








Automatic and Improved Camera Information Processing for a 3D Photo-Modelling System

Tsung-Chien Wu¹ , Jiing-Yih Lai² , Douglas W. Wang³ , Chao-Yaug Liao⁴  and Ju-Yi Lee⁵ 

¹ National Central University, Taoyuan, Taiwan, rabbit94577@gmail.com

² National Central University, Taoyuan, Taiwan, jylai@ncu.edu.tw

³ Ortery Technologies, Inc., Taiwan, dwmwang@gmail.com

⁴ National Central University, Taoyuan, Taiwan, cyliao@ncu.edu.tw

⁵ National Central University, Taoyuan, Taiwan, juylee@ncu.edu.tw

Corresponding author: Jiing-Yih Lai, jylai@ncu.edu.tw

Abstract. In this study, we propose a process of photo capturing, improved camera calibration, and improved camera alignment to generate a three-dimensional (3D) textured model of an object from its multiple two-dimensional (2D) images for presentation on a website. The proposed photo capturing can represent the object and mat photos in a controlled environment and can easily remove the background color. The mat photos are used in camera calibration to evaluate the camera parameters. The object photos and camera parameters are then used in model reconstruction to generate a 3D mesh model. Camera alignment is employed to obtain the bottom information of the object by flipping the object to take another layer of photos. The accuracy and efficiency of camera calibration and alignment are both improved for automatic generation. Finally, we describe an integrated software system for modelling to demonstrate the entire process, which can yield a high-quality 3D textured model for object presentation. Six realistic examples were presented to demonstrate the feasibility of the proposed system.

Keywords: Camera Calibration, Camera Alignment, 3D Textured Model

DOI: <https://doi.org/10.14733/cadaps.2020.124-137>

1 INTRODUCTION

Two-dimensional (2D) photos are commonly used for product presentation in e-commerce because they can reveal an object's high-quality texture and are easy to process. However, this requires an efficient (automatically) and high-quality collection of the object photos. The photo-taking device and corresponding software play an important role in the e-commerce industry [15]. In addition, as 2D photos only represent discrete views of an object, it is often necessary to capture hundreds of 2D photos with the photo-taking system and orient a photo at a given viewing angle via a viewing interface on the website [15]. However, storing and displaying so many photos while maintaining high photo quality on a website requires significant memory and CPU time for data downloading. In addition, the actual three-dimensional (3D) shape and dimensions of an object cannot be obtained in this representation. Orienting the object photos on the website is not continuous as only a limited number of views are recorded. A technique known as 3D photo-modelling [11], [15-16], [21] was developed to reconstruct the 3D model of an object using multiple 2D photos, while maintaining its texture on the model (called 3D textured model hereafter) [21]. This technology is an alternative option for product presentation because a 3D textured model requires less memory and can be freely oriented in 3D space.

High-quality object photos can be obtained using either a single-camera device (Figure 1(a)) that applies a digital single-lens reflex (DSLR) camera with a camera rail, or a multi-camera device (Figure 1(b)) that applies several DSLR cameras mounted on an arm to capture an object placed on a turntable from different latitudes. These devices are integrated with photo capturing software (Figure 1(c)) to form a photo capturing system. This system can position the camera precisely and orient the object on the turntable to different angles to capture the object photos in different views. It also provides a controlled environment, e.g., a single background color and adjustable lighting, for easily separating the object from the background. As this system is already used in the field of product presentation, we use it here as the photo capturing source of the 3D photo-modelling technology.

The following kinds of data need to be obtained for the 3D photo-modelling technology: object and camera calibration mat photos, camera parameters, a 3D mesh model, and texture data. The object photos are employed to generate the 3D textured model. The mat photos are employed in camera calibration to evaluate camera parameters. The camera parameters are mainly used to transform 2D contour points extracted from each object photo into a 3D space on which the 3D mesh model is generated. The 3D mesh model describes the object's surface geometry in triangular meshes. The texture data covered on the 3D mesh model denotes the 3D textured model. The photo capturing is performed by a capturing system [15] to obtain high-quality object and mat photos. The camera calibration can be performed by matching the feature points between mat photos and the actual mat dimensions to figure out camera parameters [10-12], [23-24].

The shape-from-silhouette (SFS) approach is commonly used to estimate an object's shape from images of its silhouettes [4], [13], [22]. This method applies multiple 2D photos to generate the visual hull of the object. The concept of visual hull generation is used to apply multiple sets of an object's silhouettes to perform polygon intersection. Based on a sufficient number of photos from different views, this method can yield an approximate model to describe the outline shape of an object. However, the quality of this model is not sufficient for 3D visualization for the following reasons. First, the existence of virtual features, such as sharp edges and artifacts, usually affect the accuracy of the model; some virtual features may even be sufficiently large to deteriorate the outline shape of the model. Second, invisible concave features on the photos usually result in redundant convex shapes on the model. The removal of artifacts is particularly important because they are difficult to detect and eliminate. Therefore, a quality improvement method must be implemented to remove virtual features while also recovering the smoothness [16].

Texture mapping generally involves five main techniques: mesh partitioning, mesh parameterization, texture transferring, correction, and optimization. These techniques relate to each other and affect the texture quality. Research in mesh partitioning can be summarized using several different approaches. Shamir [20] categorized several methods of mesh partitioning according to segmentation type, partitioning technique, and segmentation criterion. Segmentation type can be divided into surface-type and part-type. Surface-type mesh partitioning is commonly used in texture mapping [8], [17-18] because it can prevent large distortions in mesh parameterization. Mesh parameterization can be classified in terms of mesh distortion minimization, fixed or free boundary, and numerical complexity [6], [19]. For texture map generation, the main idea is to overcome the texture transferring problem. Niem et al. [14] proposed texturing and grouping the meshes, using the camera information to identify the most appropriate image source. They also proposed an algorithm to minimize the color inconsistency at the transition of two different groups and synthesize the invisible meshes using the neighboring color. Genç and Atalay [5] proposed a method to extract the pixels and render the texture dynamically. The extraction is performed by scanning the pixels horizontally and rendering every color onto the meshes. Baumberg [3] proposed an algorithm to process the color difference from two different images using a blending method. The images were filtered into high and low bands, where the low band images were averaged to minimize the color difference, and the high band images were kept to maintain the outline of the object boundary.

2 PROBLEM STATEMENT

For 3D photo-modelling of an object in e-commerce, developing an integrated system that converts an object to photos, and then to a 3D textured model for presentation is an important research focus. The ideal 3D textured model of the object can be reconstructed consistently and smoothly. However, the photo-taking device and lighting environment, photo post-processing, camera calibration, and camera alignment are also important issues because the quality of the object photos affects the photo post-processing, which in turn affects the result of camera calibration. In addition, the object must be flipped to another position to obtain information about its bottom. Camera alignment should be performed to acquire the information for another layer of cameras. The similarity between the real object and 3D textured model is the scoring factor of the 3D photo-modelling technology. Thus, the accuracy of the camera information affects the accuracy of the 3D textured model. Any defect on the 3D textured model could negatively impact the perception of the product being presented.

