COMPUTER-AIDED DESIGN & APPLICATIONS, 2018
VOL. 15, NO. 3, 288–299
https://doi.org/10.1080/16864360.2017.1397881

Quality improvement of 3D models reconstructed from silhouettes of multiple images

Watchama Phothong [a], Tsung-Chien Wu [a], Jiing-Yih Lai [a], Chun-Yeh Yu [a], Douglas W. Wang [b] and Chao-Yaug Liao [a]

[a]National Central University, Taoyuan, Taiwan; [b]Ortery Technologies, Inc., Taiwan

ABSTRACT

Shape-from-Silhouette (SFS) is a technique for estimating the shape of an object from images of its silhouette. It is a simple but effective technique for obtaining a three-dimensional (3D) model of an object by using two-dimensional (2D) images of the object in different views. The inherent problem with this method is visual features or irregularities, such as sharp edges, corners, and artifacts. These affect not only the outline shape of a model, but also the accuracy of texture mapping, which is essential for creating a 3D color model. The objective of this study is to propose a quality improvement method to address the aforementioned problems for the SFS method. The proposed quality improvement method is a multiple-step iteration procedure combining a re-meshing process and a mesh smoothing process to remove virtual features and artifacts on the original model, while preserving the image silhouettes and the smoothness of the model. Two algorithms based on regular and irregular meshes, respectively, are described and compared. Several examples are presented to verify the feasibility of the proposed algorithms.

KEYWORDS

Mesh generation; model reconstruction; quality improvement; visual hull; SFS

1. Introduction

Shape-from-Silhouette (SFS) [1–3] is a technique for estimating the shape of an object from images of its silhouette. It is a simple but effective technique for obtaining a three-dimensional (3D) model of an object by using two-dimensional (2D) images of the object in different views. One of its applications is product presentation in e-commerce, in which a 3D model combined with the object's texture, hereafter called a 3D color model, is presented to replace traditional 3D visualization. In 3D visualization, multiple 2D images showing different viewing angles are integrated and a 2D image at a given angle can be displayed via a viewing interface. However, the orientation process is not fluent owing to the limited angles recorded. A 3D color model can be displayed alone via a browser or combined with 3D visualization to alternately display the 3D model and 2D images. The generation of a suitable 3D model from images of multiple views is a prerequisite for achieving the aforementioned tasks. Two main issues are concerned regarding the construction of this type of 3D model. First, as this model is targeted at e-commerce applications, the feasibility of real-time operation on a website is the primary concern. That is, the data size should be limited to be suitable with web visualization. Second, as only silhouette data of multiple images are used for 3D modelling, the quality of this model for texture mapping should also be studied.

Approaches to generating 3D models from multiple images can be classified into two groups: shape-from-photoconsistency (SFP) and SFS. SFP approaches attempt to generate the complex shape of an object by utilizing both texture and silhouette information from multiple images, and include the space carving method [8], snakes technique [5], level-sets technique [7], graph cuts optimization [6], [16], and visual hull refinement [10]. The basic concept of these approaches is to generate photo-consistent models that minimize some measure of the discrepancy between the different image projections of their surface points. However, SFP methods are mainly suitable for artificial sculptures with handmade surface scragginess rather than machine-shaped products with smooth surfaces. In addition, factors such as homogeneousness and noise in color, inaccuracy in camera calibration, and non-Lambertian surfaces often cause instabilities in this technique when processing real images.

CONTACT Watchama Phothong p_watchama@hotmail.com; Tsung-Chien rabbit94577@gmail.com; Jiing-Yih Lai jylai@ncu.edu.tw; Chun-Yeh Yu qsc753mht@hotmail.com; Douglas W.Wang dwmwang@gmail.com; Chao-Yaug Liao cyliao@ncu.edu.tw

The SFS approach is conventional and has been studied extensively. It is based on a visual hull concept, in which the object geometry is reconstructed using the intersection of multiple sets of infinite polygons from silhouettes of 2D images of different views. The SFS approach is good to e-commerce applications because the entire process, including image taken, 3D modeling and texture mapping, can be integrated and carried out automatically. The data size on the 3D model can also be controlled by adjusting the number of silhouette points on each image. Although other 3D scanners and reconstruction methods are quite popular on the market, they are not suitable for e-commerce because of the following reasons: (1) the dense data points on the model restrict its use on web visualization, (2) some of the devices cannot provide the object's images, which are required for reconstructing the 3D color model of an object, and (3) the entire process is usually not fully automatic. However, the basic problem with the traditional SFS method in terms of the visual hull is that hidden concavities on the object surface cannot be resolved, and so the generated 3D model is not satisfactory. That is, the 3D model may have visual features or irregularities, such as sharp edges, corners, and artifacts, which do not actually exist on the object. Concavities on the object are usually formed as convex shapes on the model because they are invisible on image silhouettes. On the other hand, an object may have sharp features that should be maintained on the model. However, it is very difficult to distinguish virtual features from real object features on the model, and hence the removal of virtual features becomes difficult.

Mulayim et al. [13] investigated a 3D reconstruction method based on the SFS concept. Yemez et al. [17] proposed a method based on mesh deformation to obtain computationally efficient shapes using the silhouette method. Matusik et al. [12] computed an exact polyhedral representation of the visual hull directly from silhouettes. This method was well-suited to rendering with graphic hardware and could be computed very quickly because the computation was performed during the creation of the visual hull. Niem et al. [14] proposed a method for fast traversal of the layers of the projected cones and retrieval of the viewing edges that lie on the surface of the visual hull, achieving a real-time fully reconstructed model. However, the algorithm for obtaining 3D points from the visual hull is not accurate enough. In fact, none of the available methods in the literature address the aforementioned problems with SFS methods.

The objective of this study is to propose a quality improvement method to address the inherent problems of a 3D model generated by an SFS approach. As discussed, visual features and irregularities often occur on SFS models. The existence of sharp edges, corners, and artifacts affects not only the outline shape of a model, but also the accuracy of texture mapping, which is essential for creating a 3D color model. The proposed quality improvement method is a multiple-step iteration procedure combining a re-meshing process and a mesh smoothing process to remove virtual features and artifacts on the original model, while preserving the image silhouettes and the smoothness of the model. Two algorithms based on regular and irregular meshes, respectively, are described here and compared. Several examples are presented to verify the feasibility of the proposed algorithms.

2. Overall method

We will first describe a method for generating a 3D model based on the SFS approach because it is related to the approach used to improve the original model. To generate a 3D model from the silhouettes of multiple images, the exact polyhedral visual hulls (EPVH) method [12] is employed, in which an octree structure is established to subdivide the volume wherever needed, and a polyhedral intersection algorithm is developed to precisely compute all surface points. All surface points are exactly at the intersection of the polyhedral planes that form the visual hull. The topological relationships of the intersection points and their contributing polygons are recorded for mesh generation.

