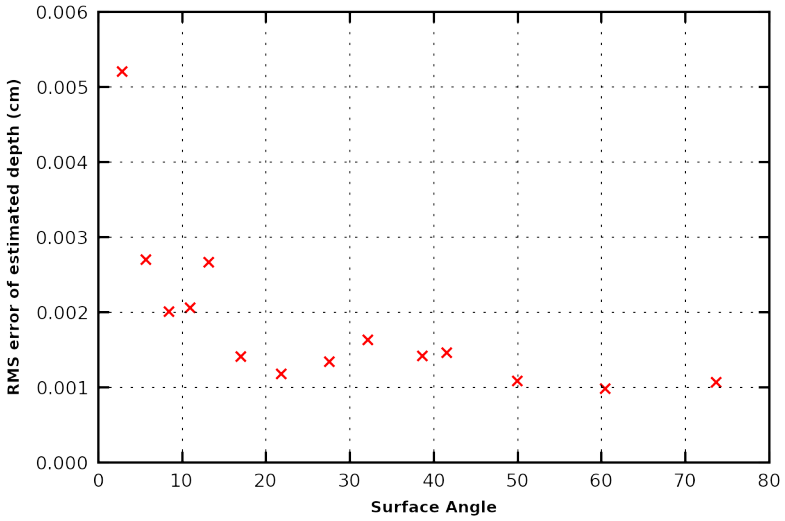
Effect of Camera and Surface Angle: The second parameter that is considered is the camera angle as defined in Sec. 2. The configuration of the scene is shown in Fig. 4a. This scene has cameras with a range of look angles, from fronto-parallel to as high as a 80° angle. It is important to note that all the cameras are aimed towards the centroid of the surface. This implies that the camera angle is the same as the surface angle (at the centroid). The dimensions of the plane used are 15cm × 13cm which results in a diagonal size of 19.85cm.

In Fig. 4b, the RMS error in the Z-depth is shown as a function of the angle to the surface. We observe that as the angle gets larger, the error reduces. This behavior is expected in the case of a flat planar surface. However, in the case of more complex shapes, there could be considerable self and scene occlusion such that these results may not necessarily hold. As the surface angle increases there is considerable perspective distortion between the two views in a stereo pair. As a result, the surface texture may look very different when seen from such viewpoints. Such differences in texture critically impact the correlation-based matching, thus decreasing the accuracy of the reconstructed point cloud.

The results using the scene in Fig. 4a are misleading because the B/H ratio is not kept constant as the camera angles are changed. When the B/H ratio is kept constant at 0.5, the change in camera angle causes the plane to undergo sharp perspective change at higher angles. This perspective change leads to a higher error. The result of a different experiment in which the camera angles are changed but the B/H ratio is kept constant at 0.5 is shown in Fig. 5. As expected, the error decreases sharply from a very low angle until 30° and then increases again as the perspective foreshortening effect becomes significant.



(a)



(b)

Figure 4: The artificial scene used to investigate surface angle effects is shown in (a) and the RMS error in the Z-depth as a function of the surface angle (in this case, also the camera look angle) shown in (b) . The B/H ratio is kept constant as the camera angle is changed in (c).

Camera Angle	1 part in N Accuracy (from RMS error)
5°	1 part in 8,400
30°	1 part in 46,400
35°	1 part in 35,600
50°	1 part in 12,200

Table 2: Trends in accuracy as a function of camera angle

The results obtained from previous experiments suggest that a base-to-height ratio between 0.5 and 1.0 together with a camera angle of around 30° leads to the greatest accuracy in depth measurements. These results are used in setting up another artificial scene to test the internal parameters of PhotoModeler Scanner by choosing optimal external parameters. Consider the scene in Fig. 7a which contains the same plane of Fig. 3a but of dimensions 12 cm $\times$ 12 cm. The major difference is that a B/H ratio of 0.6 and a surface angle of 30° is chosen to set up the cameras. Using this scene, we wish to examine the internal parameters like $\text{DSM}_{\text{sampling}}$, $\text{DSM}_{\text{texture}}$ and $\text{DSM}_{\text{radius}}$. Rendered images from the two camera viewpoints are shown in Fig. 6.

