Interactive Guidance and Navigation for Facilitating Image-Based 3D Modeling

Damon Shing-Min Liu and Te-Li Chang
National Chung Cheng University, Taiwan
168 University Road, Chiayi, Taiwan 621
damon@computer.org and tlchangfm@gmail.com

ABSTRACT

Here we present an interactive guidance and navigation system that assists user in acquiring pictures for image based 3D modeling. To reconstruct an object's 3D model, user follows our instruction to take a set of images for an object in different angles, we calculate their relative viewing positions and spare point cloud data using *structure from motion* technique. After we obtain sufficient number of images, we use *Patch-based Multi-View Stereo* (PMVS) [1] software to generate dense point cloud data. When displaying dense point cloud, we provide user an interface to eliminate those noise data points yielded from background construction or re-projection errors. Afterwards we reconstruct surface mesh as output. Our system provides informative message for failures while calculating camera poses and helps user how to resolve those problems. Furthermore, we assess the quality of camera poses reconstruction and generated point cloud to reveal the lack of angles for captured images and guide user to remedy those information.

Keywords
Image based modeling, 3D reconstruction, User interface, Point cloud, Structure from motion.

1. INTRODUCTION

More and more applications, such as movies, games, or 3D printers, need plenty of realistic 3D models. Therefore, many image based modeling systems have been proposed by previous research and also have significant results that can provide sufficient 3D object information. For ordinary users who are not familiar with modeling, they may have difficulty in getting a complete image set for reconstruction due to lack of knowledge and experience. Sometimes, like in travelling or shopping, it is costly to re-acquire that complete information. For this reason, we aim to build an interactive system that guides user to get a proper image set that can be contributed to a successful 3D reconstruction.

In general, we can use images and their relative captured position data to calculate a point's 3D location in real world. That means reconstruction of an object's 3D model not only requires the image set surrounding the object, but also the camera parameters data for each image. One of the prior researches, called *Structure from Motion* (SfM), can estimate camera parameters from the same feature point in different images; it extracts and matches feature points from each image, then it uses those feature points to estimate object's motion and structure. While SfM tracks feature points in each image, it may fail to match feature points if angle, distance, or luminance has huge difference between images. Even though feature points are matched, it still may fail to estimate object's motion and structure if there is poor geometry consistency. Although we can use some devices, like a turntable

and tripod, to maintain the consistency of features and geometry, it is not suitable for condition like in an outdoor environment or for objects of heavy weight. Therefore, the goal of our interactive guidance and navigation system is to let user shoot around objects using handheld camera, ensuring input images are sufficiently surrounded the object and each image's camera parameters can be calculated so as to guarantee 3D model can be successfully reconstructed.

2. RELATED WORK

Traditionally, 3D models are generated manually that requires specific skill thus putting up barriers for ordinary users. There are several ways to generate realistic 3D models automatically such as using a laser scan device or image based modeling technique; the former can provide very accurate 3D data but those devices are usually expensive; the latter only needs a camera for input so that it is affordable for ordinary users. In recent years, many approaches have been proposed to image based modeling. In this section we describe prior works of image based modeling (Section 2.1) and interactive modeling (Section 2.2).

2.1 Image based modeling

The information of relative position or camera parameters in each image was indispensable. There are several ways to get such information, which is called "camera calibration". Baumberg et al. [2] presented a commercial software solution to 3D scanning. They used a calibration target for camera calibration. Because the