Figure 1 schematically illustrates an overall flowchart of the proposed 3D model construction method. A calibration mat and the object are separately placed on a turntable and a set of images is captured for each of them, as shown in Figs. 1(a) and 1(b), respectively. The number of images for both sets of images should be equal, typically either 16 or 32. A calibration algorithm [11] is employed to determine the camera parameters, e.g., viewing direction, camera position, aspect ratio, and focus. The camera parameters are used to obtain the relationship of a local coordinate on each object image and the 3D coordinates and to transform all object images onto the 3D coordinates. The silhouette of each object image is then obtained and recorded as points and line contours. Each silhouette is originally represented by a series of pixels that form single or multiple contours. Multiple contours appear when at least one inner loop exists on the image. As the density of the pixels on each contour is still too high, a chord length method is employed to reduce the number of points on each contour, which yields silhouette points [11]. Every line on a contour can form a polygon and a set of polygons is created with respect to one object image. All polygons from various object images are then used to evaluate the

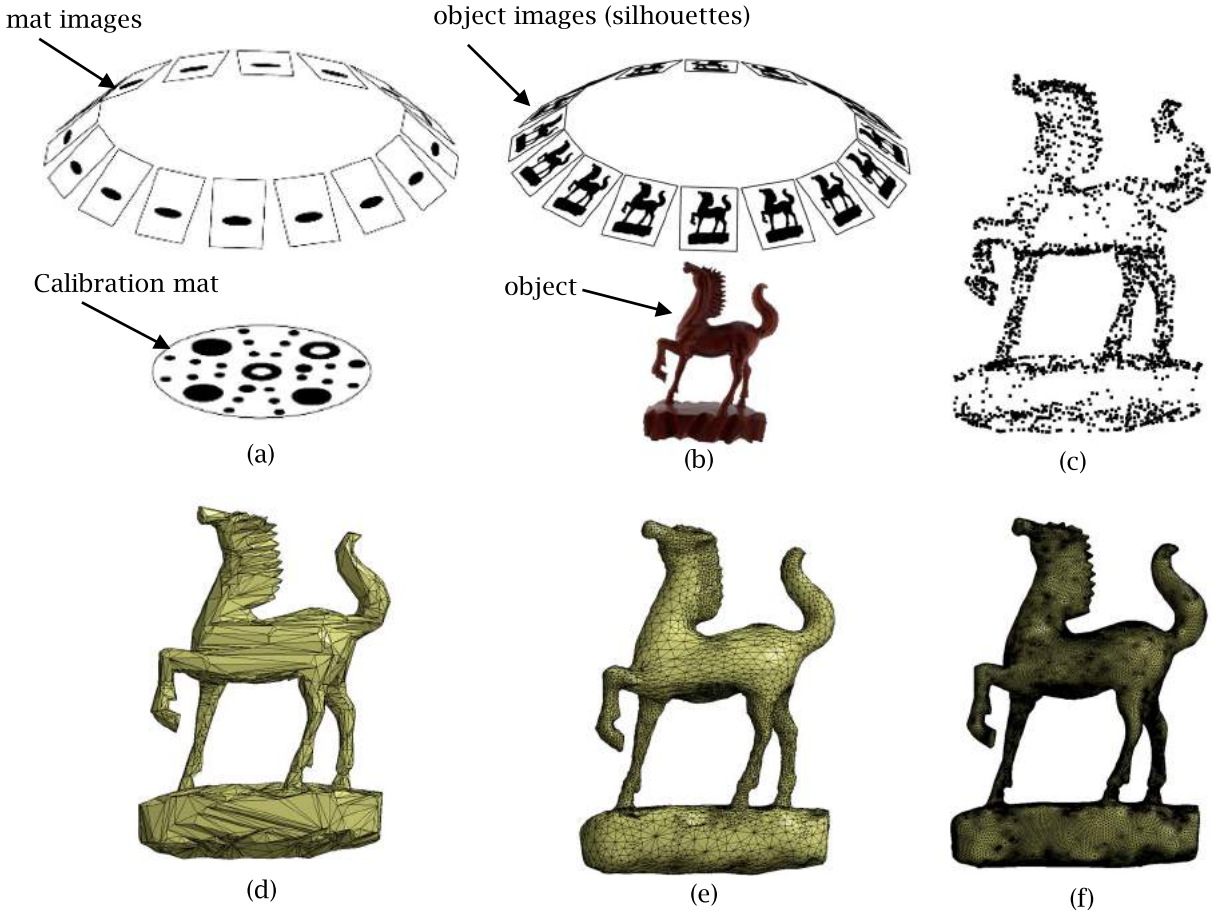

Figure 1. Overall flowchart of the proposed method for 3D model generation from multiple images: (a) calibration mat and mat images, (b) object and silhouette images, (c) 3D object surface points, (d) triangular meshes of the object, (e) the model after quality improvement with irregular mesh, and (f) the model after quality improvement with regular mesh.

intersection points (Fig. 1(c)), which are points on the object surface.

A triangulation method is then used to convert 3D points into triangular meshes (Fig. 1(d)). The mesh quality is very poor because the aforementioned problems with the SFS method all appear on this model. The proposed quality improvement method is then implemented. The main purpose is to eliminate sharp edges, corners, and artifacts, while preserving real object features on the model. It is a challenging issue because all virtual features and real object features appear simultaneously. Two quality improvement algorithms are employed, one based on irregular meshes and the other based on regular meshes. In the former, the original meshes are subdivided locally in terms of some criteria; in the latter, all meshes are subdivided uniformly. The techniques presented below mainly focus on the algorithm of irregular meshes, which can be modified to become the algorithm of regular meshes. The differences between both algorithms are explained. Figures 1(e) and 1(f) show the modified model after the quality improvement using the algorithms of irregular and regular meshes, respectively.

3. 3D mesh generation

Figure 2 depicts a flowchart of the calculation of 3D intersection points from silhouettes of multiple images. The input data includes the camera point, viewing direction, up direction, and silhouette data associated with each object image. Next, a polyhedron for an object image is generated using the silhouette points of the corresponding image. A polygon is created in terms of the perspective projection by combining the camera point and two neighboring silhouette points, as shown in Fig. 3(a). Thus, a point and line on the 2D image represent an edge and polygon, respectively, on the 3D domain, as shown in Fig. 3(b). All polygons from the entire silhouette on an object image form a polyhedron. A set of polyhedra can be obtained for all object images. Using these polyhedra, the first volume that encloses the target object is computed.