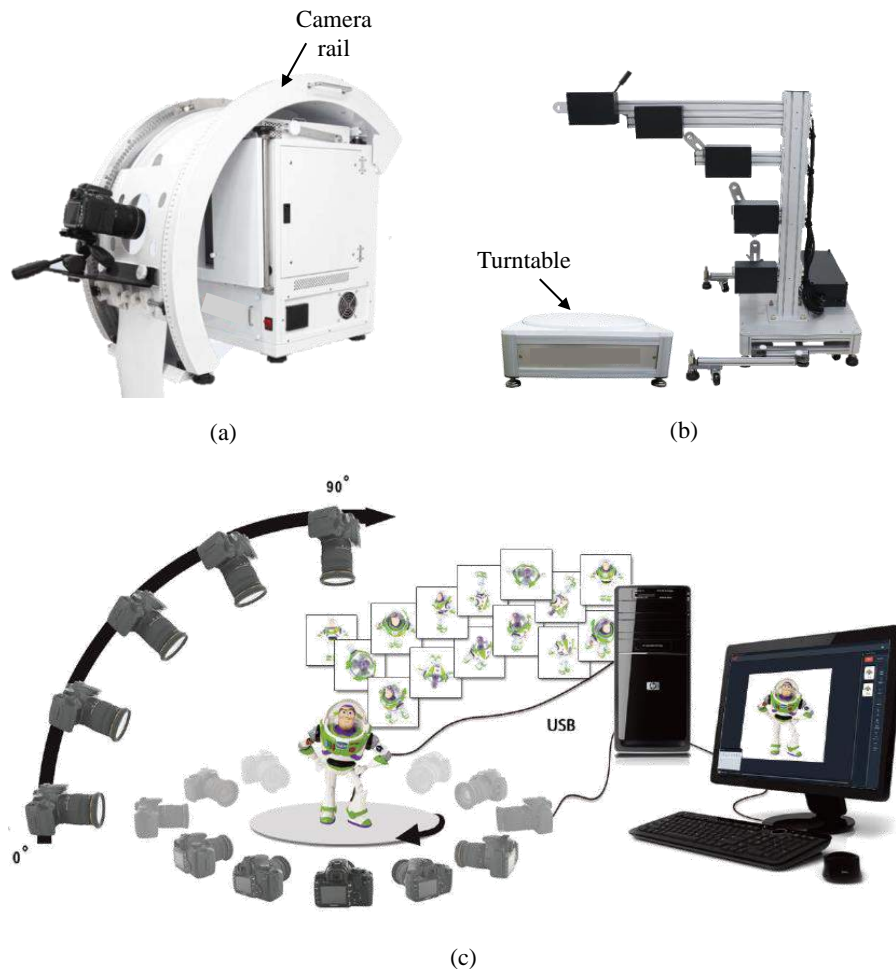


Figure 1: Photo capturing device and system: (a) single camera with camera rail, (b) multiple camera arm, and (c) photo capturing software.

Typical problems involved in this technology include the following:

1) Unskilled photo capturing: The photos are not captured with appropriate camera settings because most users are not professional photographers. The background and the object image must be separated on each photo. A noisy or erroneous boundary on the separated object image may increase the error of the 3D reconstructed model. Therefore, a controlled photo capturing environment is necessary to ensure the correct separation of the object boundary.

2) Complex photo post-processing processes: Several post-processing processes are required to deal with the original object photos, such as background removal, thresholding, and edge detection. These processes are generally complex and tedious for the user. Therefore, an automatic photo post-processing algorithm is necessary to handle these issues.

3) Camera calibration: This requires a series of feature detection and calculation in different stages to generate camera information, which includes feature detection, feature decoding, and feature matching. In each step, the features must be detected accurately and automatically to obtain suitable camera information.

4) Camera alignment: The information for the bottom and texture of the model is obtained by capturing the flipped object photos. Once the object is moved, the entire coordinate system is changed. A camera alignment process is therefore necessary to combine the two different coordinate systems for application in model reconstruction and texture generation.

The camera calibration method was developed by Liao et al. [10]. However, this method requires a user's input to adjust the thresholds for different cases of calibrated mat photos. In this study, an automatic process is proposed to generate a 3D textured model, which includes automatic setting of all thresholds in camera calibration. On the other hand, the camera alignment method was also developed by Liao et al. [11]. However, its accuracy was not satisfied when significant artifact exists on the SFS model. The computational efficiency

was also unsatisfied owing to substantial computation required to align different pairs of model and object photo profiles in multiple views.

This study develops algorithms for processing camera information to yield background removed and well lighted object photos and camera parameters for generating 3D textured models. The camera information is evaluated using a system's workflow; this workflow requires a photo capturing device and accompanying software, and 3D photo-modelling software to generate the 3D textured model. The issues addressed in this study are a part of the system's workflow, which primarily comprises the following techniques: capturing object photos from multiple views, pre-processing object photos, calibrating the camera, and aligning the camera coordinates in multiple views. To simplify manual operation, these techniques are integrated into two software workflows. Six realistic examples are presented to demonstrate the feasibility of the proposed 3D photo-modelling system.

The main contribution of this study is that it describes the technologies involved for camera information—photo capturing, photo pre-processing, camera calibration, and camera alignment—for generating and presenting 3D textured models. The entire procedure from photo capturing to 3D model generation is almost fully automatic, except for the camera alignment, which requires manual operation to change the pose of an object. The proposed system is more flexible and suitable for unskilled photographers to deal with various kinds of realistic objects.

3 THE PROPOSED CAMERA INFORMATION PROCESSING

The proposed camera information processing is integrated with a photo capturing device and software for generating a 3D textured model of an object captured in multiple viewing angles. The main techniques involved are photo capturing, photo pre-processing, camera calibration, and camera alignment. Figure 2 shows the overall flowchart of the proposed camera information processing. The photo capturing device can capture a series of photos for the calibration mat and the object under a controlled environment (Figure 2(a)). The mat photos are employed in the camera calibration procedure to evaluate a set of camera parameters for each viewing angle (Figure 2(b)). Pre-processing of object photos and calibration mats is required to remove the background color and detect the mat features, respectively (Figure 2(c)). The first set of camera information is calculated by applying the mat information (Figure 2(d)). An initial mesh model can be generated by using the first set of object photos and camera parameters with an SFS method [16]. The camera information of the flipped object is then calculated by applying the initial mesh model and the silhouette of the flipped object (Figure 2(e)).