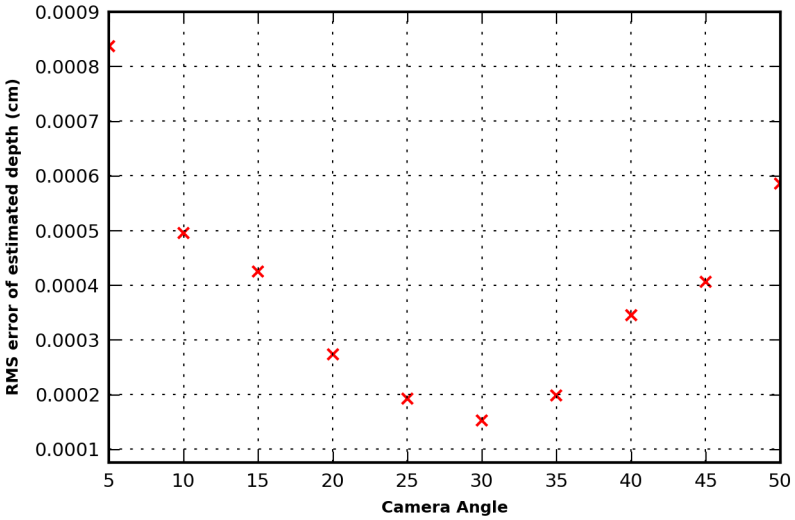
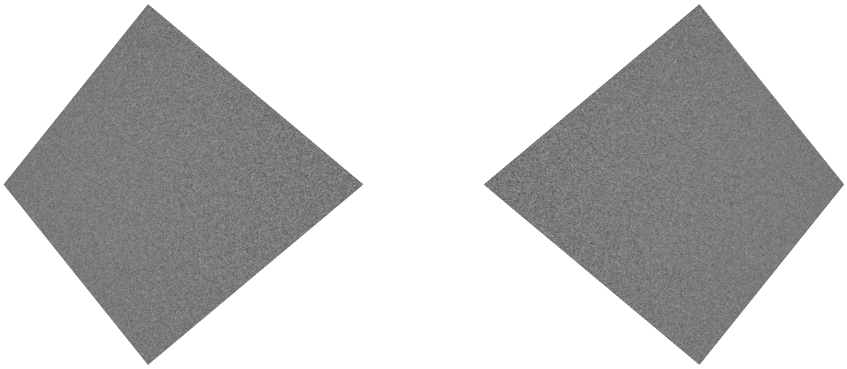
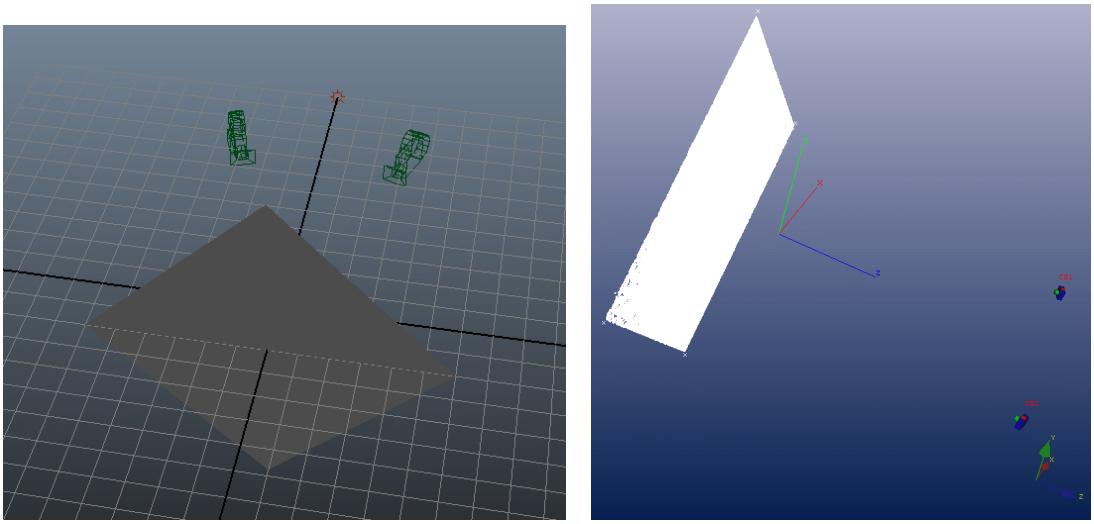


Figure 5: The B/H ratio is kept constant as the camera angle is changed. .



(a) (b)

Figure 6: The rendered images of a plane with camera (a) C_1 at $(3,0,8)$ and (b) C_2 at $(-3,0,8)$



(a) (b)

Figure 7: The artificial scene used to examine internal PM parameters in (a) and the same scene solved in PhotoModeler in (b)

Effect of $DSM_{sampling}$: We first consider varying the $DSM_{sampling}$ factor. The downsampling factor controls the resolution of the images used in the dense matching. The resolution of the image used is an important factor in uniquely representing the neighborhood of a certain pixel in the image. The $DSM_{sampling}$ factor is used primarily as a means to reduce the computational time in creating a dense surface model. As the downsampling factor is increased, a corresponding loss in high-frequency content will be observed in the image. The high-frequency content of an image corresponds to edges and sharp features which are prime factors in uniquely determining pixel neighborhoods. As observed in Fig. 8, the error increases as the downsampling factor is increased. The relationship is almost linear and is an intuitive result. Thus, for maximum accuracy, a high resolution image is recommended.

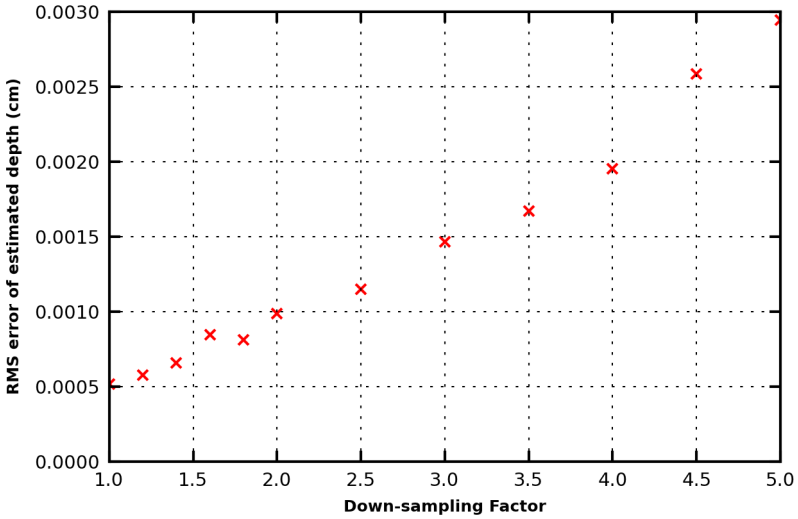


Figure 8: The trend in RMS error as a function of downsampling factor $DSM_{sampling}$.