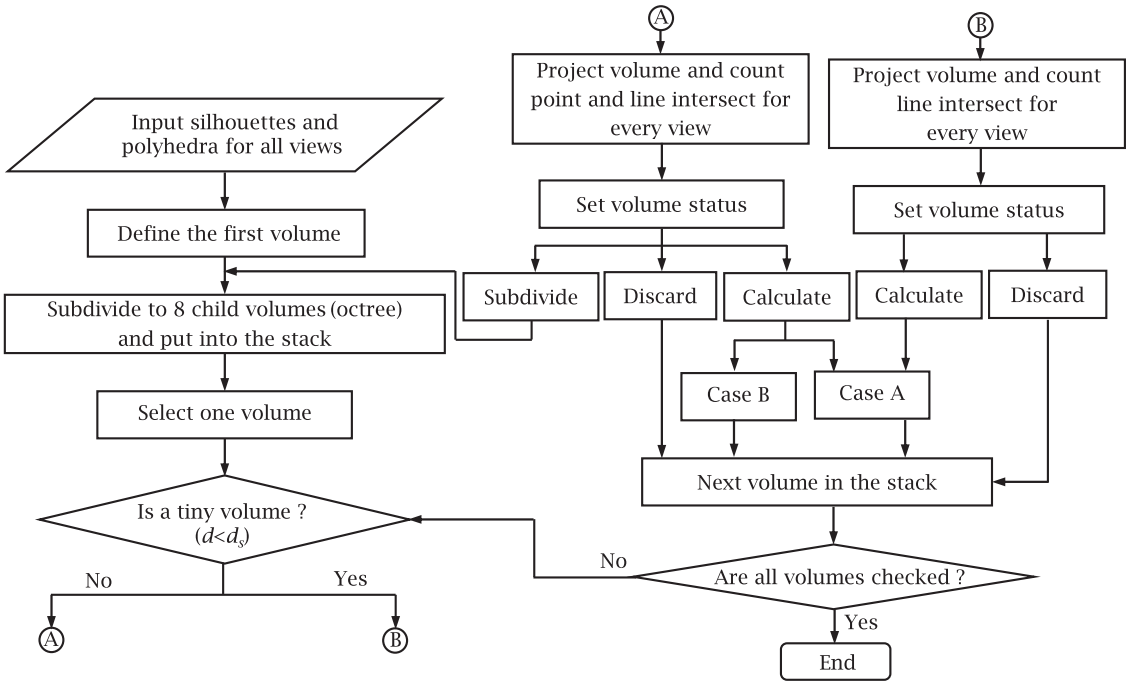

Figure 2. Flowchart for calculating 3D points from multiple sets of silhouette data.

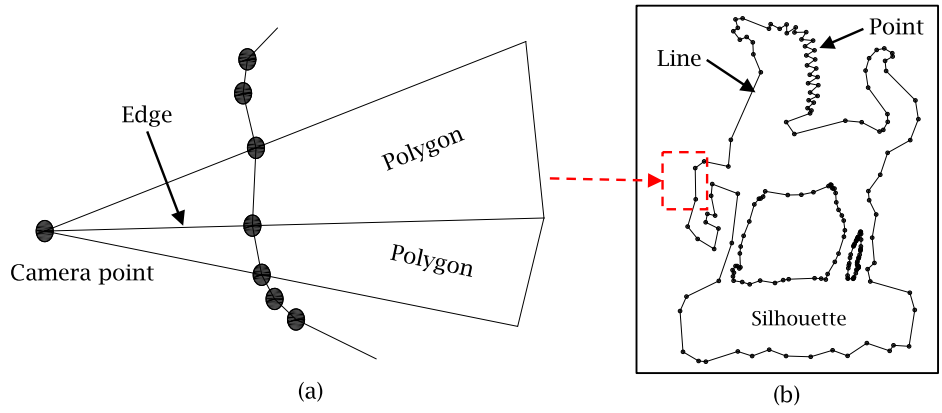

Figure 3. Terms used in this study: (a) a polygon is generated from a camera point and two neighboring silhouette points, and (b) a point and line on the 2D image represent an edge and polygon, respectively, on the 3D domain.

Subsequently, an octree structure is established to subdivide the volume and the number of polygons inside each sub-volume is checked. As Fig. 4 depicts, the volume is subdivided into eight sub-volumes and all sub-volumes are put on a stack. Each of the sub-volumes on the stack is checked in sequence. A sub-volume is projected onto all object images. If a line (Fig. 3(b)) on an image silhouette is inside or intersects the outer contour of the projected sub-volume, then a polygon is inside this sub-volume. This test is performed for all image silhouettes and the number of polygons inside this sub-volume is counted. If it is less than 3, this sub-volume is deleted; if it is larger than 3, this sub-volume is put on a stack and checked again later; if it is equal to 3, a 3D intersection point is computed using those three polygons. Each 3D intersection point is evaluated based on two conditions (Fig. 5): (1) case A: three polygons are from three different images and (2) case B: two of the connected polygons are from one image, and the third polygon is from the other image. The topological relationships of the polygons and 3D intersection points are recorded, including indices of the polygons contributing to each intersection point and indices of the points lying on each polygon. With this kind of topological data, 3D triangular meshes can be generated in a systematic manner. The flowchart of this process is shown in Fig. 2. A tiny volume is defined to terminate the subdivision. When the length d of a sub-volume is less than d_s, the length of the tiny volume,

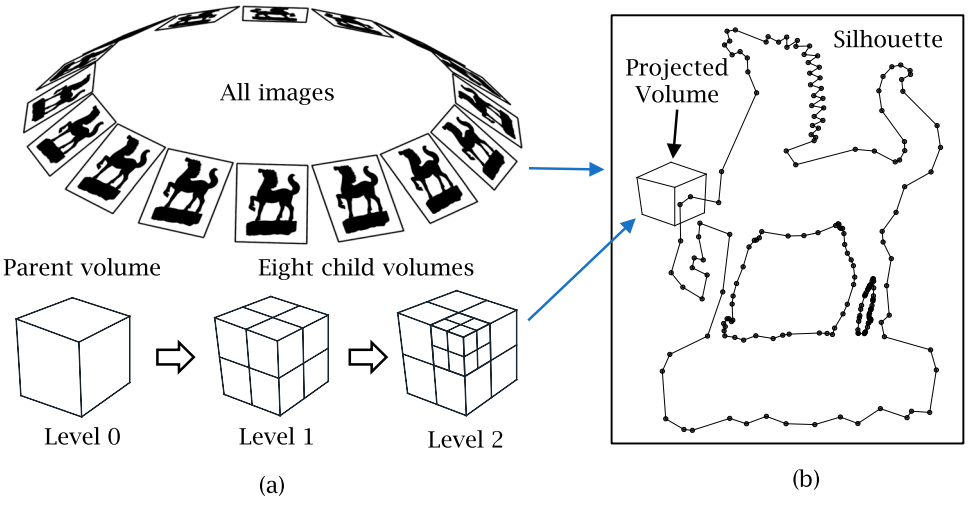

Figure 4. Volume subdivision and projection of each volume: (a) volume subdivision in accordance with an octree structure, and (b) the projection of each volume onto each image plane for checking the status of the volume.

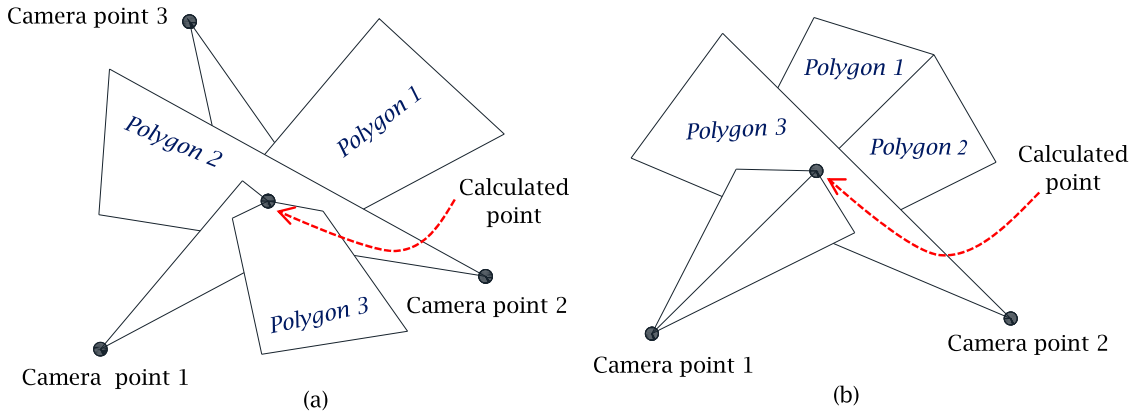

Figure 5. Calculated 3D point: (a) Case A, and (b) Case B.

the subdivision is terminated. If a sub-volume is larger than the tiny volume, it is projected and the point and line intersects are counted for every view. If not, the sub-volume is projected and only line intersect is counted for every view, as shown in Fig. 2. Here d_s is 0.2 mm. A detailed description of the generation of 3D points is provided in Phothong et al. [15].