relationships of calibration pattern are already known, it can be used to estimate cameras position. After estimating cameras position, they used "shape from silhouettes" approach to reconstruct object's 3D model so that their system can handle untextured or reflective objects and uncontrolled lighting. Using calibration target can easily estimate cameras position but it limits objects scale. In other words, objects must be smaller than calibration target that is able to place on it. Hua and Liu [3] introduced an approach for 3D surface reconstruction from two uncalibrated views. They used feature points (i.e., corners) matching between two images to estimate fundamental matrix and computed camera parameters. Subsequently they used camera parameters to compute projection matrix and reconstructed 3D structure. Because they only use corners as feature, their approach can only reconstruct the polyhedron object. Furthermore, they did not propose multi-view solution, implying that their approach may fail to reconstruct multi-images. Liu et al. [4] developed a multicamera 3D studio to capture multiview video and proposed a reconstruction algorithm. Their algorithm can overcome occlusion, noise, textureless problem and reconstruct a free-viewpoint video, but their system used twenty cameras placing in a dome region that is unfit for ordinary users. Snavely et al. [5] presented *structure from motion* (SfM) and an image based rendering algorithm which is able to reconstruct numerous well-known world site scenes. They detected and matched the same SIFT feature in different images, then used those feature points to reconstruct cameras and sparse geometry. Other works also used SfM to reconstruct camera parameters and sparse 3D point cloud, and used several ways to refine [6, 7] or reconstruct 3D scenes [8-11]. Kim et al. [11] proposed an instant reconstruction of 3D surface. They selected reference and target image, then integrated and refined 3D triangular patch. On the other hand, some research used low-cost and easy-to-use consumer depth cameras like Kinect to digitalize 3D objects [12].

H. M. Nguyen et al. [13] evaluated the most promising 3D reconstruction software packages. Their evaluation shows that in deficient images condition using correspondence-based approaches is better than silhouette-based methods. Their evaluation also shows that minimum number of input images is twelve, and the number of input images for good details is greater than or equal to twenty.

2.2 Interactive modeling

There are many methods that can generate 3D model using only 2D images. However, the result of reconstructed model depends on the input images. Therefore, there is other research for interactive modeling system. G. Simon [14] described a pure image based modeling system. Their system used video camera with three keyboard keys as interaction devices for modeling and tracking. Using camera movement and keyboard click in correct position, user draws a line stroke in video image then constructs a polyhedral model. Their system focuses on polyhedral scenes so it cannot reconstruct complex objects. K. Kim et al. [15] proposed a real-time solution for modeling and tracking. User defines a planar facet and extends site to create the 3D model, and system automatically fits the facet edges to the image contour to refine the control points. Through this way, they can build polygonal prisms and cylinders model. Their system can immediately build various models but the models must be polygonal or circular-based. L. Quan et al. [16] proposed a semi-automatic technique for modeling plants directly from images. This technique not only used SfM for modeling, but also extracted and reconstructed leaf and branch structure via plant's physical presence. They used the similarity of plants to generate leaf and branch, this method specifically suits for plants but not suits for other object. K. Fudono et al. [17] proposed an interactive 3D modeling interface that indicates camera movement and displays preview of reconstruction result. They used marker sheet to estimate camera position and posture, then used *shape from silhouette* method to reconstruct 3D model. Furthermore, they initialized a voxel model placed on the center of the marker sheet and computed colored and uncolored voxels for determining and indicating best view position. Because they used *shape from silhouette* method, the base color of marker sheet, region of table, and the region of wall surface have to be of the same color, those restrictions limit the flexibility of modeling. Q. Pan et al. [18] demonstrated an augmented reality guidance method guiding the user in an interactive modeling process based on ProFORMA [19]. First, they used *bundle adjustment* to create point cloud, then converted points into a mesh through a Delaunay tetrahedralisation. Second, they assumed an icosahedron placed at its center of mass and calculated each face's uncertainty score. High viewpoint's uncertainty scores represent orientations which should be visited, whereas low viewpoint's scores represent orientations from which there is already a lot of information. Using the score and augmented arrow guide users to rotate the object for providing new information. When in modeling, their system needs user to rotate the object in front of a stationary camera. However, sometimes the object cannot be rotated, so that our system needs user to capture images around object instead of rotating the object. H. Du et al. [20] utilized Kinect style consumer depth cameras to scan personal spaces into 3D models. They used SIFT feature and depth information to indoor 3D mapping. When matching failures, their system will "Rewind and Resume".

That is the system will pause the mapping process, and wait for a new frame that can be successfully registered. Moreover, they considered a bounding box that contains the currently reconstructed point cloud. Using 3D grid voxels to represent inside of the bounding box and classify voxels into one of the three categories: *occupied* (Red), *empty* (Green), and *unknown* (Blue), that can assist users in finding uncaptured areas. User's goal is then to paint all areas in either green or red by exploring the 3D space. Since the consumer depth cameras have not been widely available for every user and it does not suit for outdoor environments, we want to build a modeling system which solely uses hand-held cameras for modeling.