Effect of DSM_{radius}: The next internal parameter examined is the window size DSM_{radius}. The window refers to a patch or neighborhood of fixed size around individual pixels in an image. These patches represent a template which is compared with other patches in the dense matching stage. Intuition suggests that a larger window size would lead to better matching, hence leading to more accurate 3-D point reconstructions. However, it is important to caution that this reasoning only holds when the surface is planar and in a fronto-parallel configuration with the viewing camera. Such a situation is not very likely to exist in real world conditions. A very small window size is able to resolve fine gradations on the surface but will also include noisy point reconstructions. On the other hand, a large window size will lead to an overly smooth surface and a loss in the finer structure present on the surface. In terms of accuracy of estimated depth, we observe that a larger window size leads to more accurate reconstructions for a planar surface as shown in Fig. 9.

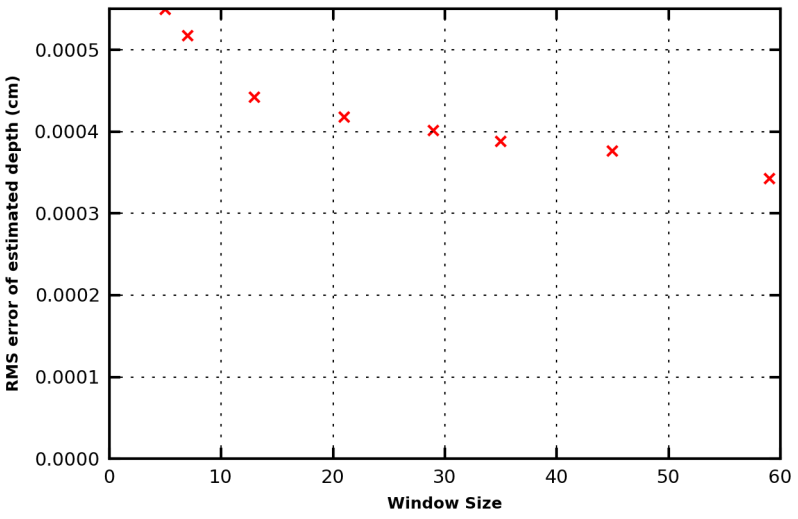
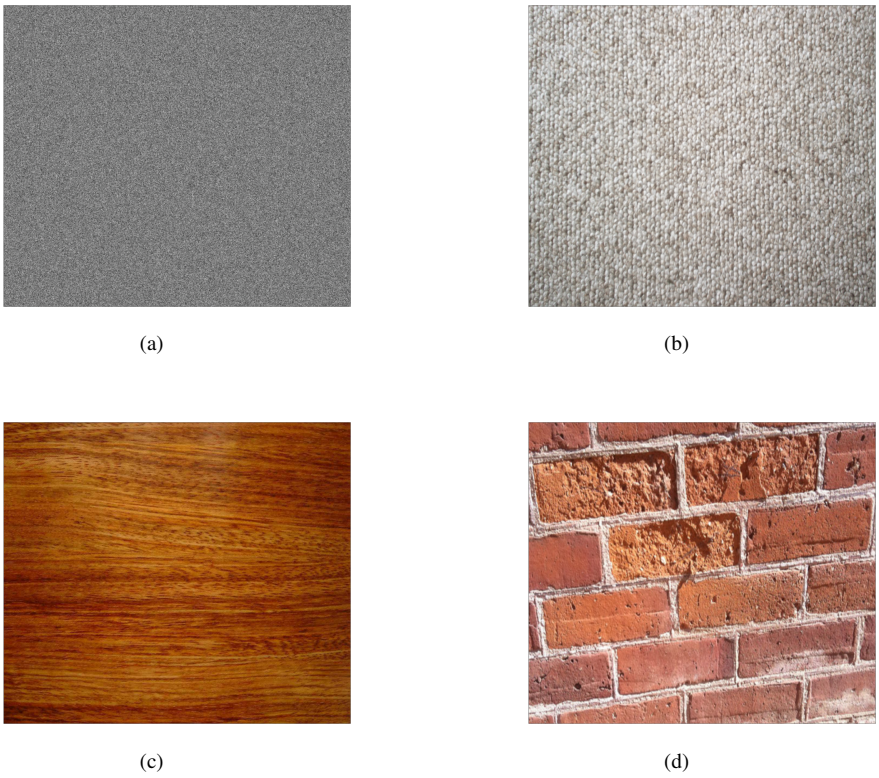


Figure 9: The trend in RMS error as a function of window size.

Effect of $\text{DSM}_{\text{texture}}$: The final parameter examined is the texture setting $\text{DSM}_{\text{texture}}$. This parameter can be used to control the quality of the point cloud while trading off the density of the point cloud. In Fig. 11a, the window size used is the default radius $\text{DSM}_{\text{radius}} = 3$ which translates to a window size of 7. We notice that the RMS error of the carpet and the random textures do not change much with the texture parameter. However, the wood and brick textures show an improvement in RMS error as the texture parameter is increased. When the window size is decreased to a side length of 5 pixels, even the random texture is shown to be affected by the texture parameter in Fig. 11b. Rendered views of the four textures used in the experiment are shown in Fig. 10. The brick texture performs the worst because it consists of fairly uniform texture on the face of each brick. The wood texture performs slightly better although it suffers because it does contain weakly repeating patterns. The carpet texture performs even better than the random texture. This could be attributed to the fact that the carpet texture has variation in color while the random pattern is mapped from a grey-scale image. The results shown in Fig. 11 reinforce the importance of sufficient texture on the surface being modeled.