All 3D points generated by the intersection of all polyhedra are essentially located on plane polygons (Fig. 6(a)). A set of 3D points lying on the boundary of a plane polygon can be triangulated to form planar triangular meshes. The combination of all planar triangular meshes from all polygons represents 3D meshes of the object. The 3D points on each plane polygon are irregularly distributed in the previous step, and should be arranged counterclockwise. Consider a plane polygon S_1 as in Fig. 6(a) with points $\boldsymbol{P}_1$ to $\boldsymbol{P}_7$ on its boundary. Starting from $\boldsymbol{P_1}$, all polygons that contribute to $\boldsymbol{P}_1$ are selected (S_2 and S_3). The points that share the same pair of polygons, namely S_1 and S_2, and S_1 and S_3, respectively, are then selected, which are $\boldsymbol{P}_7$ and $\boldsymbol{P}_2$ in Fig. 6(a). The point which can yield a counterclockwise direction is chosen as the next point, namely $\boldsymbol{P}_2$ in Fig. 6(a). This procedure is repeated for all points on S_1 until it finally yields a set of 3D points arranged in a counterclockwise direction. Once a polygon is complete, the procedure shifts to the next polygon. This process stops when all polygons have been processed.

Two algorithms are employed to generate triangular meshes from a sequence of 3D points on the boundary of a polygon. For the algorithm of irregular meshes, each set of 3D points on a surface polygon can be connected directly to form triangular meshes. As the points only lie on the boundary of a polygon, this can easily result in poor triangles, such as long and narrow triangles, as shown in Fig. 6(b). This triangulation algorithm is straightforward and is not explained here. For the algorithm of regular meshes, the original meshes should be as regular as possible. Therefore, a vertex insertion algorithm is employed for triangulation, as shown in

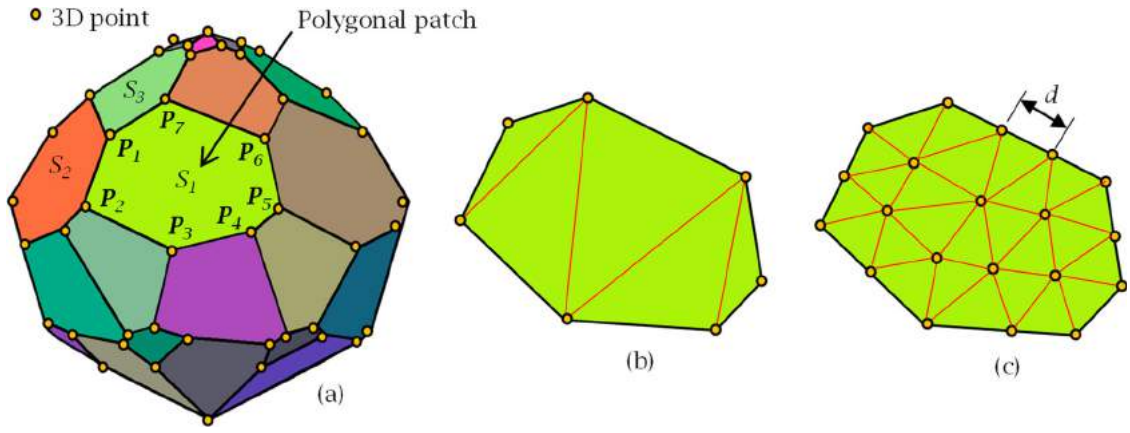

Figure 6. 3D points distributed on the boundary of each polygonal patch: (a) polygonal object, (b) triangulating a patch of irregular meshes, and (c) triangulating a patch of regular meshes.

Fig. 6(c). In this triangulation algorithm, all edges are subdivided into similar lengths and new vertices are inserted to create faces as close to regular triangles as possible. As Fig. 6(c) depicts, a maximum length d is specified to equally divide each boundary edge into segments, which can yield a set of boundary vertices with similar length between every two neighboring vertices. By using this set of boundary vertices, triangular meshes can be generated layer by layer from the boundary contour toward the inside [9]. Long and narrow triangles can be significantly reduced as both the mesh size and angles of each triangle can be controlled. Finally, all triangular meshes are integrated and saved as a triangular model, representing the original 3D model from the SFS method.

4. Quality improvement of 3D meshes

As discussed, the quality of the original 3D model must be further improved before it can be used for texture mapping. All 3D points are generated from silhouettes of 2D images. As all images are captured with a single-lens reflex camera in a controlled environment, the noise on the silhouettes can be neglected [11]. The primary error on the 3D model comes from the limitation of the visual hull method. As the original 3D model only represents the convex hull of an object, virtual features, such as sharp edges, corners, and artifacts exist all over the model. Artifacts usually appear as convex shapes near concavities of an object or regions which are invisible on images. The size of an artifact is often large enough to distort the outline shape of the object. On the other hand, virtual sharp edges and corners appear frequently on the original model because of the shape created by convex hulls. Conventional mesh smoothing or noise suppressing methods are invalid here because they cannot distinguish virtual features from real object features, and may remove both of them together.

To eliminate virtual features while preserving real object features on the original 3D model, a quality improvement method is proposed for upgrading each vertex iteratively by simultaneously enforcing a silhouette consistency force and a smoothing driving force. The silhouette consistency force is to maintain the silhouette consistency between the projected boundary of 3D meshes on each image plane and the corresponding image silhouette. The smoothing driving force is to smooth virtual features and artifacts. However, these two driving forces are conflicting in nature. When the silhouette consistency force is stronger, the silhouette consistency improves, but virtual features and artifacts may remain significant. In contrast, when the smoothing driving force is stronger, the model can become smoother, but the silhouette consistency may become worse. On the other hand, the mesh size has an important influence on both driving forces. The mesh size should be kept relatively small near real object features and relatively large near flat regions. In this way, the effect of the silhouette consistency force can be restricted to regions near the projected boundary, while the effect of the smoothing driving force can be enhanced near flat surfaces. A re-meshing process can be implemented to adjust the mesh size locally or globally. The main disadvantage of the quality improving method is that the number of meshes on a model can thus be increased tremendously. Therefore, a mesh reduction process should be implemented after mesh optimization.