3. APPROACHES

When modeling an object from images, it first needs to know each image's camera parameters (CP), then uses images and their corresponding camera parameters to generate 3D model. In our experiments, we used "Bundler" [21] to estimate CP and made some modification to suit for our system. Bundler is a *structure from motion* system for unordered image collections; it takes a set of images, image features, and image matches as input then produces a 3D reconstruction of camera and (sparse) scene geometry as output. In our system we sequentially guide user to take images input, the new input images will only relate to those images which is nearby. For this reason we made some modification: when matching feature points from an image pair, rather than match all images pairs, system only matches those images pairs which differ from previous azimuth angle by an amount smaller than or equal to 45˚; that is, given two images and their previous azimuth angles ϕ_i, ϕ_j, system only matches those images which have $\left| (\phi_i - \phi_j) \right| \leq 45°$. After inputting new captured images (including the initial two images), we use an expected azimuth angle, i.e., the place where system expects user to capture image of the object, since there does not have any information. This modification to Bundler not only saves the execution time of matching features, but also solves the problem of estimating wrong position in input images having similar pattern. When object has similar pattern in different face, Bundler will mistakenly consider those images were shot from the same face because they have similar SIFT features.

While SfM estimating CP, it may fail to estimate due to reason like new image is too far from previous images, lack of feature or blurred. Without CP, it is impossible to reconstruct object's 3D model even though it has depth information generated from depth sensors like Kinect. However, even if it can successful estimate CP, it may still compute wrong CP due to similar feature or mis-matching (false positive). For this reason, system must have ability to

determine those conditions and guide user to fix those problems. So the goal of our system is guiding user to get sufficient information for modeling, i.e., input image set not only has to successfully estimate CP but also has to have surrounded object sufficiently. For this reason we designed an acquiring process that takes images captured from user as input, and interactive guides user where needs to take next image. At the same time, system estimates image's camera parameters between each image and evaluates input image set to see whether it is sufficient for reconstructing the entire 3D model. Our system flowchart is shown in Figure 1. The rest of the section describes the factors that affect estimation of CP and failure handling. Furthermore, in some application like augmented reality, we desire to know the size of a model in real world, which means we have to know the transformations from real world to virtual 3D space and the transformations are described as well.
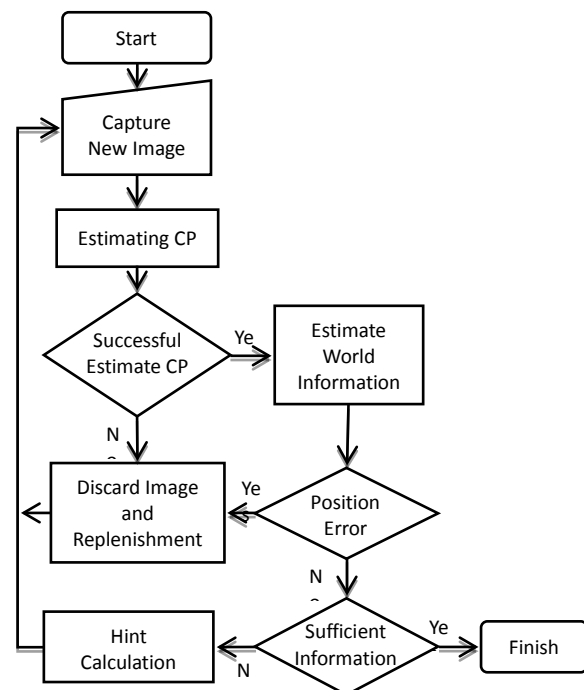

Figure 1. System flowchart.