(a) (b) (c) (d)

Figure 10: The four different textures used in experiments (a) Random, (b) Carpet, (c) Wood and (d) Brick

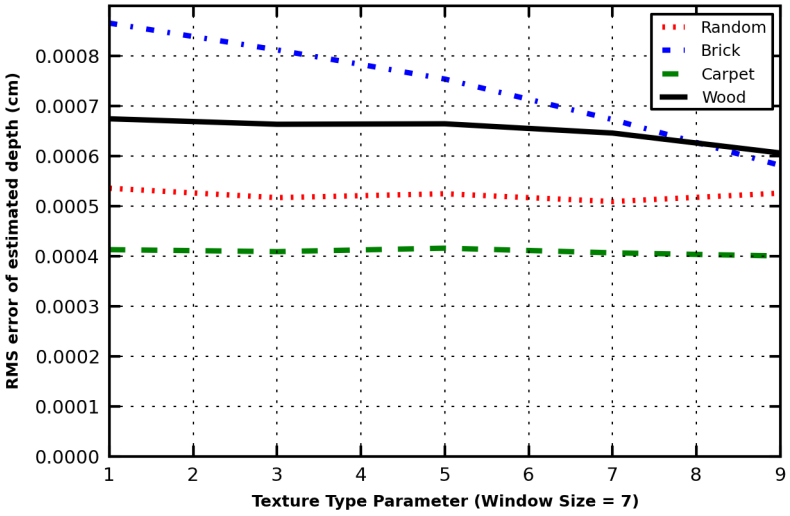
Conclusions from simulated scene experiments: Using the results from this set of experiments, we can conclude that the best accuracy can be achieved for planar surface in the scene when a base-to-height ratio between 0.5 to 1.0 is used together with a camera angle of around 30°. Moreover, a high resolution image is always better in the case of achieving higher accuracy. If the surface has a weak texture then a larger window size with a large texture parameter must be used. When a strong texture is present on the surface, a smaller window size can be used to save on computational time. In Table 3, summary statistics are shown for the planar surface with ideal parameter choices. This table shows that an accuracy of 1 part in $44,000$ can be achieved for a planar scene with perfectly known camera parameters using RMS error as the accuracy benchmark. The mean absolute error (MAE) is given by

$$\text{Mean Absolute Error} = \frac{\sum_{i=1}^{N} |z_i - z_{\text{true}}|}{N}, \tag{3}$$

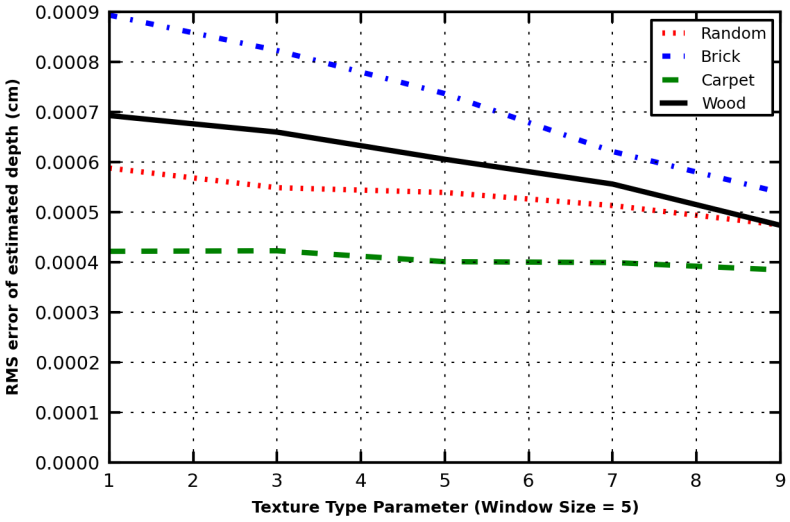
where N is the total number of points in the point cloud, z_i refers to the depth estimate of a scene point and z_{true} refers to the true depth of the scene point. A significantly higher accuracy measure is achieved using the MAE. The maximum error shows that the worst case accuracy measure is 1 part in 1300. It is defined by

$$\text{Maximum Error} = \max_i |z_i - z_{\text{true}}|, \tag{4}$$

and quantifies the largest deviation of the reconstructed surface from the ideal planar surface.



(a)



(b)

Figure 11: The trends in RMS error as a function of the texture parameter for different texture types in (a) with a window size of 7 pixels and in (b) with a window size of 5 pixels

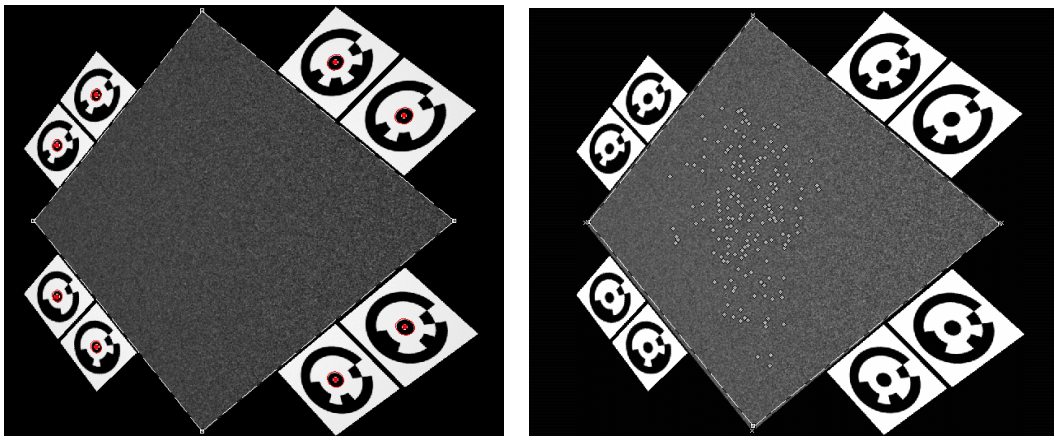
Statistic	Value	1 part in N Accuracy
RMS Error	0.000385	1 part in 44,100
Mean Absolute Error	0.000030	1 part in 575,300
Max Error	0.012651	1 part in 1300
Standard Deviation	0.000384	N/A