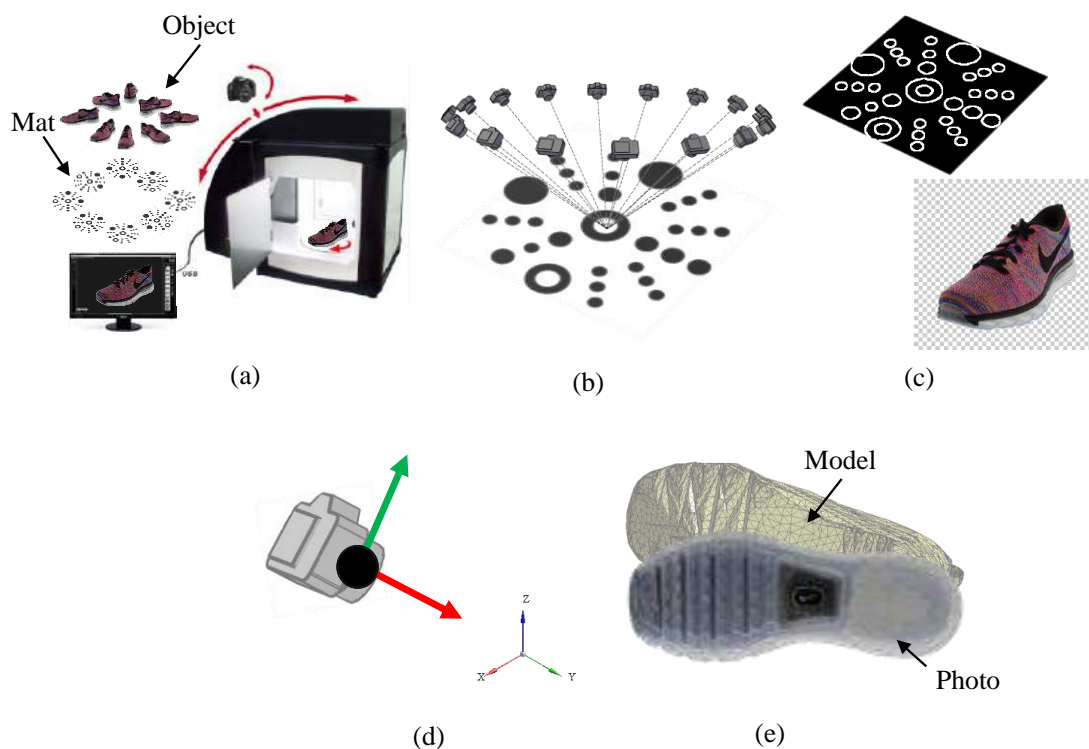


Figure 2: Overall processes of camera information processing: (a) photo capturing, (b) camera calibration for each camera, (c) mat and object photo processing, (d) camera information, and (e) camera alignment.

3.1 Photo Capturing

The main idea of photo capturing is to sequentially capture the object and mat photos. Traditionally, the object and mat photos are captured in an arbitrary environment, and silhouettes of the object are manually extracted. However, this procedure requires tedious manual work and is time-consuming. For automatic extraction of the silhouettes, a controlled lighting environment should be provided to enhance the difference between the object image and the background, especially the color at the object boundary.

The proposed photo capturing system is composed of a capturing device, a control, and processing software [15]. The capturing device contains a set of lighting equipment and a turntable. The lighting equipment can build up an environment to make the object more appealing for the consumers. The turntable is equipped to rotate the object to capture the photos in 360°. However, an interface is needed to control the features of this integrated device, which is achieved by the integrated software. This software can also remove the background of the object to create a transparent or pure color background for different applications.

Figure 3 depicts the components and control device of the photo capturing device. The components include the front, back, left, and right diffusion panels. These panels are the light source for the capturing device. They can adjust the lighting environment of the device, and they can be controlled separately to set different directions of lighting strength. The turntable driving mechanism can drive the turntable to rotate the desired angles. All of the components are controlled by two control boards, one for the turntable, and one for the lighting panels. The control signal is sent from the integrated software, which can control the device components using a separated module in a user-friendly interface.

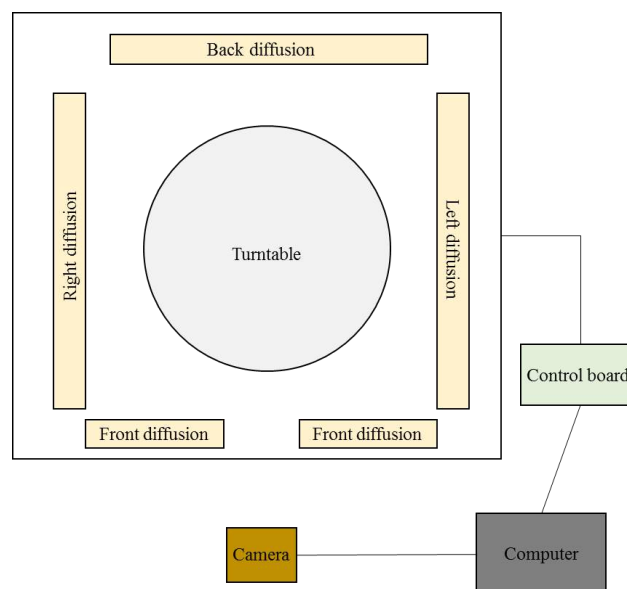


Figure 3: The components of the photo capturing device.

This controlling software is also bundled with a background-removing feature to generate a transparent background photo. The purpose of this feature is to retain the perfect color of the object and remove the background color. Figure 4 depicts the background-removing interface, including several parameters for setting, e.g., single or multiple loops, and removing strength. It can also preview the result after removing the background. Once the parameters are set, the object photos are captured, with the background color removed automatically. Further, the software also provides a tool to recover the background in case the result after removing the background is unsatisfactory.

The object photos are captured sequentially in a 360° rotation for the object placed on the turntable, which can yield a row of object photos. The orientation of the object on the turntable can also be changed to obtain additional rows of object photos. These can provide additional data to more accurately describe the object shape on the top and bottom. Figure 5(a) shows two rows of photo capturing for a shoe, enabling the description of the bottom of the shoe. For mat photos, each photo should be captured at the same angle as that of the corresponding object photo. Unlike for object photos, it is unnecessary to remove the background of mat photos. Figure 5(b) depicts a typical mat photo. The object photos are saved as a *.png file format to offer additional information for background removal, while the mat photos are saved as a *.jpg file format.