3.1 Factor affecting estimation

In our experiments we found two common conditions that caused failure for estimating CP: 1) far distance between existing image set and new input image, and 2) poor existing geometry (Figure 2). When inputting a new image, system will find the existing feature points which can be matched by new image then estimate CP. If existing image set and new image have a far distance, it fails to estimate CP because new image can only be matched few feature points in existing geometry; on the other hand, if existing geometry only has few feature points, even if the new

image is close to existing image set, only few feature points can be seen so that it still fails to estimate CP.
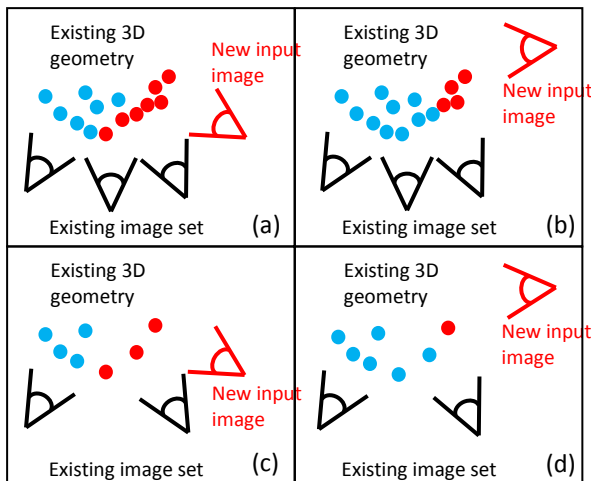

Figure 2. The relationship between existing information and new input image; Red points represent points that can be seen in the new input image; (a) Rich 3D geometry and near distance case; (b) Rich 3D geometry and far distance case; (c) Poor geometry and near distance case; (d) Poor geometry and far distance case.

In experiments we found that 1) if existing geometry had rich amount of feature points but new input image was far from existing images, it failed to estimate the new input image's CP; on the other hand, 2) if existing geometry only had few feature points but new image was close to existing images, it not only failed to estimate the new input image's CP, but also failed to estimate the closest images' CP. Because when Bundler optimizes CP it will detect and remove outlier points, and there are only few points in existing geometry so that the close image is likely to be removed after re-running the optimization. In case 1), system simply guides user to re-shoot close to existing images; in case 2), system detects where needs to replenish points by finding the local farthest distance, then guides user to move backward to re-shoot at that position. Note that if both of those conditions cause failure to estimate CP, system will only guide user to shoot close to existing images since we do not know where user last time shot this image.

After system successfully estimates CP, we examine the new input image to see whether it is at expected position. For this we perform three examinations: *backward to the object* examination, *upside down* examination, and *different from hint position* examination. When user takes a picture of object, he or she must face the object and hold camera upright. If estimated position is backward to the object or upside down, it must be a wrong position and system should reject this image. Furthermore, if the difference between new image's azimuth angle and

the expected azimuth angle is more than $60°$, then we consider it is a position error since we only match those images which differ from previous azimuth angle by an amount smaller than or equal to $45°$. Note that we do not examine whether user moves forward or backward in this step, i.e., even though user takes a shot backward, this image is still useful for modeling as long as this image can be contributed to successfully estimate CP.

Before system evaluates the sufficiency of existing information and calculates hint while guiding user, it needs to define the virtual space coordinates. That is because CP is described as a relative position in each camera, it lacks information of object center, the direction of x, y, z axis, and azimuth angle. In our experiments we averaged center of mass and user focus point as object center. User focus point is the intersection points of all images' normal vectors. After defining object center, we want to know the axis directions. We used regression plane as a horizontal plane to define the z-axis direction, and used the direction from origin to first image as x-axis direction. Regression plane is a plane minimizing the sum of distance from plane to every image. We also added a weighted object center to prevent the plane tending to images.

Now system can evaluate existing information, Q. Pan et al. [18] used a viewpoint uncertainty icosahedron to score and display existing information. The uncertainty scores were calculated using surface triangles and cameras orientation. Here we made some modification in our system: instead of using *uncertainty scores*, we used "*certainty scores*" which are high certainty scores representing the certain face already has a lot of information and low certainty scores representing the orientation which needs to be visited. We used points and cameras' orientations to calculate certainty scores. Here we made two modifications because our system did not have surface triangles to estimate unseen faces. Furthermore, we let icosahedron to "stand alone z-axis", that is, there are ten faces surrounded z-axis and other ten faces are above/below the top/bottom, because we consider the horizontal direction is more important than other directions.