Figure 7 depicts a flowchart of the quality improvement algorithm using irregular meshes. It starts by computing the maximum and minimum edge lengths d_{max} and d_{min} respectively, of each position of the object from the original model to use in the mesh subdivision, edge collapse, and edge subdivision in the re-meshing process. Mesh subdivision is employed first to reduce all mesh sizes. Edge collapse is then employed to eliminate shorter

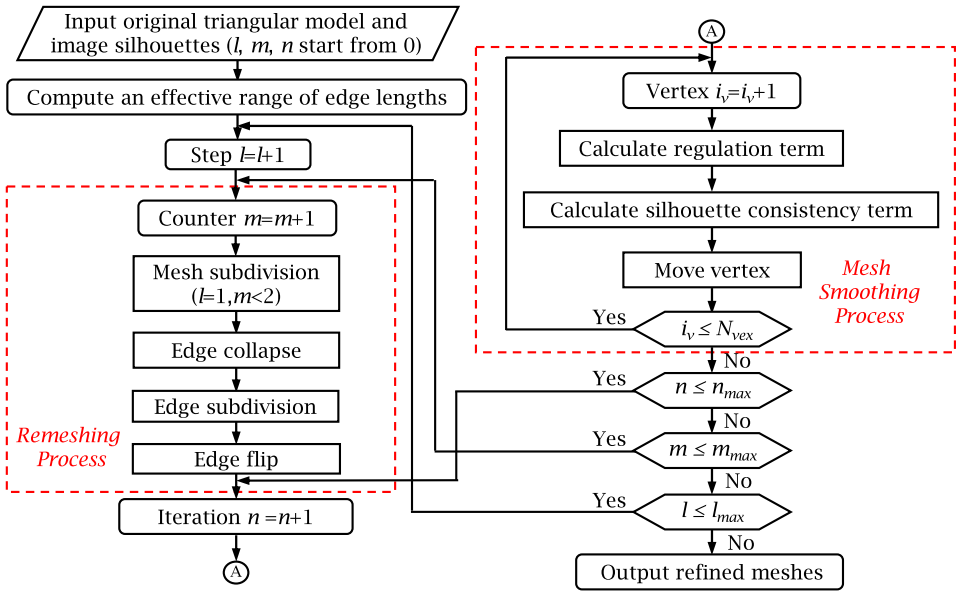

Figure 7. Overall flowchart of the proposed quality improvement method using irregular meshes.

edges. The parameter d_{min} is used as a threshold for edge collapse. Similarly, edge subdivision is employed to subdivide meshes with longer lengths. The parameter d_{max} is used as a threshold for edge subdivision. The symbol l denotes the iteration step. In each iteration step, both the re-meshing process and the mesh smoothing process are implemented in sequence. The remeshing process, including mesh subdivision, edge collapse, edge subdivision, and edge flip, is implemented iteratively m_{max} times. Mesh subdivision is performed when $l = 1$ and $m < 2$. Edge collapse and edge subdivision are combined to keep the mesh size relatively large near regions with a low density of vertices on the original model, while keeping the mesh size relatively small near regions with a high density of vertices on the original model. The mesh smoothing process, including a silhouette consistency term and a regulation term, is implemented iteratively n_{max} times. Each vertex on the 3D mesh is upgraded iteratively in accordance with the silhouette consistency term and the regulation term. This process stops when all vertices have been processed. It could be possible to define a quality criterion, such as an error indicating the deviation of the points between two continuous iterations, for terminating the iterative process automatically. However, as there are three loops of iteration in the proposed algorithm, it may require excessive CPU time to evaluate a quality index in each iterative loop. More importantly, it is difficult to determine same quality indices suitable for both simple and complex objects. In this study, we investigate the effect of l_{max}, n_{max} and m_{max} and suggest a better set of parameters for different examples.

The maximum edge length d_{max} and the minimum edge length d_{min} are determined in terms of the size and the density of vertices on the original model. The diagonal length D of the bounding box on the original model is computed first. Two lengths l_{max} and l_{min} are determined to represent the maximum and minimum edge lengths allowed, respectively, on the meshes. Here, l_{max} and l_{min} are chosen as follows: $l_{max} = 6\% \times D$ and $l_{min} = 0.1\% \times D \times F$, where F denotes the ratio between d_{max} and d_{min}. The bounding box of the original model is divided into cell boxes, as shown in Fig. 8, where the cell length in each axis is chosen as 2% of the length in that axis. The

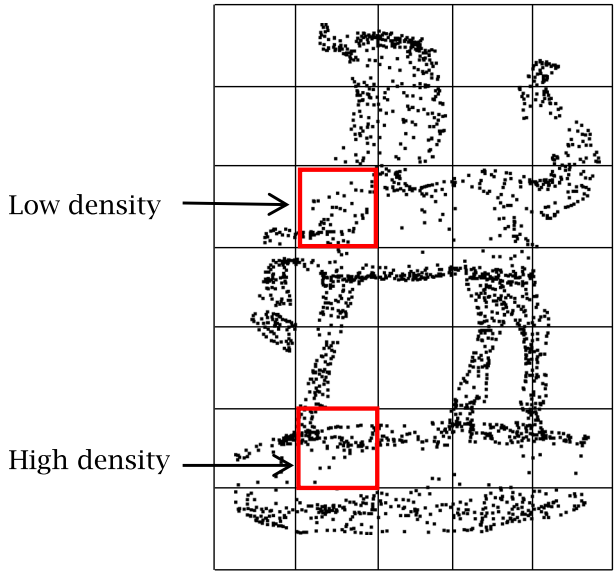

Figure 8. Density of vertices on each cell box.

number of vertices on each cell box is counted. A broken-line chart can be plotted with the number of vertices in a box as the horizontal axis and the number of boxes counted in that number of vertices as the vertical axis. The broken-line chart is divided into $N_{max}+1$ regions, and d_{max} and d_{min} on each region are computed as follows. The maximum edge length d_{max} in the i^{th} region is determined as follows:

$$d_{max} = l_{max} - \frac{(l_{max} - l_{min}) \cdot N_i}{N_{max}} \tag{1}$$

where N_i denotes the index of each region. For example, if N_{max} is 4, then N_i should be 0, 1, 2, 3, and 4. It is noted that N_{i} starts from the region with a low density of vertices on the boxes and ends at the region with a high density of vertices on the boxes. A set of d_{max} and d_{min} must be evaluated at each i. Here, d_{max} is evaluated using Eq. (1) and d_{min} is evaluated using $d_{min} = d_{max}/F$, where F is typically set to $5 \sim 10$. In Eq. (1), if N_i is near N_{max}, then d_{max} is close to l_{min}, which indicates that d_{max} should be kept smaller for the region with a higher density of vertices. In contrast, if N_i is near 0, then d_{max} is close to l_{max}, which indicates that d_{max} should be kept larger for the region with a lower density of vertices. With this approach, we can adaptively determine the maximum d_{max} and minimum d_{min} edge lengths in terms of the size and density of vertices on the original model.

The re-meshing process includes four operators as described below:

(1) Mesh subdivision (Fig. 9(a)): this operator subdivides all faces of the model. It is performed first to reduce all face sizes, where each face is subdivided into four sub-faces. This process is stopped when all faces are subdivided.

(2) Edge collapse (Fig. 9(b)): this operator collapses an edge into a vertex. If the length of an edge is shorter than d_{min}, this edge is diminished. Two faces neighboring this edge can thus be deleted. The area removed can be distributed to the faces that connect to the new vertex. In this procedure, all edges are checked one by one. An edge collapse is implemented whenever an edge length is shorter than d_{min}.