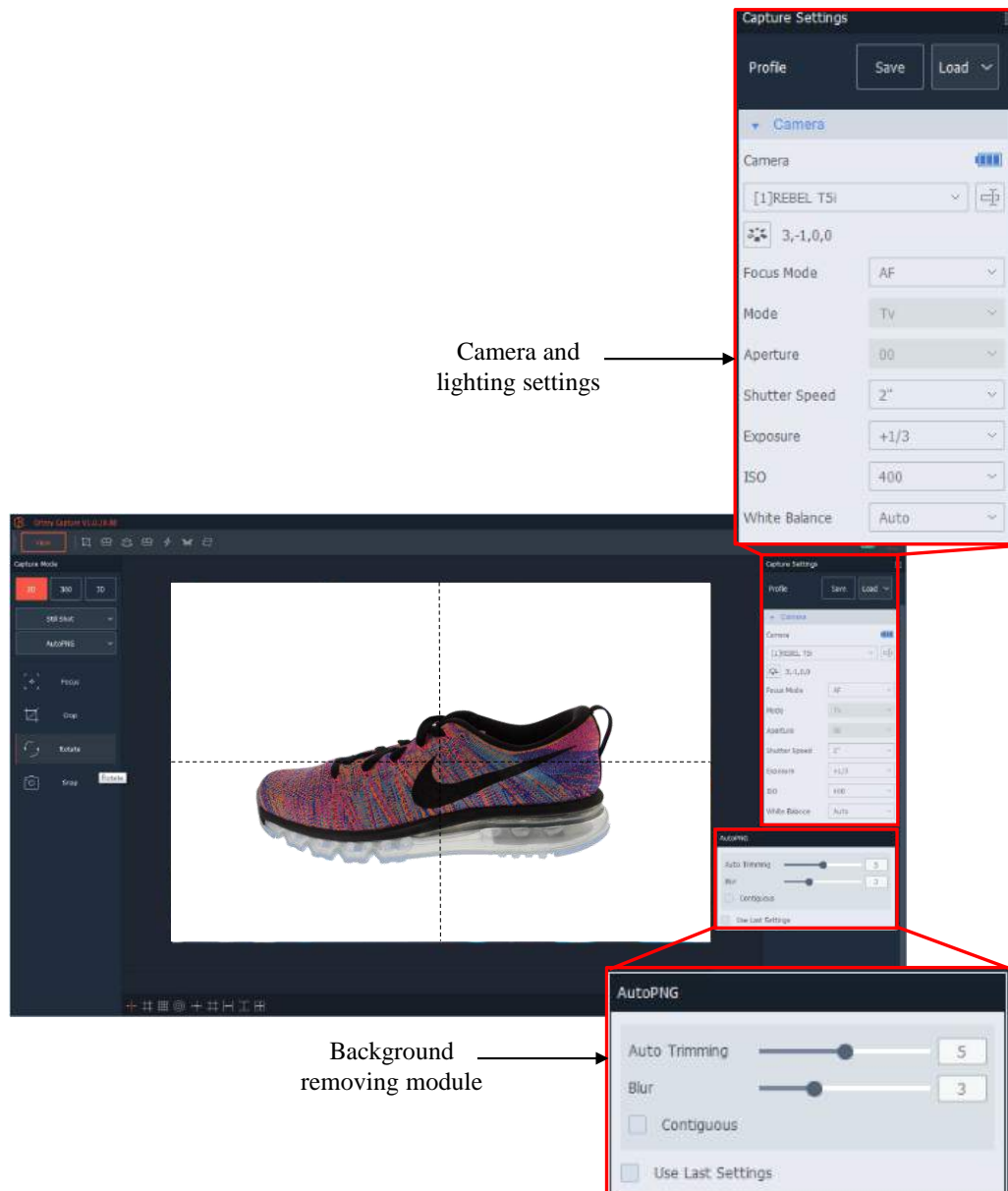


Figure 4: The interface for capturing the object photo and removing the background.

3.2 Camera Calibration

The basic idea of camera calibration is to find the corresponding feature points between each mat photo and the actual mat dimensions, and to solve the extrinsic and intrinsic matrices to obtain the camera parameters on each viewing angle [10]. As each mat photo and object photo are captured at the same viewing angle, the camera parameters at each viewing angle can be used to generate the position and orientation of the corresponding object photo in 3D space. All object photos in 3D space can thus be related, allowing the 3D points on the model to be calculated.

The first step of camera calibration is thresholding the mat photos to distinguish the feature points from the background. As the purpose of this step is to filter out the background noise, the threshold is very important. For the automatic workflow, an automatic thresholding process has been developed. The original mat photos are first converted into grayscale; the automatic thresholding can determine the filtering value on each photo by detecting its highest grayscale value. The threshold T_B of a mat photo is set to 80% of the maximum grayscale value, as shown below

$$T_B = G(Max) \times 0.8, \quad (3.1)$$

where $G(Max)$ denotes the maximum grayscale value on a mat photo. Once the threshold value T_B is determined, each mat photo can be converted individually into a binary photo, as shown in Figure 6(a).

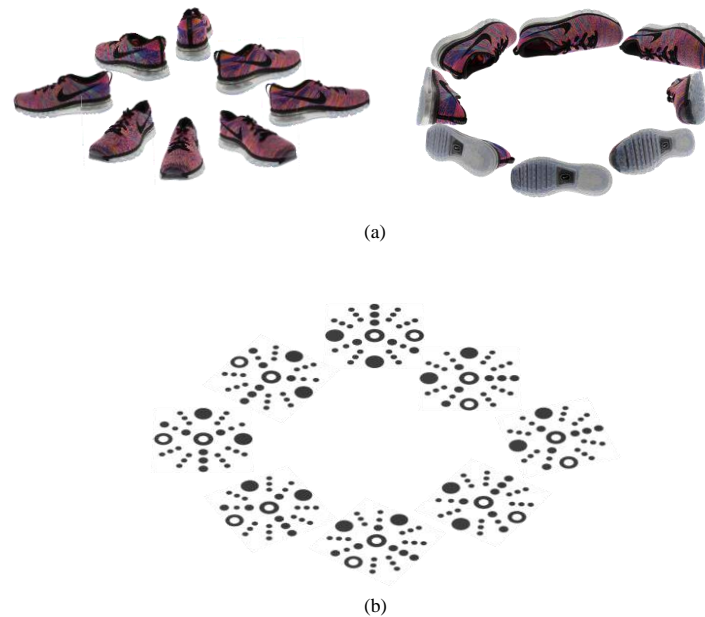


Figure 5: Photo source for 3D photo-modeling: (a) two rows of object photos, and (b) camera calibration mat photos.

The next step is to recognize feature points; each feature point is recognized as an ellipse. The original method calculates the distance between the center of the bounding box and the corner vertex [10]. Previously, if the distance is less than a threshold, the corresponding feature is detected as an ellipse. However, for the automatic procedure, the past criterion of this detection is not robust enough. In this study, there are two new criteria for detecting the ellipse. The first criterion compares the area difference between the bounding box and the feature. The second criterion compares the perimeter difference between the bounding box and the feature. If both values are less than two respective thresholds, the feature is detected as an ellipse, as shown in Figure 6(b).