After calculating the certainty score, it can simply use certainty scores to guide user to move to the face which is lack of information. But if the face is too far for existing image set, it will fail to estimate CP since there has no matched feature points. Even though it can successfully estimate CP, it is inconvenient for users because system may guide users to change moving direction rather frequently. Therefore our system has two types of hint: *horizontal hint* and *replenish hint*. System initially guides user to move around the object horizontally since we consider the horizontal is more important than others and it also

provides convenience and consistency. After the horizontal direction moving is finished, system calculates replenish hint that guides user to complete insufficient information using certainty score.

3.2 Geometry verification

After acquiring sufficient amount of images and computing their CP, user can reconstruct dense 3D geometry using PMVS [1]. In our experiments, the result of PMVS often generated noise points due to table's surface texture, background or reprojection error. We provide multiple user interfaces for user to clean out those noise points: *delete plane*, *delete/preserve polygon*, and *delete cube*. After user finished cleaning process, he or she can reconstruct mesh and mapping surface texture from images. We use "Fourier surface reconstruction [22]" to reconstruct surface mesh because it is robust to noises and tends to outperform other methods in [23]. Since our work focuses on acquiring sufficient images for modeling, we do not further exploit other methods to refine or replenish the mesh.

3.3 Size estimation

When modeling an object, it involves series of transformations from real world to virtual 3D space: camera projects the lights from real world to sensor that transforms world coordinates to sensor coordinates, sensor transforms the lights to digital image that transforms sensor coordinates to image coordinates, and image based modeling technique transforms 2D images to 3D model that transforms image coordinates to virtual 3D coordinates. The corresponding equation can be written as:

$$Real\ World_{(mm)} \times Magnification \times \frac{Resolution_{(pixel)}}{Sensor\ Size_{(mm)}}$$

$$\times \frac{Image\ to\ Center_{(unit)}}{Focal\ Lengt\ h_{(pixel)}} = Virtual\ Space_{(unit)} \qquad (1)$$

The magnification ratio depends on the distance from camera to object and the focal length; the unit is unit length in OpenGL. When bundler estimates CP, it uses focal length (in pixel) of initial image to set as the distance from initial image to object, then projects image's feature points into virtual space. For Equation 1, we use "jhead" [24] to get image's EXIF (Exchangeable Image File format) information which includes resolution, sensor size, focal length, and distance, then use focal length and distance to estimate magnification ratio value. Since we have the corresponding equation, we can approximate the scale of object in real world. We provide user an interface to select two points on the screen, system automatically calculates using Equation 1 then outputs the result.

4. EXPERIMENTS

We have implemented our interactive guidance and navigation for image based modeling. All experiments were run on a PC running Windows 7

with an Intel Core i7-2600 with 8GB of main memory. Images were captured by Canon EOS 400D with Canon EF-S 18-55mm f/3.5-5.6 Lens. The following subsections describe the details.

4.1 Image acquirement and modeling

We tested several objects including those have similar pattern, non-uniform shape, partially lack of feature. The objects we tested are shown in Figure 3. Table 1 shows the detailed information of input images and the results. In our experiments, all input images' resolution is 2816 x 1880.

Figure 3. Testing objects; from left to right is Asahi, Dinosaur, Penguin, and Gouf.

Table 1. Input images' information.

	Asahi	Dinosaur	Penguin	Gouf
Total input (#images)	35	45	39	73
Successful estimation (#images)	29	34	31	59
Far distance (#images)	6	11	8	14
Lack of geometry (#images)	1	0	0	1
Average features (#points)	7,581	6,277	7,742	2,101
Reconstructed points (#points)	458,239	243,319	311,251	321,965
Points after cleaning (#points)	423,026	180,385	295,024	258,463
Vertices of mesh (#vertices)	557,781	172,272	589,152	223,810
Face of mesh (#faces)	1,113,920	344,540	1,178,272	447,572

Asahi: Asahi is an ideal object for testing horizontal hint because its shape is a simple cylinder with rich features, it also has a similar pattern between 180 but it has poor features at top face so that it was frequently failed to estimate CP on replenish hint. It has three failures in horizontal moving because that is impossible for user moves to instructed position rather precisely. Figure 4 (a) shows our system can effectively guide user to capture images. The result

also shows our system can successfully estimate CP from the object which has a similar pattern. The result after performing PMVS is shown in Figure 5 (a), and the triangle mesh after performing Fourier surface reconstruction [22] is shown in Figure 5 (e).

