Mending Fractured Models Using Surface Patches

Pavlos Stavrou[1] , Georgios Papaioannou[2] and Theoharis Theoharis[1]

1 National Kapodistrian University of Athens , Department of Informatics ,
{ p.stavrou , theotheo }@di.uoa.gr
http://graphics.di.uoa.gr
2 Athens University of Economics and Business, Department of Informatics
gepap@aueb.gr
http://www.aueb.gr/users/gepap/

Abstract

Repairing and preserving cultural artefacts is a painstaking and time consuming process requiring vast amounts of effort on behalf of area specialists. Digitizing such artefacts not only enhances their endurance in time but also yields for the opportunity to automatically repair them using three dimensional algorithms. The initial step of our approach attempts to join models along their matching fractured surfaces, with respect to surface continuity and detail preservation, by creating a surface patch between them. In particular, our method extracts and smoothes the outlines of fractured surfaces and determines sets of best matching vertices among them, which, subsequently, are used as a guideline to create a surface patch that joins the two parts. We demonstrate our method using real digitized broken artefacts as well as manually created broken models.

1 Introduction

Numerous cultural artefacts, stemming from antiquity, are continuously being discovered while even more lay in museum storage facilities, unexposed to the public, before they are restored. Artefact restoration today is a manual task requiring much detail and attention, as it is performed on the actual valuable artefacts, making it a lengthy process, while corrosion over time and large numbers of fractures to be matched, in order to create a whole object, burden the workload.

Taking advantage of high quality range scanning devices, we are able to acquire polygonal representations of such artefacts, thus having the opportunity, through a wide variety of computer algorithms, to proceed in their semi or fully automatic restoration. In addition, the heavy workload is transferred to the computer, allowing for the human factor to have an intervening role in the process to enhance results.

One of the most important algorithms that can be employed to aid in matching and reassembling fractured artefacts was proposed by Papaioannou et al. [1]. The authors use surface curvature properties to identify fractured surfaces and then match and align different parts along their fractured surfaces. Recently Huang Q. et al. [2] proposed a method for reassembling 3D solids introducing graph-cuts segmentation for fracture identification and the novel integral invariants for computing multi-scale surface characteristics applied to pair-wise matching.

Another genre of algorithms that can be used to repair digitized artefacts is that of hole filling. The most popular hole-filling techniques make use of implicit surfaces [3] to represent the objects to be repaired, while more advanced techniques rely on volumetric diffusion [4] and surface in-painting to fill holes.

Our contribution to the repair process is based on connecting matching 3D objects along their fractured surfaces by creating a geometry consistent surface patch between them in order to produce a manifold, unified, three dimensional representation of the repaired artefact.

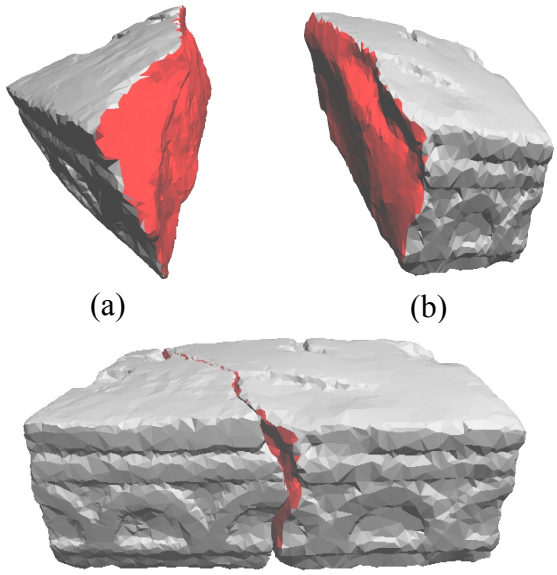


Fig. 1 Fractured cornerstone object. Fractured surfaces are marked in red. (a) The left part. (b) The right part. (c) The two parts aligned after matching.

The proposed algorithm accepts the aligned, matching parts and indexes to their connecting surfaces as an input. It is important for each part to be manifold in order for the algorithm to produce coherent results. To correctly index the surfaces a segmentation algorithm must first be applied to the objects. We chose to use the one proposed by Papaioannou et al. [5] due to its simplicity and speed. As for the proper alignment, we again use the method proposed by Papaioannou et al. [1].