The third step involves decoding the features by calculating the distance of each center of the ellipse to the central ellipse on the 2D domain. Each feature point is assigned a number, which matches the corresponding feature point on the ideal mat. In the original decoding procedure [10], the number of total features recognized was not checked, thus resulting in an insufficient number of features for calibration. Therefore, in the proposed method, a checking process has been added to make sure the feature number is correct. Because the features on the mat have an arrangement, the system can continuously check the number of detected ellipses on adjacent groups to ensure the detected features are correct. Once all features are detected, each of the features is assigned a corresponding number, as shown in Figure 6(c).

The feature points are matched one by one according to their assigned number, as shown in Figure 6(d). According to actual mat dimensions, the calibrated dimensions can be determined by this process. After matching the feature points, the camera parameters can be evaluated using actual and theoretical mat dimensions. However, various errors, such as noise on the photos, errors on feature points, and errors on circle fitting, may cause deviations in the estimated camera parameters from their true values. Therefore, optimization is performed to simultaneously minimize the errors for all mat photos, yielding optimized camera parameters. The objective function F_{obj} of this optimization problem is formulated as follows:

$$F_{obj} = \sum_{i=1}^n \sum_{j=1}^m \|m_{ij} - \hat{m}(A, R_i, t_i, m_j)\|^2, \quad (3.2)$$

where n is the number of mat photos, m is the number of feature points on a single mat photo, m_{ij} is the j^{th} feature point extracted from the i^{th} photo, and $\hat{m}(A, R_i, t_i, M_j)$ is the j^{th} feature point M_j transformed to the camera coordinate by the extrinsic and intrinsic matrices of the i^{th} photo. Equation (3.1) can be solved using the Levenberg-Marquardt method [12]. By applying the aforementioned method, the final extrinsic and intrinsic matrices can be obtained. The camera parameters can be collected by decomposing these two matrices. The extrinsic matrix T_{ext} is shown in Equation (3.2)

$$T_{ext} = \begin{bmatrix} r_{11}r_{12}r_{13}t_x \\ r_{21}r_{22}r_{23}t_y \\ r_{31}r_{32}r_{33}t_z \end{bmatrix}, \quad (3.3)$$

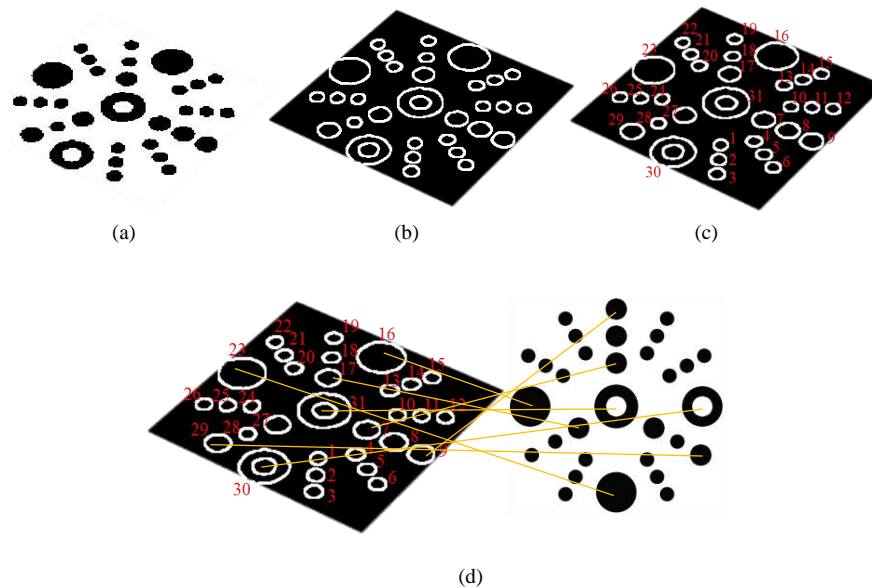


Figure 6: Image processing for camera calibration: (a) image thresholding, (b) feature extraction, (c) feature encoding, and (d) feature matching.

where r_{11} to r_{33} denote the coefficients related to the rotational matrix, and t_x to t_z denote the coefficients related to the translational vector. The camera position, viewing vector, and vertical vector can be evaluated by applying these two types of parameters. The camera position is evaluated using the rotational matrix and the translational vector, whereas the camera viewing and vertical vector are evaluated using only the rotational matrix. The intrinsic matrix T_{int} is shown in Equation (3.3)

$$T_{int} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (3.4)$$

where α is the ratio parameter of the u direction in camera coordinates, β is the ratio parameter of the v direction in camera coordinates, γ is the angle between the u and v directions (also called the distortion parameter), and (u_0, v_0) is the center of the photo. The camera field of view (FOV) in the x and y directions and the focal length can be evaluated using the intrinsic matrix. The FOV is evaluated by the width and height of a photo, which are evaluated by α and β , respectively. The focal length is evaluated by α , the width, and the aperture width of the photo.

The silhouette extraction is also processed in this phase. The object photos are the transparent background photos. The photo can be transformed into a binary image by the value of the alpha channel in a *.png file. The silhouette information can be extracted using the boundary extraction method. A series of pixel data representing the object silhouette can be obtained. The camera parameters and the silhouette data can be collected using the aforementioned process, as shown in Figure 7.

3.3 Camera Alignment

The purpose of the camera alignment process is to obtain the information about the bottom of the object. However, the object must be flipped to capture photos of its underneath. Once the object is moved, the entire coordinate system must be reset. The camera alignment process can merge the new coordinate system into the original coordinate system so that both sets of object photos use the same coordinate system. A procedure for camera alignment was previously developed [11], but it was not accurate enough. We have therefore developed an improved method, as described below. The basic idea of camera alignment is to apply the original calibrated camera information to evaluate the flipped angle. However, the original camera position cannot be changed when taking the second layer of object photos. The input data is the original calibrated camera information and the initial mesh model generated using the first set of object photos. First, the model is projected onto the 2D domain. The purpose of this projection is to compare the object profile on the flipped photo with that of the projected model. In the original procedure [11], the entire mesh model is projected onto

the 2D domain, which is time-consuming. In the modified method, only meshes near the boundary are projected onto the 2D domain, which can save significant computational time, as shown in Figure 8.

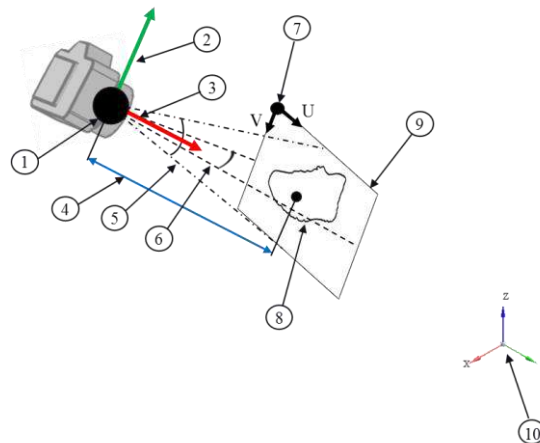


Figure 7: Camera information: (1) camera position, (2) camera vertical vector, (3) camera viewing vector, (4) focal length, (5) camera vertical FOV, (6) camera horizontal FOV, (7) local coordinates, (8) object silhouette, (9) image plane, and (10) global coordinates.