Dinosaur: Dinosaur is a complex object with various surface textures and rich features; it is suitable for testing system's error toleration and flexibility since it has a non-uniform shape, i.e., it needs to be shot densely in both front and rear side, otherwise it cannot estimate CP. As shown in Figure 4 (b), we notice failures are all gathering at front and rear side where it has narrow shape with less features. When failed to estimate CP, system guides user to shoot closer by decreasing hint's azimuth angle, related distance, and the length of direction arrow and letting user rapidly find appropriate position. The result after performing PMVS is shown in Figure 5 (b), and the triangle mesh after performing Fourier surface reconstruction is shown in Figure 5 (f).
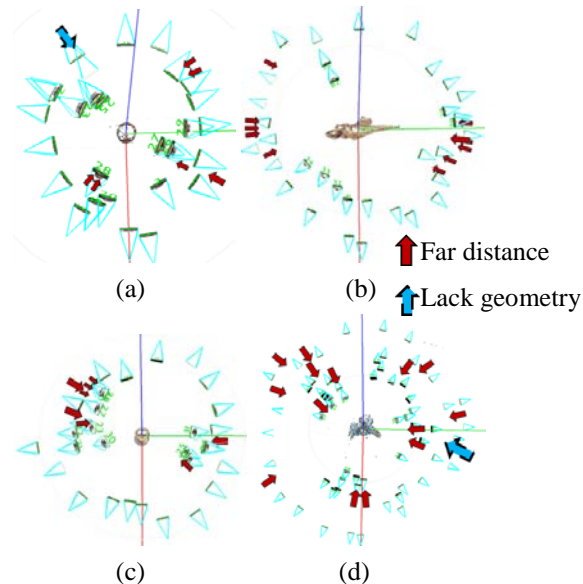

Far distance

Lack geometry

Figure 4. The estimated CP position of (a) Asahi; (b) Dinosaur; (c) Penguin; (d) Gouf. The arrow points out where the errors accrue.

Penguin: Penguin is an object which has non-uniform features; it has rich features at middle around object and poor features on upper/lower side. In this object, there are five failures in replenish since it has poor features on upper side. Figure 4 (c) shows failures are all gathering at both sides of the wings where it has less features. Even though the object is partially lack of features, our system still successfully estimated CP; as shown in Figure 5 (c), it did not generate 3D points on upper and lower side because PMVS is not suitable for smooth surface. Since we have images' CP, it can use other algorithm, like *shape from silhouettes* or *voxel coloring*, to generate 3D geometry. The triangle mesh after performing Fourier surface reconstruction is shown in Figure 5 (g).
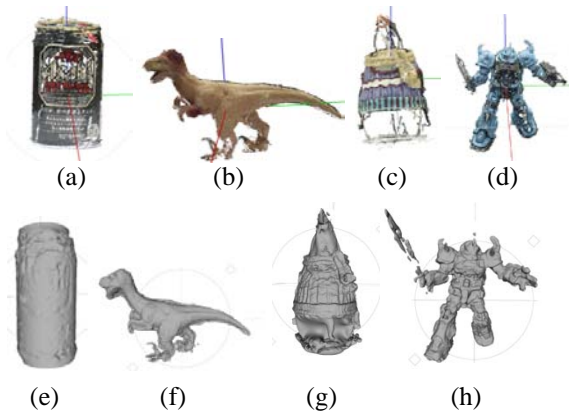

Figure 5. Dense geometry and triangle mesh of (a)(e) Asahi; (b)(f) Dinosaur; (c)(g) Penguin; (d)(h) Gouf.

Gouf: Gouf is a complex object too. Unlike Dinosaur, Gouf does not have various surface texture so that it has less feature points than Dinosaur's. The characteristic is that it has a frame for standing upper table so that it can capture images below horizontal line. In this object, there have forty-two images for replenish that is more than other objects; not only because it can be shot below horizontal, but also it is an lanky object so that it needs to be taken more images from horizontal to replenish target. The result after performing PMVS is shown in Figure 5 (d), and the triangle mesh after performing Fourier surface reconstruction is shown in Figure 5 (h).