Table 3: Summary statistics using ideal parameters

3.2 Simulations with artificial scenes and unknown camera pose

In this section, the constraints on the project are relaxed by assuming that the camera positions and orientations are unknown. Two methods are used to solve for the unknown camera pose within PhotoModeler. The experiments carried out in this section reflect the methodology depicted by the right-hand branch of the flowchart in Fig. 1. The first method requires user intervention before capturing images of the surface to be modeled. A few high-contrast coded targets are placed around the surface. These targets can then be automatically detected in the scene photographs and their centers can be marked with sub-pixel accuracy. The unique pattern of the coded target also allows them to be accurately matched. Rendered versions of the scene with coded targets are shown in Fig. 12a and the detected

centers are marked in red. The second method is fully automatic and requires minimal user intervention. It uses feature detection to find interest points in the image. These interest points are then robustly matched between images. In Fig. 12b, detected interest points are shown on the randomly textured plane. In either case, PhotoModeler uses the matched points to solve for the positions and orientations of the camera. It is important to point out that the internal parameters of the camera are still assumed to be perfectly known.



(a) (b)

Figure 12: The coded targets with their centers marked with sub-pixel accuracy in (a) and the interest points detected on the surface using a feature detector in (b)

In Fig. 13, the RMS error of the Z-depth for the scene in Fig. 7a with the plane at $Z = -2$ is shown as the number of cameras in the scene is increased. An increasing number of cameras is used with the expectation that the redundancy introduced by a larger number of correspondences between points on each image would lead to a stronger orientation solution. The multiple cameras are placed in front of the plane in a configuration that tries to match the ideal conditions described in Sec.3.1 as closely as possible. We notice that the RMS error of the RAD targets decreases slightly as the number of cameras is increased but with a very gentle slope. The SM procedure (described in Sec. 2) shows a more dramatic decrease in the RMS error. Both methods produce diminishing returns at 6 cameras. We expect the accuracy to get better even beyond 6 cameras but with a much lower rate. This can be attributed to the fact that the scene being considered is a relatively simple scene. A more complex scene, will generally achieve higher accuracy with a larger number of images.

Table 4 lists the accuracy measure for both cases using RAD targets and SmartMatch. Using targets in the scene achieves a maximum accuracy of 1 part in $18,000$. The fully automatic SmartMatch procedure achieves a maximum accuracy of close to 1 part in $10,000$. Thus, there is a large decrease in achievable accuracy when the camera pose is not known and must be jointly estimated with the reconstruction of the scene (in contrast to the 1 part in $44,000$ accuracy achieved using known camera orientation in Sec. 3.1). Using targets allows precise localization and unique matching between points in a stereo pair. They also impart a degree of invariance of viewpoint changes, lighting changes etc. On the other hand, the feature points detected using SmartMatch are not entirely invariant to large changes in viewing angle and non-linear pixel intensity changes. As a result, spurious matches may be included in the orientation procedure which can lead to inaccurate estimation of camera pose.

Number of Cameras	Accuracy using RAD Targets	Accuracy using SmartMatch
2	1 part in $16,100$	1 part in $1,400$
3	1 part in $17,400$	1 part in $3,300$
4	1 part in $17,700$	1 part in $9,700$
6	1 part in $18,000$	1 part in $9,000$

Table 4: Summary statistics using ideal parameters

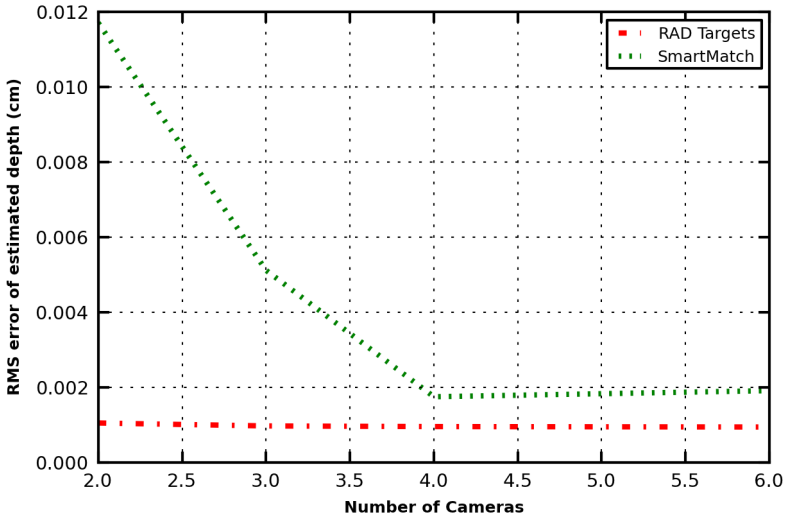


Figure 13: The RMS error as a function of the number of cameras for projects oriented using RAD targets and by using interest points detected by SmartMatch

3.3 Accuracy analysis using real world scenes

In this section, the collection of the ground-truth data is described in Sec. 3.3.1 and the results of DSM accuracy studies on real world scenes are presented in Sec. 3.3.2. The ground-truth data used is in the form of a 3D point cloud output from a state-of-the-art laser scanner. The laser scanner used is the FARO Focus^{3D} which has a normal scanning range from 0.6m to 20m for a surface with at least 10% reflectivity. When the ambient light is low, its range can be extended from 0.6m to 120m for a surface with at least 90% reflectivity. The accuracy of the Focus^{3D} is documented to be $\pm$2mm within a 10m range. As with most technical specifications, this accuracy is presumably obtained in ideal laboratory settings. We use this benchmark to test the real-world accuracy of PhotoModeler Scanner under general operating conditions. This is an important distinction as we are comparing the accuracy of PhotoModeler under general non-ideal conditions to the accuracy of a reference laser scanner whose accuracy is determined under ideal laboratory conditions. Our main motivation in using real world scenes is to obtain an accuracy analysis under the most general and typical scenarios that a typical PhotoModeler user expects.