Given two objects, Obj_1 and Obj_2 (Fig.1), matched along fractured surfaces $S_{1,m}$ and $S_{2,n}$, the first step to constructing our surface patch is to extract the outlines of the fractured surfaces by creating a chain of adjacent vertices that belong to edges common between faces of the fractured surface and its adjacent surfaces. Our goal is to connect vertices of matching outlines following a direction that will preserve the objects' continuity. Due to dense sampling along fractures and the increasing number of vertices required to represent them, we choose to work with a smoother, lower resolution outline. To achieve that, we acquire an image of each fractured surface using the z-buffer and construct a pixel outline. The 2D outline is transported to 3D space by an inverse projection procedure and is smoothed using a gaussian filter. Vertices of original outlines are mapped to vertices of the newly constructed outlines using a nearest neighbour metric.

We connect vertices between smoothed outlines with respect to the direction dictated by the surface's average normals and the vector defined by the vertices on the centroids of each outline. Finally, we connect vertices of the original outlines using the connected sampled vertices to which they are mapped as a guide and create the surface patch by inserting edges for each connection and triangulating the outcome.

2 Surface Outline Creation

The construction of the original surfaces' outlines is a fairly simple process since we have segmented each object properly. A boundary face on a surface shares at least one common edge with another face from a different surface. Starting at an arbitrary boundary face of the surface we walk towards its adjacent boundary faces, passing only once through each face, and adding vertices that belong to corresponding common edges to a vertex chain, until we complete our cycle returning to the starting face. The vertex chain $O_{k,m} = \{v_{k,m}(1),..,v_{k,m}(M_m)\}$, M_m being the number of vertices in $O_{k,m}$, represents our surface outline.

In order to construct the 2D sampled outline we first acquire an image of the surface using the z-buffer. Using this technique, we are able to compare objects of different types in a unified manner. For an object Obj_k, each surface $S_{k,m}$ has an average normal vector $\boldsymbol{n}_{av}(S_{k,m})$ dependent on the normal vectors of its faces(f) :

$$n_{av}(S_{k,m}) = \left\| \sum_{i|f_f \in S_{s,m}} A_i n_i \right\|^{-1} \cdot \sum_{i|f_f \in S_{s,m}} A_i n_i \quad (1)$$

where A_i is the area of face f_i and $\boldsymbol{n}_i$ its normal vector. The sampled outline $H_{k,m} = \{h_{k,m}(1),\ldots,h_{k,m}(N_m)\}$, N_m being the number of points in $H_{k,m}$, is derived in the following manner :

We compute the average vector $\boldsymbol{n}_{av}(S_{i,m},S_{j,n})$ of the matching surfaces average normal vectors. Each surface is then transformed in a way that, depending on the angle between $\boldsymbol{n}_{av}(S_{i,m},S_{j,n})$