4.2 Size estimation

To test the accuracy of our size estimation, we placed a ruler next to objects and modeling objects using turntable and tripod, then measured the ruler after performing PMVS to generate dense 3D geometry. We measured the reconstructed ruler per centimeter ten times, then averaged the results for reducing measurement error. As shown in Table 2, the result shows our system can reasonably estimate object's size. The average accuracy of size estimate is 86.8%. The Coca-Cola 1 and Coca-Cola 2 are the same object in different distance and altitude so that they have different accuracy. Since the distance, focal length, altitude etc. will affect the accuracy, here we tested the relation between distance and focal length; we first fixed the distance between camera and object to test the effect of focal length, then fixed focal length to test the effect of distance.

Table 3 shows that the focal length has a small effect to size estimation's accuracy, but the distance between camera and object has a significant effect to accuracy. Since Bundler, PMVS and our work all use pinhole camera model for estimation, there have a discrepancy between pinhole camera and real camera that cause inaccurate size estimation; furthermore, size estimation also involves viewpoint of camera, CCD/CMOS size, camera parameters estimation,

modeling algorithm etc., we think it has room for improvement in the future.

Table 2. Result of size estimation; "Object's Height" is measure by ruler in real world; red font number indicates the maximum of average error; green font number indicates the minimum of average error.

	Distance	Focal Length	Object's Height	Accuracy (per centimeter)
Asahi	0.46m	55mm	122.48mm	88.1%
Dinosaur	0.77m	55mm	143.59mm	92.4%
Penguin 1	0.35m	55mm	94.92mm	81.3%
Gouf	0.59m	41mm	195.64mm	92.3%
Coca-Cola 1	0.35m	38mm	116.68mm	**74.5%**
Coca-Cola 2	0.27m	37mm	116.68mm	**97.0%**
			Average Accuracy: 86.8%	

Table 3. The result of testing the relationship between distance and focal length; "Object's Height" is measured by ruler in real world.

	Distance	Focal Length	Object's Height	Accuracy (per centimeter)
Coca-Cola 3	0.27m	18mm	116.68mm	80.0%
Coca-Cola 4	0.27m	28mm	116.68mm	81.9%
Coca-Cola 5	0.27m	35mm	116.68mm	76.2%
Coca-Cola 6	0.35m	35mm	116.68mm	98.8%
Coca-Cola 7	0.46m	35mm	116.68mm	65.2%
Penguin 2	0.27m	18mm	94.92mm	66.2%
Penguin 3	0.27m	28mm	94.92mm	64.7%
Penguin 4	0.27m	35mm	94.92mm	61.5%
Penguin 5	0.35m	35mm	94.92mm	84.3%
Penguin 6	0.46m	35mm	94.92mm	96.7%

5. CONCLUSION AND FUTURE WORK

We present an interactive system that guides user to capture images for modeling and displaying real world information such as related position and object size. Through our system, user can get sufficient images for modeling and timely replenish insufficient images. Moreover, our system integrates PMVS to generate dense point geometry and provides simple but effective cleaning tools for cleaning out noise points. After we obtain object's 3D geometry, our system can estimate object's size to let user know the size of object. In the future, we want to combine other algorithm like *shape from silhouette* and *voxel coloring* for modeling the objects which do not have various surface textures; besides, we will try to use a *dynamic shape* to replace icosahedron for scoring and displaying existing information, then use octree to divide space for calculating the density of points as score value for providing more precise prediction of appropriate position where user needs to capture more images.

6. REFERENCES

[1] Furukawa, Y., and Ponce, J. Accurate, dense, and robust multi-view stereopsis. IEEE Transactions on Pattern Analysis and Machine Intelligence 32, pp. 1362–1376, 2010.

[2] Baumberg A., Lyons A., and Taylor R. 3D S.O.M.—A commercial software solution to 3D scanning. Graphical Models 67, No. 6, pp. 476-495, 2005.

[3] Hua S., and Liu T. Realistic 3D reconstruction from two uncalibrated views. International Journal of Computer Science and Network Security 7, No. 6, pp. 178-183, 2007.

[4] Liu Y., Dai Q., and Xu W. A point-cloud-based multiview stereo algorithm for free-viewpoint video. IEEE Transactions Visualization and Computer Graphics 12, No. 3, pp. 407-418, 2010.