Images of the scene were taken using a Sony A200 DSLR camera at a fixed focal length of 20mm. Each image has a resolution of 3872×2592 pixels (10 Mega-Pixel resolution). A precise camera calibration was also carried out to estimate lens parameters with a high accuracy for use in the 3D reconstruction.

3.3.1 Establishing the ground-truth

In this section, the collection of ground-truth data and its assimilation into a comparable form with PhotoModeler Scanner output is explained. A natural scene is captured as a 3D point cloud using a the Focus^{3D} laser scanner. The scene is setup with manually placed spherical targets and high-contrast 2-D targets. The centers of the spherical targets can be detected using the software bundled with the Focus^{3D}. Furthermore, the same targets can also be detected with sub-pixel accuracy in PhotoModeler Scanner. The use of such targets plays two important roles: 1) they can be used to solve for the camera pose and 2) they can be used to define an coordinate alignment transform between point clouds measured in PhotoModeler's coordinate system and the laser scanner coordinate system. Such a transform establishes a common coordinate system between the two point clouds and is an essential step in comparing the relative accuracy of PhotoModeler Scanner output. It would be important to point out that the localization of the targets to a high accuracy is an important limiting factor in this accuracy study. There are two sources of error: the first comes from the laser scanner software and the second comes from the sub-pixel marking in PhotoModeler. We will not comment further on these errors as they are assumed to be much lower in scale than errors introduced by the 3D reconstruction. The high-contrast coded 2-D targets (as shown in Fig. 15) are also used by PhotoModeler for automated marking and matching. The centers of these targets can also be localized with a high accuracy and are used to solve for the camera pose.

3.3.2 Comparison of PhotoModeler Scanner with LIDAR

In this section, we describe the workflow used in determining the relative accuracy of point clouds generated from PhotoModeler Scanner . This workflow is also depicted graphically in Fig. 14. A set of input images with accurately calibrated intrinsic parameters forms the input to PhotoModeler Scanner. The camera pose corresponding to each input image is solved using either targets or automatically detected feature points. A dense surface model is then created

from each pair of appropriately selected photos. Next, the pair-wise clouds are merged into a single cloud. This results in a composite representation of the scene covered by the photos. The laser scanner captures the scene in a coordinate system that is different from the coordinate system of a multi-view camera setup. To align their respective coordinate systems, a suitable transformation must be estimated. This requires matching a minimum of three 3-D points between the laser scanner point cloud and the PhotoModeler Scanner point cloud.

For the coordinate alignment procedure, we used spheres targets in the scene. Laser scanners can accurately detect the centers of sphere targets. More importantly, PhotoModeler Scanner also provides the ability to mark spherical targets which can be used to set up a correspondence between the laser scanner point cloud and the stereo point cloud. Once the coordinate system transformation has been estimated, each point in the laser scanner point cloud is transformed using the coordinate transformation to obtain comparable point clouds.

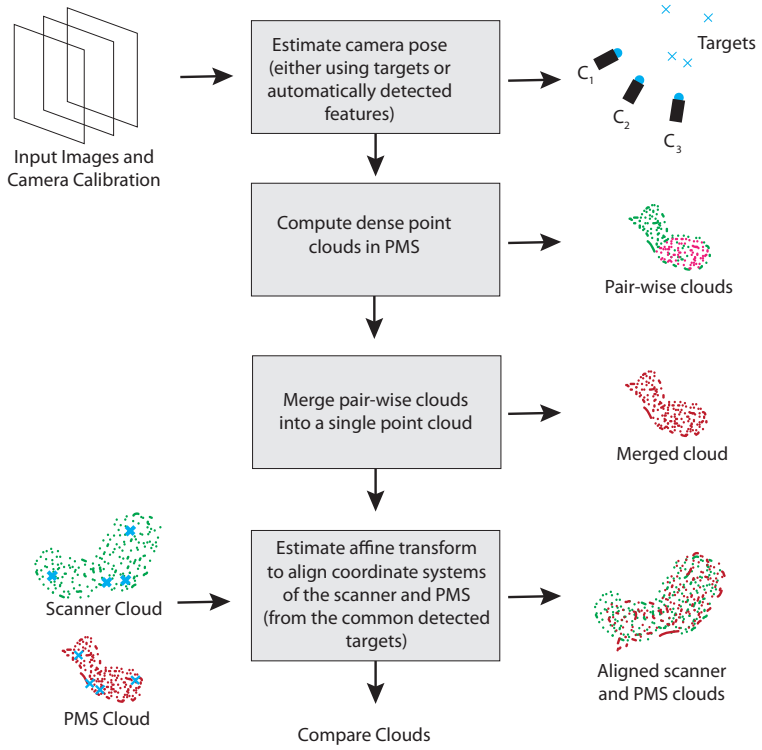


Figure 14: A graphical depiction of the workflow used to compare a point cloud from PhotoModeler Scanner and a point cloud from a laser scanner

A point cloud comparison tool CloudCompare [2] is used to determine the relative error between PhotoModeler Scanner output and the laser scanner output. The point clouds are compared in two different ways. In either method, the maximum bounding cuboid for both point clouds is computed and the 3-D volume V represented by this bounding cuboid is discretized into voxels v. This is done using an efficient octree structure which allows fast searching of massive point clouds.