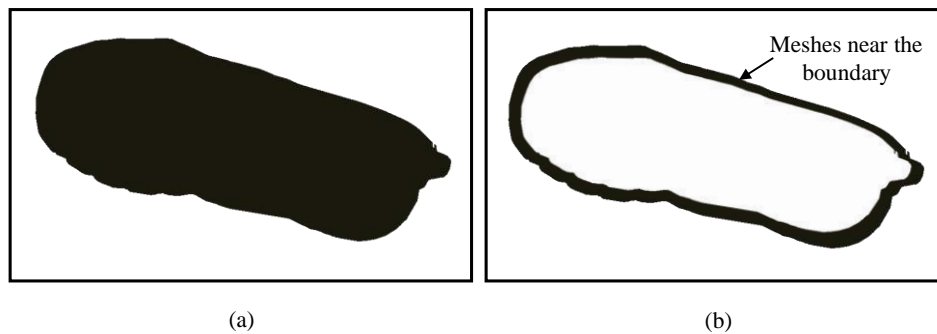


Figure 8: Model projection: (a) original method of projection, and (b) improved method of projection.

The next step is to adjust the model orientation to fit the photos. Before adjusting the orientation, artifacts of the initial mesh model are ignored to prevent possible errors. As artifacts usually occur on the bottom of the mesh model, the meshes with a y coordinate value less than 0 are ignored. Next, the position and orientation of the projected model are adjusted using the iterative closest points (ICP) method [9]. The boundary profile of the projected model and the silhouette of the object photo are respectively regarded as two groups of point cloud. Based on the singular value decomposition (SVD) method [2], a rotational matrix R can be evaluated to align these two sets of data. The equation for calculating the translational vector T for the point cloud of the projected model is shown in Equation (3.4)

$$T = -R \cdot \vec{p} + \vec{q}, \quad (3.5)$$

where T is the translational vector for two sets of data, R is the rotational matrix for two sets of data, \vec{p} is the centroid of the point cloud for the projected model, and \vec{q} is the centroid of the point cloud for the object photo. However, to improve the accuracy of the camera alignment process, the centroid of the point cloud for two sets of data has to be redefined. The original centroid of the point cloud was calculated based on the shape. The ICP accuracy is affected by the error of the center position. Thus, the improved center determination is based on the centroid of the point cloud. When the density of the point cloud is high near a particular region, the centroid of the points cloud can approach that region, as shown in Figure 9.

Through the ICP process, the initial alignment can be performed. However, the alignment is still not accurate enough to merge the two sets of data. The next process is fine-tuning the new set of camera data. The fine-tuning process is based on the Golden Section method [7]. This method can search the best solution by applying the searching section ratio 0.618.

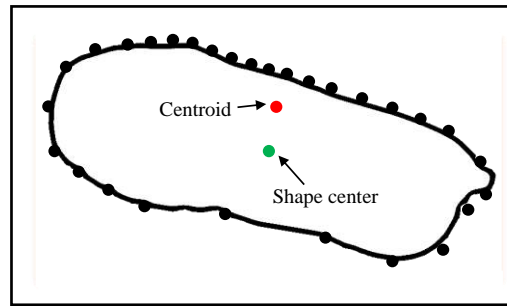


Figure 9: Point cloud center determination improvement.

The equation for calculating the fine-tuning position is shown in Equation (3.5)

$$n \geq \frac{\ln(\frac{\varepsilon}{l})}{\ln 0.618} + 1, \quad (3.6)$$

where n is the maximum iteration time, ε is the minimum alignment error target, and l is the section length for searching the solution. First, the root mean square (RMS) error for two sets of data is the factor for the searching process. However, in the original method [11], all errors were included to determine the RMS error. Some of these might be too large to be considered reasonable errors. To increase the searching accuracy and efficiency, we ignore errors that are larger than a threshold, and to search for the closest points, we consider the computational efficiency. We therefore apply a quad-tree structure to increase the efficiency of the searching process, as shown in Figure 10, where the horizontal and vertical lines are the quad-tree subdivision illustration. When each point on the object photo profile must be compared with all points on the model profile to evaluate the closest point, the overall computational time required would be very large. Instead, the domain of points is subdivided in a quad-tree method until each sub-region contains a point from the object photo profile only. The closest point computation can thus be performed by searching the points from the model profile on its neighboring regions. This can increase the speed of searching considerably.

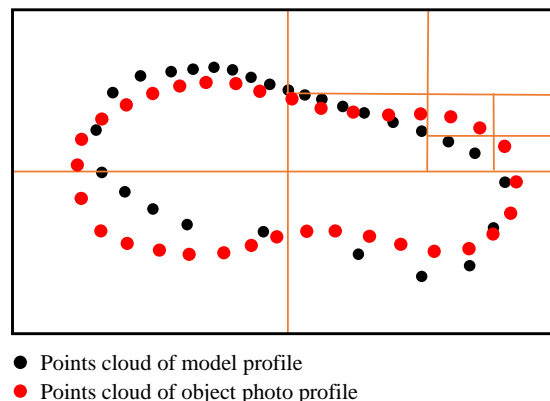


Figure 10: Quad-tree structure to improve the search for the closest point.

4 EXAMPLES AND DISCUSSION

The algorithms of the object photo capturing, camera calibration, and camera alignment are integrated into a capturing software and a modelling software. The modelling software is also combined with (1) an SFS modelling procedure to create an initial mesh model [16], (2) a mesh optimization process to optimize the shape and surface smoothness of the initial mesh model [16], and (3) a texture mapping process to generate the texture map of the 3D textured model [21]. Figure 11 depicts the entire workflow of this integrated system for a shoe example, which shows the screenshot of the process from the object photo capturing to the viewing of the final model. The capturing device can be controlled by this software (Figure 11(a)), such as the setting of the camera, turntable, and lighting. The background removal of object photos can also be implemented using this software. The following plots are the modelling process of the 3D photo-modelling. Following the guidance of the modelling software, the 3D photo-modelling process can be performed step by step to obtain the 3D textured model. The numbers of photos for modelling is set as two rows. Each row of object photos is set as 16 shots. These two rows of photos are captured with the normal pose and lying-down pose, respectively, to obtain sufficient information of the object shape. Figure 11 (b) depicts the camera calibration process. After the mat photos are loaded by the software, the aforementioned improved camera calibration

method is performed. Figure 11 (d) depicts the camera alignment process. To accelerate the convergence of the alignment, the determination of the initial flipped angle is integrated as an operating interface so that the user can rotate the initial mesh model manually to fit with the photos of the flipped object. The model reconstruction, model optimization, and texture mapping processes are shown as Figure 11 (c), (e), and (f), respectively. Figures 12(a) to (f) depict the collected results from the object to the 3D textured model. By applying the proposed improvement method, sufficient information about the model shape, texture, and bottom can be obtained.