[5] Snavely N., Seitz S. M., and Szeliski R. Modeling the world from Internet photo collections. International Journal of Computer Vision 80, No. 2, pp. 189-210, 2007.

[6] Furukawa Y., and Ponce J. Accurate camera calibration from multi-View stereo and bundle adjustment. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008), pp. 1-8, 2008.

[7] Irschara A., Zach C., Frahm J.M., and Bischof H. From structure-from-motion point clouds to fast location recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009), pp. 2599-2606, 2009.

[8] Shu B., Li T., Qiu X., and Wang Z. An automatic image based modeling system by patch growing. In: Proceedings of Computer Graphics International Conference (CGI 2009), pp. 83-88, 2009.

[9] Furukawa Y., Curless B., Seitz S. M., and Szeliski R. Towards Internet-scale multi-view stereo. In: Proceedings of IEEE Conference Computer Vision and Pattern Recognition (CVPR 2010), pp. 1434-1441, 2010.

[10] Nguyen M. H., Wünsche B., Delmas P., and Lutteroth C. Realistic 3D scene reconstruction from unsconstrained and uncalibrated images taken with a handheld camera. In: Proceedings of the International Conference on Computer Graphics Theory and Applications (GRAPP 2011), pp. 67-75, 2011.

[11] Kim K., Sugiura T., Torii A., Sugimoto S., and Okutomi M. Instant surface reconstruction for incremental SfM. In: Proceedings of the IAPR Conference on Machine Vision Applications (MVA 2013), pp. 371-374, 2013.

[12] Clark J. Object digitization for everyone. Computer 44, No. 10, pp. 81-83, 2011.

[13] Nguyen H. M., Wünsche B., and Delmas P. 3D models from the black box: investigating the current state of image-based modeling. In: Proceedings of International Conference on Computer Graphics, Visualization and Computer Vision 2012 (WSCG 2012), 2012.

[14] Simon G. Immersive image-based modeling of polyhedral scenes. In: Proceedings of the IEEE/ACM International Symposium on Mixed and Augmented Reality, pp. 215-216, 2009.

[15] Kim K., Lepetit V., and Woo W. Real-time interactive modeling and scalable multiple object tracking for AR. Computers & Graphics 36, No. 8, pp. 945-954, 2012.

[16] Quan L., Tan P., Zeng G., Yuan L., Wang J., and Kang S. B. Image-based plant modeling. ACM Transactions on Graphics 25, No. 3, pp. 599-604, 2006.

[17] Fudono K., Sato T., and Yokoya N. Interactive 3-D modeling system using a hand-held video camera. In: Proceedings of 14th Scandinavian Conference on Image Analysis, pp. 1248-1258, 2005.

[18] Pan Q., Reitmayr G., and Drummond T.W. Interactive model reconstruction with user guidance, In: Proceedings of IEEE International Symposium on Mixed and Augmented Reality, pp. 209-210, 2009.

[19] Pan Q., Reitmayr G., and Drummond T.W. ProFORMA: probabilistic feature-based on-line rapid model acquisition. In: Proceedings of British Machine Vision Conference, 2009.

[20] Du H., Henry P., Ren X., Cheng M., Goldman D. B, Seitz S. M., and Fox D. Interactive 3D modeling of indoor environments with a consumer depth camera. In: Proceedings of 13th International Conference on Ubiquitous Computing, pp. 75-84, 2011.

[21] Snavely N., Seitz S. M., and Szeliski R. Photo tourism: exploring image collections in 3D, ACM Transactions on Graphics 25, No. 3, pp. 835-846, 2006.

[22] Kazhdan M. Reconstruction of solid models from oriented point sets. In: Proceedings of the Third Eurographics Symposium on Geometry Processing, pp. 73-82, 2005.

[23] Berger M., Levine J., Nonato L. G., Taubin G., and Silva C. T. An end-to-end framework for evaluating surface reconstruction. SCI Technical Report UUSCI-2011-001. SCI Institute, University of Utah, 2011.

[24] Wandel M. Exif JPEG Header Manipulation Tool. http://www.sentex.net/~mwandel/jhead/ (accessed Jan. 6 2014), 2013.