In the first comparison method, every point $X = (x, y, z)$ in the PhotoModeler Scanner cloud Q is compared to its nearest neighbor $NN(X)$ in the reference laser scanner cloud T. Thus $NN(X) = \arg\min_{X'_j \in v,T} |X - X'_j|$. Each point in the PhotoModeler Scanner cloud is assigned a distance metric (regarded as an error) from the reference laser scanner cloud. The resulting scalar field $H(x, y, z)$ is then used to estimate various statistical parameters like the mean and standard deviation of the error.

In the second method, the nearest neighbor approach is abandoned in favor of a locally fit surface between the template points in the voxel that X belongs to. This provides a continuous function to compare the query PhotoModeler Scanner cloud against. We use two different local best fit surfaces (LBFS), namely, a least-square plane and a Delaunay triangulation. Similar to the first method, each point in the PhotoModeler Scanner cloud is assigned an error or deviation from the locally best-fit surface in the reference scanner cloud. The error at each point is

$$H(x, y, z) = \|X - NN(X)\|^{\frac{1}{2}}, \quad \text{for the first comparison method, and} \tag{5}$$

$$H(x, y, z) = \|X - LBFS(X)\|^{\frac{1}{2}}, \quad \text{for the second comparison method,} \tag{6}$$

where $LBFS(X)$ represents the local surface fit in the voxel v which X belongs to and $\|\cdot\|$ refers to the L_2 norm. In this voxel v, the points $X'_j \in T$ are used in fitting the local surface, which can either be a plane or a Delaunay triangulation.

The notation Q for the PhotoModeler Scanner cloud refers to it being a *query* point cloud. The scanner point cloud T is the *template* cloud.

The scene chosen for this accuracy surface is a brick wall which is a relatively flat surface. Some views of the wall used are shown in Fig. 15. The scene is set up with two different kinds of targets. To ensure a high accuracy in the camera orientation, coded targets are placed in the scene which can be automatically marked and matched in PhotoModeler to obtain the extrinsic parameters of the camera views. More importantly, spherical targets are also placed in the scene so that PhotoModeler can align the coordinate axes of the laser scanner and the multi-view stereo system. As an added advantage, the sub-pixel accuracy of the sphere target marking in PhotoModeler can be used to refine the original orientation (that was done using coded targets).

(a) (b)

Figure 15: Two different views of the wall scene

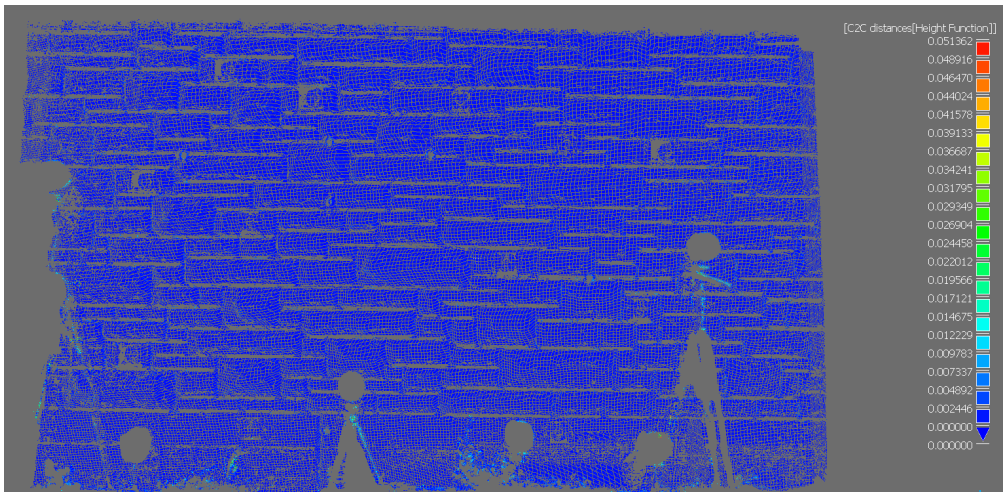
The orientation of the cameras results in the reconstructed scene depicted in Fig. 16b. After this stage, a number of suitable pairs are chosen for dense matching. Each pair of photos results in an output point cloud. These pair-wise point clouds are then merged into a single point cloud. Next, the laser scanner point cloud is imported into PhotoModeler Scanner together with the marked locations of the spheres (as output by the laser scanner software). The point cloud generated by PhotoModeler Scanner is then aligned with the laser scanner point cloud using a Helmert transformation [3]. The Helmert transform solves for rotation, translation and scaling parameters by introducing minimal distortions.

Subsequently, the clouds are imported into the comparison tool CloudCompare [2]. As an added precaution, the laser scan cloud and PhotoModeler Scanner cloud are mutually registered using an iterative closest point (ICP) procedure [4]. This ensures that the coordinate system alignment is as tight as possible. Every point from the query point cloud (the PhotoModeler Scanner cloud) is then compared to the template point cloud and a measure of the difference between them is found. This difference is shown as a color map in Fig. 16a. As observed, most of the points generated by PhotoModeler Scanner are highly accurate in relation to the laser scanner. A histogram of the differences is shown in Fig. 16c.