(3) Edge subdivision (Fig. 9(c)): this operator subdivides faces locally. If any edge of a face is longer than d_{max}, the corresponding edge is subdivided into two edges by a vertex. Two faces neighboring this edge can thus be subdivided into two faces. This operation can be implemented locally because it does not give rise to any topological problems. In this procedure, all edges are checked one by one. An edge subdivision is implemented whenever an edge length is longer than d_{max}.

(4) Edge flip (Fig. 9(d)): this operator adjusts the regularity of the faces. The minimum angle of two triangles neighboring the same edge is computed. When this angle before edge flip is smaller than that after edge flip, edge flip is implemented. Narrow triangles can thus be reduced after edge flip is implemented.

In mesh smoothing, each vertex in 3D space is individually checked to calculate its new position in terms of a silhouette consistency term and a regulation term. Let $\boldsymbol{v}_i$ be the original position of a vertex in 3D space and $\boldsymbol{v}'_i$ be its new position. The new position $\boldsymbol{v}'_i$ can be computed as follows:

$$\boldsymbol{v}'_i = \boldsymbol{v}_i + \alpha \boldsymbol{F}_{sil} + \beta \boldsymbol{F}_{int} \tag{2}$$

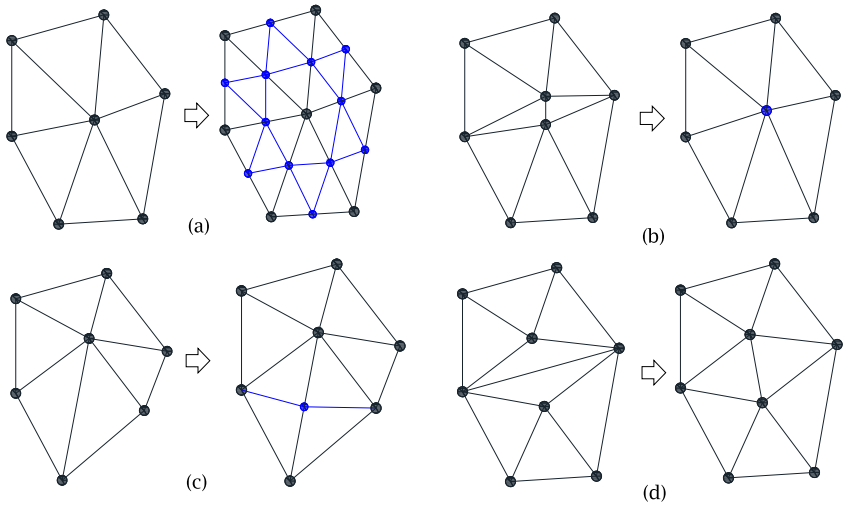

Figure 9. Four re-meshing operations: (a) mesh subdivision, (b) edge collapse, (c) edge subdivision, and (d) edge flip.

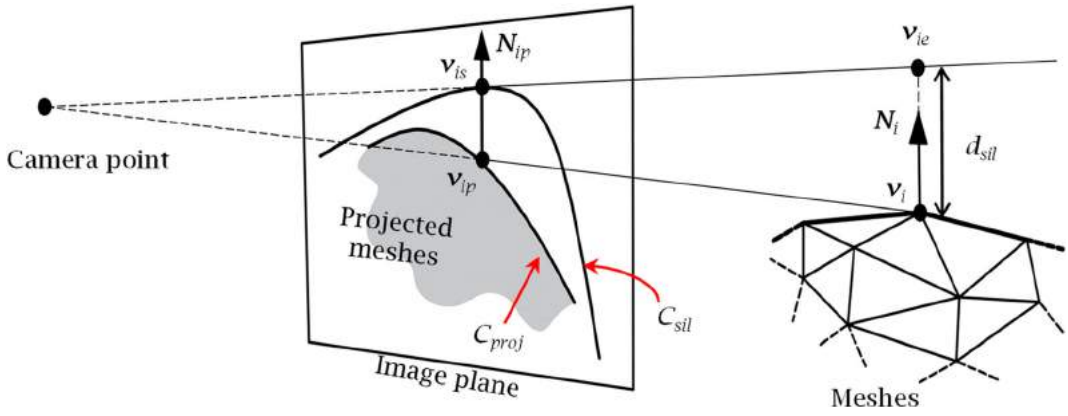

Figure 10. The silhouette consistency term for the mesh smoothing process.

where $\boldsymbol{F}_{sil}$ denotes the silhouette consistency term, $\boldsymbol{F}_{int}$ denotes the regulation term, and α and β are weights. Here, $\boldsymbol{F}_{sil}$ essentially represents a displacement of the vertex $\boldsymbol{v}_i$ so that its projected vertex $\boldsymbol{v}_{ip}$ on a 2D image can approach its image silhouette. Thus, $\boldsymbol{F}_{sil}$ can be represented as

$$\boldsymbol{F}_{sil} = d_{sil}\boldsymbol{N}_i \tag{3}$$

Where $\boldsymbol{N}_i$ is the surface normal at $\boldsymbol{v}_i$ and d_{sil} is the amplitude of the silhouette consistency term. Here, $\boldsymbol{N}_i$ is evaluated using the average surface normal of the faces neighboring $\boldsymbol{v}_i$. For a projected vertex $\boldsymbol{v}_{ip}$ located on the projected boundary, the amplitude d_{sil} of $\boldsymbol{F}_{sil}$ is evaluated in accordance with Fig. 10, where C_{proj} indicates the projected boundary, C_{sil} indicates the image silhouette, and $\boldsymbol{N}_{ip}$ indicates the projected directional vector of $\boldsymbol{N}_i$. The line $L(\boldsymbol{v}_i, \boldsymbol{N}_{ip})$ intersects the image silhouette C_{sil} at a point $\boldsymbol{v}_{is}$. Then, project $\boldsymbol{v}_{is}$ back to the 3D space to intersect the line $L(\boldsymbol{v}_i, \boldsymbol{N}_i)$ at a point $\boldsymbol{v}_{ie}$. The distance between $\boldsymbol{v}_i$ and $\boldsymbol{v}_{ie}$ represents the amplitude d_{sil}. Note that the projection of a 3D vertex onto different images can yield different projected vertices. If a projected vertex is located on the projected boundary, d_{sil} is evaluated. Multiple d_{sil} can be obtained if several projected vertices of a 3D vertex are located on the projected boundaries. Their average is used as the amplitude d_{sil} for $\boldsymbol{F}_{sil}$. A projected vertex that is not located on the projected boundary does not need to move to the silhouette boundary. However, to avoid a sudden jump in displacement at the transition of these two kinds of vertices, a displacement is assigned on each of the non-boundary vertices. This can be done using a progressive process, in which the amplitude d_{sil} of each non-boundary vertex is evaluated using the average d_{sil} of all its neighboring vertices.

The regulation term $\boldsymbol{F}_{int}$ essentially computes the difference vector from the vertex $\boldsymbol{v}_i$ to the center of its neighbors, and can be represented as

$$\boldsymbol{F}_{int} = \frac{\sum_{v_j \in M_i} \boldsymbol{v}_j}{|M_i|} - \boldsymbol{v}_i, \tag{4}$$

where M_i denotes a set of neighboring vertices of the vertex $\boldsymbol{v}_i$. This expression has a strong vertical force to make vertices coplanar and a strong lateral force to make vertices uniformly distributed. In each iteration step (l) in Figs. 2 and 3, the re-meshing process is performed m_{max} times. In each re-meshing process, the mesh smoothing process is performed n_{max} times. The weights α and β in Eq. (2) are set to 0.1 and 1, respectively, in the current algorithm. However, an improved algorithm with variable α and β could be studied as the mesh size and dense of the data could be varied during the iteration.