Figure 11: Operation workflow for 3D photo-modeling system: (a) object photo capturing software, (b) input mat photos and camera calibration, (c) model reconstruction, (d) camera alignment, (e) model optimization, and (f) texture mapping.

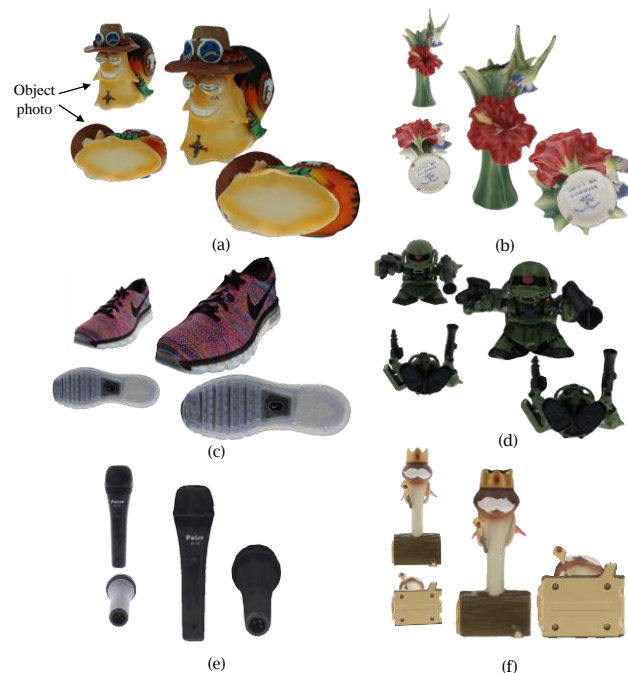


Figure 12: Results of 3D textured model that includes bottom information: (a) snail doll, (b) vase, (c) shoe, (d) Gundam, (e) microphone, and (f) doll.

Table 1 shows the duration of each main step. The simulations were performed on a personal computer with a 3.20 GHz central processing unit (CPU) and 8 gigabytes (GB) of random-access memory (RAM). The model optimization stage takes the longest amount of time because it requires the projection of all 3D vertices onto a 2D domain and the iterative extraction of boundary vertices on the 2D domain for each object photo. The alignment process to determine the camera information for the lying-down pose photos of the object is another longest amount of time consumption stage. The proposed method improves the vertices projection from the 3D domain to the photo domain and controls the mesh number to within 10,000-15,000 meshes. The time cost for the entire process can be controlled to within 2–5 minutes for most examples. Table 2 shows the computational time required for original and improved alignment methods. The computational time for the improved alignment method is reduced in a range from 11.46% to 37.74%. This result indicates that the improved alignment method is obviously better than the original method.

Object	Number of mesh	Calibration time (Sec)	Modeling time (Sec)	Alignment time (Sec)	Optimization time (Sec)	Texturing time (Sec)	Total time (Sec)
Snail doll	15100	10	3	47	83	26	169
Vase	15510	13	5	116	154	18	306
Shoe	10000	18	6	85	158	22	289
Gundam	10000	14	6	57	126	18	221
Microphone	10000	15	2	33	73	22	127
Doll	10000	13	3	67	117	21	178

Table 1: Time consumption for 3D photo-modeling.

Object	Number of mesh	Original time (Sec)	Improved time (Sec)	Reduced time (%)
Shoe	10000	96	85	11.46
Gundam	10000	72	57	20.83
Microphone	10000	53	33	37.74

Table 2: Time comparison between original and improved alignment method.

Figure 13 depicts the distribution of the mesh number on each range of pixel numbers for the following five cases: 8192×8192/OBB, 8192×8192/AABB, 4096×4096/OBB, 4096×4096/AABB and commercial (3DSOM) software [1], where 3DSOM is commercial software. For a texture map of better quality, the number of meshes with fewer numbers of pixels should be as small as possible. It is evident in Figure 13 that the texture resolution of the case 8192×8192/OBB is the best among the five cases, because it has the minimum number of meshes with fewer pixels. In addition, the texture resolution of 3DSOM software is the worst as most of the meshes having pixels less than 2000. Therefore, the texture resolution of the proposed method is better than that of 3DSOM software.

5 CONCLUSIONS

In this study, we proposed a process of photo capturing, improved camera calibration, and improved camera alignment to generate 3D textured models for presentation on a website. The proposed photo capturing device can capture object and mat photos in a controlled environment and easily remove the background color on object photos. The camera calibration can automatically and accurately reconstruct the capturing coordinate system for further applications. The camera alignment can combine two different coordinate systems accurately and efficiently to merge the object photos from two different poses. These processes are combined with mesh model reconstruction, mesh model optimization, and texture mapping to generate the 3D texture model of the object. Figures 12 and 13 indicate that the color pixels on all photos can be mapped onto the texture map correctly and the resolution of the pixels on the original photos can be retained on the texture map, respectively. Also, Table 1 indicates that the overall CPU time required for six examples are within 2-5

minutes. For other more complex examples, the maximum CPU time required will not exceed 10 minutes. Therefore, a high-quality 3D textured model can be output accurately and efficiently by applying this series of processes for presentation in e-commerce. Other than e-commerce application, the proposed 3D textured model could be used in design modification, 3D animation creation, and gaming object creation. In addition, the study of multi-view stereo reconstruction method would be another direction for the reconstruction of 3D textured model. If the photo consistency issue for multiple photos could be solved, the camera information, object model, and the texture map could be further improved.

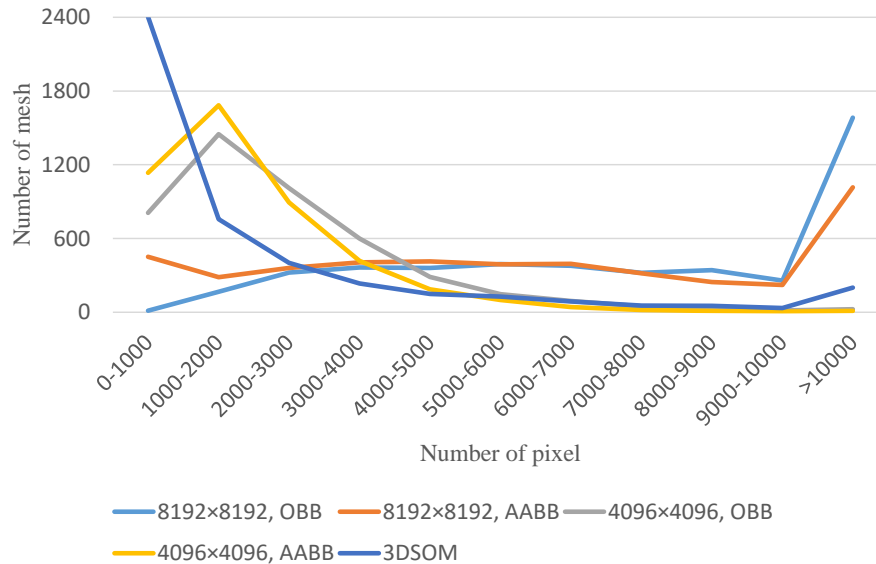


Figure 13: The line chart of mesh number vs. pixel number for five cases: 8192×8192/OBB, 8192×8192/AABB, 4096×4096/OBB, 4096×4096/AABB and 3DSOM software.