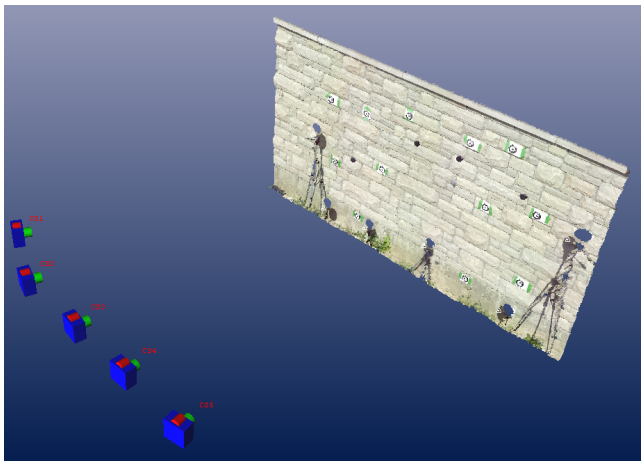
Comparison Method	Mean Error	Accuracy Measure	Accuracy (w.r.t 3m range)
Point to Point	0.001737	1 part in $3,200$	$\pm 1.737mm$
Point to Plane	0.0010405	1 part in $5,300$	$\pm 1.04mm$
Point to Delaunay Triangulation	0.000957	1 part in $5,800$	$\pm 0.9mm$

Table 5: Summary statistics of PhotoModeler Scanner Cloud v/s Laser Scan Cloud

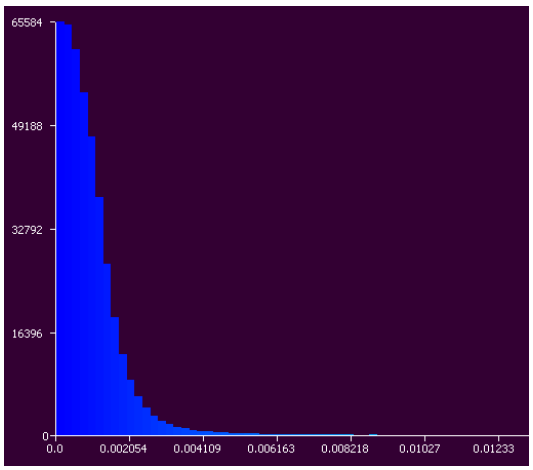
In Table 5, the results of running different comparison methods are shown. We observe that a simple nearest neighbor point-to-point comparison results in a mean error of $1.737mm$, that of using a point-to-best fit plane method results in a mean error of $1.04mm$ and using a point-to-Delaunay triangulation results in a mean error of $0.9mm$. The camera views are at an average of $3.6m$ away from the surface being modeled. As mentioned previously, the FARO Focus^{3D} has an accuracy of $\pm 2mm$ within a $10m$ range. The accuracy of PhotoModeler Scanner is well within this limit at the operating range of this experiment even under non-ideal real world conditions.



(a)



(b)



(c)

Figure 16: (a) The cloud to cloud difference between the PhotoModeler Scanner cloud and the laser scanner cloud shown as a color map. The smallest differences appear as dark blue on the scale. (b) The camera orientation and dense surface modeled in PhotoModeler Scanner. (c) The histogram of errors (point to point) between the PhotoModeler Scanner cloud and the laser scanner cloud (in cm).

4 Conclusion

In this section, the results of the simulations and analysis carried out are used to provide guidelines for ensuring a high level of accuracy in photogrammetric solutions using PhotoModeler Scanner. The accuracy of a photogrammetric project depends on the setup of the scene and the camera to surface geometry. For best results, a B/H ratio between 0.5 and 1.0 should be used. While this may seem too large to provide sufficient overlap, we can overcome this by also introducing a moderate camera angle between each pair of camera viewpoints. In addition to providing overlap, the moderate angle (between the cameras and subtended at the surface) also provides a more accurate solution. It is also recommended to work with the highest resolution of images available for best results. A larger window size also leads to more accuracy when the surface is uniquely textured. However, the computational trade-off is not worth the incremental gain in accuracy beyond a certain window size.

With regards to the project setup, it was also established that some effort in placing targets in the scene can reap rich rewards in terms of gain in accuracy. Moreover, as the number of cameras (image viewpoints) is increased, the added redundancy also allows for a more accurate orientation and in turn, a more accurate surface. Ensuring that the surface being modeled has a pseudo-random non-repeating texture also contributes to the eventual accuracy of the solution.

Finally, it was shown that the 3-D position accuracy of dense points from PhotoModeler Scanner is well within the accuracy range of a point cloud generated by an industry-accepted laser scanner.

5 Acknowledgments

We would like to acknowledge the kind assistance of Eugene Liscio (AI2-3D, http://www.ai2-3d.com) in providing the real world scene photographs and laser scanner ground-truth point cloud.

References

[1] H. Hasegawa, K. Matsuo, M. Koarai, N. Watanabe, H. Masaharu, and Y. Fukushima, "DEM accuracy and the base to height ratio of stereo images," in *Proceedings of the Japanese Conference on Remote Sensing*, vol. 27, 1999, pp. 91–94.

[2] D. Girardeau-Montaut, "Cloudcompare v2.1," http://www.danielgm.net/cc/.

[3] G. Watson, "Computing Helmert transformations," *Journal of Computational and Applied Mathematics*, pp. 387 – 395, 2006.

[4] Z. Zhang, "Iterative point matching for registration of free-form curves," *International Journal of Computer Vision*, vol. 13, pp. 119 – 152, 1992.