5. Examples and discussion

Several examples were employed to evaluate the performance of quality improvement algorithms with irregular meshes. The object was placed on a turntable and sixteen images distributed uniformly around the object were captured by a camera. The silhouette of each image was extracted and expressed as silhouette points, in which the distribution of the points is irregular in accordance with the shape of the contour. The inputs were silhouette points for all sixteen images, and camera parameters associated with all images captured. The output was the surface triangular model of the object. A comparison of the results using regular and irregular meshes for different examples is presented.

Figure 11 shows the original images and the original 3D meshes constructed using the proposed mesh generation method for three examples, where the left and right plots in each figure panel denote the original image and the 3D meshes, respectively. Since the proposed 3D modeling method is based on visual computation, we will

Figure 11. Original images and triangular meshes reconstructed: (a) shoe and (b) horse statue.

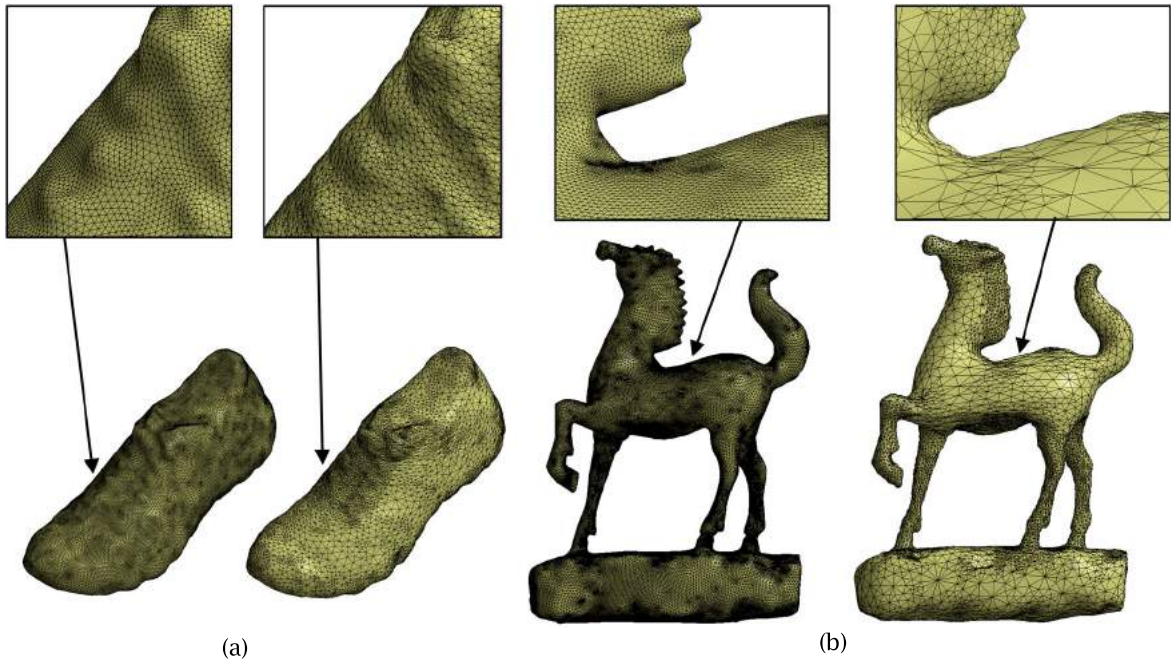

Figure 12. Quality improvement of meshes for two examples: (a) shoe and (b) horse statue.

focus on the accuracy of the outline appearance. Note that artifacts appear as convex shapes on all three models, especially near the bottom and the concave regions of each model. Some of the artifacts are even large enough to distort the outline shape, such as for the shoe (Fig. 11(a)). In addition, all three models are obviously not smooth because sharp edges and corners appear all over the models. Note that the number of faces in all models was less than 10,000, which will not induce any sluggishness during data download and website operation.

The results of the quality improvement for the aforementioned three models are shown in Fig. 12, where the left and right plots in each figure panel depict regular meshes and irregular meshes, respectively, and the parameters l_{max}, m_{max}, and n_{max} are respectively set to 1, 5, and 10. The results show that the outline shapes of all three objects are preserved quite well with all virtual features and artifacts completely removed. Table 1 lists the number of vertices and faces for the original 3D meshes, the modified meshes using irregular meshes, and a comparison with those using regular meshes. The number of vertices and faces increased tremendously for regular meshes but had a lower increase for irregular meshes. Note that the number of vertices and faces increased to 106,403 and 212,810, respectively, for the case "Horse" with the algorithm of regular meshes, which is because this case has many sharp feature edges that should be maintained. However, for the same case, the number of vertices and faces required were only 25,869 and 51,742, respectively, with the algorithm of irregular meshes. The data size of the model after quality improvement may exceed the constraint required for web visualization. A mesh simplification algorithm proposed by Garland [4] could be employed to deal with this issue, which can reduce the number of meshes, while preserving important topological and geometric characteristics

Table 1. Triangular meshes obtained and CPU time required for three examples.

	Initial 3D model		Irregular meshes improvement			Regular meshes improvement		
Case	Vertices	Faces	Vertices	Faces	CPU time (sec)	Vertices	Faces	CPU time (sec)
Shoe	5,129	10,254	14,178	28,352	295	31,953	63,902	478
Cat doll	4,264	8,524	12,153	24,302	249	40,520	81,036	346
Horse	6,793	13,592	25,869	51,742	411	106,403	212,810	2,180

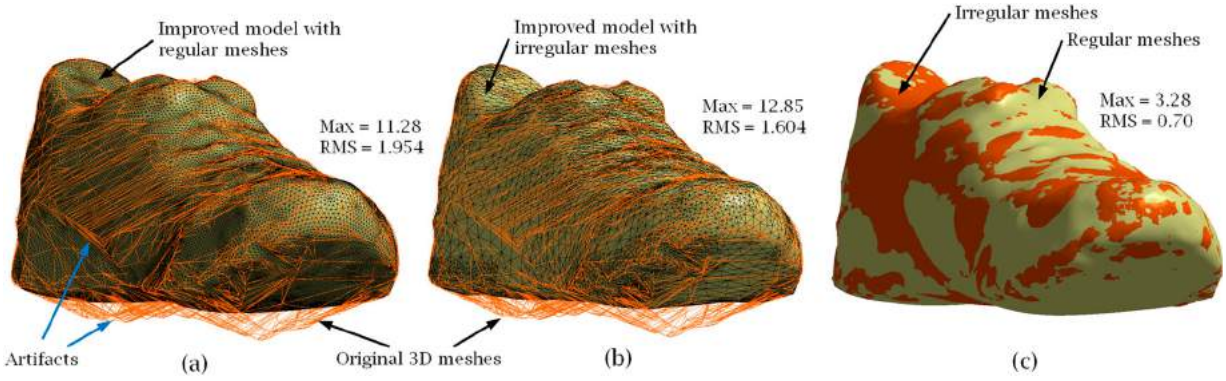

Figure 13. Comparison of shoe results: (a) original 3D meshes vs. improved model with regular meshes, (b) original 3D meshes vs. improved model with irregular meshes, and (c) improved model with regular meshes vs. that with irregular meshes, unit in mm.

of the original model. Nevertheless, the algorithm of irregular meshes is still better than that of regular meshes as it employs less number of meshes to achieve similar smoothing result. The CPU time required for both quality improvement algorithms is also indicated, with shorter times for irregular meshes than regular meshes. The primary reason is that mesh subdivision is implemented on all meshes in the algorithm of regular meshes, whereas it is implemented adaptively in the algorithm of irregular meshes.