Tsung-Chien Wu, <http://orcid.org/0000-0002-2299-7361>

Jiing-Yih Lai, <http://orcid.org/0000-0002-0495-0826>

Douglas W. Wang, <http://orcid.org/0000-0002-8039-5027>

Chao-Yang Liao, <http://orcid.org/0000-0001-8203-9520>

Ju-Yi Lee, <http://orcid.org/0000-0002-2244-4863>

REFERENCES

- [1] 3DSOM. <http://www.3dsom.com/>
- [2] Bretscher, O.: Linear Algebra with Applications, 2nd Edition, Prentice Hall, New York, 2001.
- [3] Baumberg, A.: Blending images for texturing 3D models, BMVC, 3, 2002, 404-413. <http://dx.doi.org/10.5244/C.16.38>
- [4] Franco, J. S.; Boyer, E.: Exact polyhedral visual hulls, BMVC, 1, 2003, 329-338. <http://dx.doi.org/10.5244/C.17.32>
- [5] Genç, S.; Atalay, V.: Texture extraction from photographs and rendering with dynamic texture mapping, Proceedings 10th International Conference on Image Analysis and Processing, 1999, 1055-1058. <http://dx.doi.org/10.1109/ICIAP.1999.797737>
- [6] Hormann, K.; Lévy, B.; Sheffer, A.: Mesh parameterization: theory and practice, ACM SIGGRAPH 2007 Courses on - SIGGRAPH '07, 2007, 1. <http://dx.doi.org/10.1145/1281500.1281510>
- [7] Kiefer, J.: Sequential minimax search for a maximum, Proc. Amer. Math. Soc., 4(3), 1953, 502-506. <http://dx.doi.org/10.2307/2032161>
- [8] Lévy, B.; Petitjean, S.; Ray, N.; Maillot, J.: Least squares conformal maps for automatic texture atlas generation, ACM Transactions on Graphics (TOG), 21(3), 2002, 362-371. <http://dx.doi.org/10.1145/566654.566590>
- [9] Low, K.-L.: Linear Least-Squares Optimization for Point-to-Plane ICP Surface Registration, Technical Report TR04-004, University of North Carolina at Chapel Hill, NC, 2004.
- [10] Liao, C.-Y.; Xiong, Y.-S.; Wang, D. W.; Lai, J.-Y.; Lee, J.-Y.: A camera calibration process for 3D digital model reconstruction of huge objects, 2016 International Conference on Machining, Materials and Mechanical Technologies (IC3MT 2016), Matsue, Japan, 2016.

- [11] Liao, C.-Y.; Ren, Y.-L.; Wang, D. W.; Lai, J.-Y., Lee, J.-Y.: Model for 3D printing created from multiview 2D silhouette images, Proceedings of International Conference on Leading Edge Manufacturing in 21st Century: LEM21, 2017. <http://dx.doi.org/10.1299/jsmelem.2017.9.130>
- [12] Marquardt, D. W.: An algorithm for least-squares estimation of nonlinear parameters, Journal of the Society for Industrial and Applied Mathematics, 11(2), 1963, 431-441. <http://dx.doi.org/10.1137/0111030>
- [13] Mulayim, A. Y.; Yilmaz, U.; Atalay, V.: Silhouette-based 3-D model reconstruction from multiple images, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 33(4), 2003, 582-591. <http://dx.doi.org/10.1109/TSMCB.2003.814303>
- [14] Niem, W.; Buschmann, R.: Automatic modelling of 3D natural objects from multiple views, Image Processing for Broadcast and Video Production, 1995, 181-193. http://dx.doi.org/10.1007/978-1-4471-3035-2_15
- [15] Ortery. <http://www.ortery.com/>
- [16] Phothong, W.; Wu, T.-C.; Lai, J.-Y.; Yu, C.-Y.; Wang, D. W.; Liao, C.-Y.: Quality improvement of 3D models reconstructed from silhouettes of multiple images, Computer-Aided Design and Applications, 15(3), 2018, 288-299. <http://dx.doi.org/10.1080/16864360.2017.1397881>
- [17] Sander, P. V.; Snyder, J.; Gortler, S. J.; Hoppe, H.: Texture mapping progressive meshes, Proceedings of the 28th annual conference on Computer graphics and interactive techniques – SIGGRAPH '01, 2001, 409-416. <http://dx.doi.org/10.1145/383259.383307>
- [18] Sander, P. V.; Wood, Z. J.; Gortler, S.; Snyder, J.; Hoppe, H.: Multi-chart geometry images, Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing, 2003, 146-155.
- [19] Sheffer, A.; Praun, E.; Rose, K.: Mesh parameterization methods and their applications, Foundations and Trends in Computer Graphics and Vision, 2(2), 2006, 105-171. <http://dx.doi.org/10.1561/0600000011>
- [20] Shamir, A.: A survey on mesh segmentation techniques, Computer Graphics Forum, 27(6), 2008, 1539-1556. <http://dx.doi.org/10.1111/j.1467-8659.2007.01103.x>
- [21] Wu, T.-C.; Lai, J.-Y.; Phothong, W.; Wang, D. W.; Liao, C.-Y.; Lee, J.-Y.: Editable texture map generation and optimization technique for 3D visualization presentation, Computer-Aided Design and Applications, 15(3), 2018, 378-389. <http://dx.doi.org/10.1080/16864360.2017.1397888>
- [22] Yous, S.; Laga, H.; Kidode, M.; Chihara, K.: GPU-based shape from silhouettes, Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia, ACM GRAPHITE '07, 2007, 71-77. <http://dx.doi.org/10.1145/1321261.1321274>
- [23] Zhang, Z.: A flexible new technique for camera calibration, IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11), 2000, 1330-1334. <http://dx.doi.org/10.1109/34.888718>
- [24] Zhang, Z.: Camera calibration with one-dimensional objects, IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(7), 2004, 892-899. <http://dx.doi.org/10.1109/TPAMI.2004.21>