Figure 13 depicts the comparison of the original 3D meshes, the modified meshes using regular meshes, and the one using irregular meshes for the case "Shoe", where the size of the show is 132x300x130 mm. Artifacts are notable on the original 3D meshes, especially near the bottom and one side waist of the shoe, as shown by the wireframe meshes in Figs. 13(a) and 13(b), because these regions are invisible on images. These artifacts have completely been removed by both methods, as shown in Fig. 13(a) for the improved model with regular meshes, and Fig. 13(b) for the improved model with irregular meshes. The maximum (MAX) errors between the modified model and the original 3D meshes are 11.28 and 12.85 mm, respectively, for regular meshes and irregular meshes. This result indicates that both methods can effectively reduce the effect of artifacts. Note that the root-mean-square (RMS) error between the original 3D meshes and the modified model was not provided in Figs. 13(a) and 13(b) because this value has no meaning in this case (the original 3D meshes have many virtual features that do not actually exist on the object). Figure 13(c) compares the improved 3D models using regular and irregular meshes. The result shows that both models are quite similar, with MAX and RMS errors being 3.28 and 0.70 mm, respectively. However, as several parameters in the smoothing process can be adjusted, it could be possible to further study the smoothing process to improve the accuracy of the algorithm of irregular meshes. If so, the algorithm of irregular meshes would be more feasible than the algorithm of regular meshes for real applications.

6. Conclusion

A 3D model based on the SFS approach inherently possesses virtual features and artifacts. These not only affect the smoothness of the model, but also distort the accuracy of the outline shape. Since both virtual features and real object features appear simultaneously, conventional mesh smoothing techniques cannot handle this problem. The proposed quality improvement method in terms of irregular meshes employs a re-meshing process and a smoothing process to eliminate all virtual features and artifacts while preserving the smoothness of the model. It can maintain an irregular distribution of mesh sizes with a low density of meshes near flat regions and a high density of meshes near real object features. The results show that all artifacts, sharp edges, and corners are completely eliminated for all three cases, which is the most important contribution of the proposed method. In addition, the method of irregular meshes can yield fewer vertices and faces on the model than that of regular meshes. The

CPU time required for the method of irregular meshes is also much lower than that of regular meshes. However, as the overall iteration procedure is very complex and several parameters are used in the algorithm, further study should be performed to improve the robustness of the proposed quality improvement method.

ORCID

Watchama Phothong http://orcid.org/0000-0002-3239-4564
Tsung-Chien Wu http://orcid.org/0000-0002-2299-7361
Jiing-Yih Lai http://orcid.org/0000-0002-0495-0826
Chun-Yeh Yu http://orcid.org/0000-0002-9807-7086
Douglas W. Wang http://orcid.org/0000-0002-8039-5027
Chao-Yaug Liao http://orcid.org/0000-0001-8203-9520

References

[1] Bellmann, A.; Hellwich, O.; Rodehorst, V.; Yilmaz, U.: A benchmarking dataset for performance evaluation of automatic surface reconstruction algorithms, 2007 IEEE Conference on Computer Vision and Pattern Recognition, 2007. https://doi.org/10.1109/CVPR.2007.383349

[2] Brian, S. M.; Curlee, B.; Diebel, J.; Scharstein, D.; Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms, 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006. https://doi.org/10.1109/CVPR.2006.19

[3] Chen, Q.; Medioni, G.: A volumetric stereo matching method application to image-based modeling, Computer vision and pattern recognition, 1009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999. https://doi.org/10.1109/CVPR.1999.786913

[4] Garland, M.; Heckbert, Paul, S.: Surface simplification using quadric error metrics, SIGGRAPH'97 Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, 1997, 209–216. https://doi.org/10.1145/258734.258849

[5] Hernández, C.; Schmitt, F.: Silhouette and stereo fusion for 3D object modeli*ng, Computer Vision and Image Understanding, 96(3), 2004, 367–392. https://doi.org/10.1016/j.cviu.2004.03.016

[6] Hornung, A.; Kobbelt, L.: Hierarchical volumetric multi-view stereo reconstruction of manifold surfaces based on dual graph embedding, 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006. https://doi.org/10.1109/CVPR.2006.135

[7] Keriven, R.: A variational framework to shape from contours, Technical Report CERMICS-2002-221b. ENPC, 2002.

[8] Kutulakos, V.; Seitz, S.: A theory of shape by space carving, International Journal of Computer Vision, 38(3), 2000, 199–218. https://doi.org/10.1023/A:1008191222954

[9] Lai, J. Y.; Hsu, S. H.: On the development of a hole filling algorithm for triangular meshes, Journal of the Chinese Institute of Engineers, 30(5), 2007, 877–889. https://doi.org/10.1080/02533839.2007.9671314

[10] Lazebnik, S.; Furukawa, S.; Ponce, J.: Projective visual hulls, International Journal of Computer Vision, 74(2), 2007, 1371–65. https://doi.org/10.1007/s11263-006-0008-x

[11] 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, Machining, Materials and Mechanical Technologies, Matsue Terrsa, Matsue, Japan: 7-11 October 2016.

[12] Matusik, W.; Buehler, C.; Mcmillan, L.: Polyhedral visual hulls for real-time rendering, Rendering Techniques 2001, Proceedings of the Eurographics Workshop in London, 2001, 115–125. https://doi.org/10.1007/978-3-7091-6242-2_11

[13] Mulayim, A. Y.; Yilmaz, U.; Atalay, V.: Silhouette-based 3D model reconstruction from multiple images, IEEE Transactions on systems, man and cybernetics, part B, 33(4), 2003, 582–591. https://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, Workshops in Computing, 181–193. https://doi.org/10.1007/978-1-4471-3035-2_15

[15] Phothong, W.; Wu, T.-C.; Lai, J.-Y.; Wang, D. W.; Liao, C.-Y.; Lee, J.-Y.: Fast and accurate triangular model generation for the shape-from-silhouette technique, Computer-Aided Design and Applications, 14(4), 2016, 436–449. https://doi.org/10.1080/16864360.2016.1257186

[16] Sinha, S.; Pollefeys, M.: Multi-view reconstruction using photo-consistency and exact silhouette constraints: a maximum flow formulation, In Proceedings of the Tenth IEEE International Conference on Computer Vision, Washington, DC, USA, 2005, 349–356. https://doi.org/10.1109/ICCV.2005.159

[17] Yemez, Y.; Sahilioglu, Y.: Shape from silhouette using topology-adaptive mesh deformation, Pattern Recognition Letters, 30(13), 2009, 1198–1207. https://doi.org/10.1016/j.patrec.2009.05.012