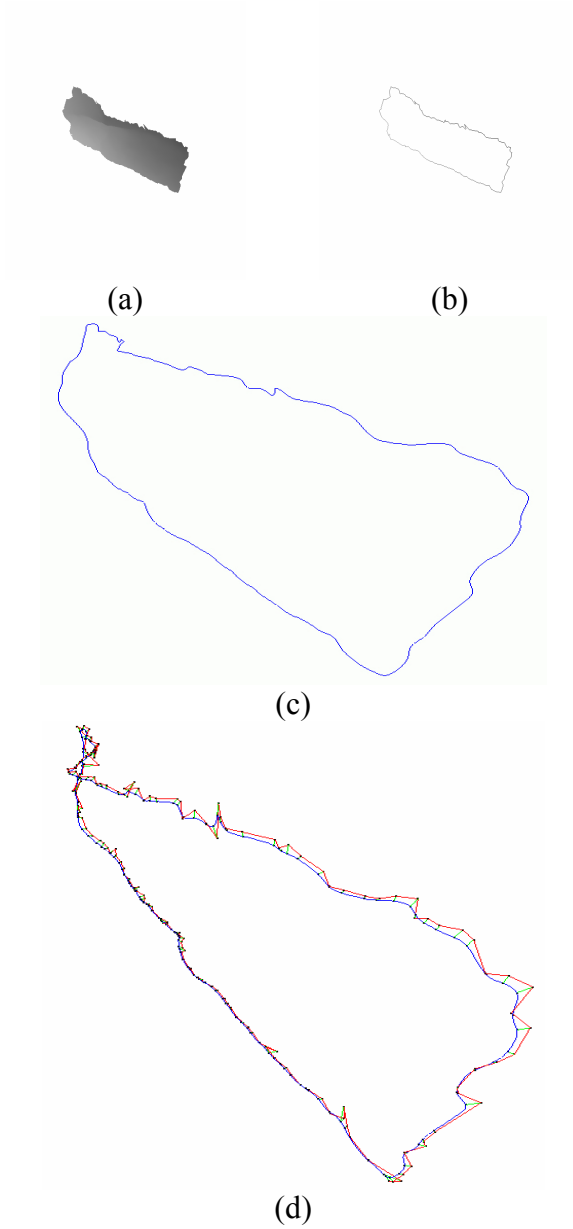


Fig. 2 Sampled outline creation process. (a) Acquired z-buffer of a surface. (b) Image-based boundary. (c) The smoothed reconstructed 3D outline. (d) Original to smoothed outline vertex mapping.

and $\boldsymbol{n}_{av}(S_{k,m})$, either $\boldsymbol{n}_{av}(S_{i,m},S_{j,n})$ or $-\boldsymbol{n}_{av}(S_{i,m},S_{j,n})$ coincides with the z axis, so that the surface's faces are not backfacing. If the average sum of any axis coefficients is near 0 (opposing vectors), we set it equal to the first surface's corresponding coefficient. This allows us to project each surface on a common reference plane, since the outline connection depends on both surfaces' average normal vectors. Next we render the surface into an $N_b \times N_b$ z-buffer using an orthographic projection $\mathbf{P}_{buf}$, acquiring its image-based representation (Fig. 2a), given by :

$$S_{k,m}^{(proj)} = P_{buf} Z_{n_{av}} T_{-c} S_{k,m} \qquad (2)$$

Z_v represents the transformation that aligns vector v with axis z and is defined as : $Z_v = R_{x,\theta} R_{y,\phi}$, where $R_{v,\theta}$ represents the rotation matrix around axis v by angle θ, $\theta = a\tan\left(v_y \Big/ \sqrt{v_x^2 + v_z^2}\right)$ and $\phi = -a\tan\left(v_x / v_z\right)$. $\mathbf{T}_{-c}$ represents the translation matrix by vector $-c$, where c is the centroid of all vertices in $S_{k,m}$.

We extract the outer boundary of all non-background pixels stored in the depth buffer (Fig. 2b), excluding small areas of unconnected pixels that may appear due to grave surface degradation and corrosion, and transform it back to three dimensions using the inverse transformation (Fig. 2c). Next we smooth the resulting boundary using a Gaussian filter on its nodes to finally acquire the desired sampled, smooth outline $H_{k,m}$. Filter size may vary depending on the resolution of the z-buffer we use. For our experiments a 512x512 sized z-buffer, along with a filter of width w = 6, performed ideally.

Finally, we map each vertex $v_{k,m}$(i) of the original outline to a point on $H_{k,m}$ by using a simple nearest neighbor algorithm with the constraint that a vertex $v_{k,m}$(i) can only be mapped to a point $h_{k,m}$(j) that is the same or precedes the point to which vertex $v_{k,m}$(i-1) was previously mapped. We thus avoid intersections between the triangles of the surface patch being created. Therefore each point $h_{k,m}$(j) is linked to a set of adjacent vertices of $O_{k,m}$ (Fig. 2d) .

3 Sampled Outline Point Connection

In order to connect points between outlines $H_{1,m}$ and $H_{2,n}$, corresponding to surfaces $S_{1,m}$ and $S_{2,n}$, we start by examining connections from $H_{1,m}$ to $H_{2,n}$. First we construct a vector $\boldsymbol{n}_{dir,12}$ defined by the average sum of surface $S_{1,m}$ average normal vector $\boldsymbol{n}_{av}(S_{1,m})$ and the vector defined by $c_{2,n} - c_{1,m}$, where $c_{2,n}$, $c_{1,m}$ are the centroids of outlines $H_{2,m}$ and $H_{1,n}$ respectively. This vector gives a fairly decent approach of the direction that point connection must follow in order to preserve the objects'

continuity and geometrical features, but also to facilitate point connectivity that can avoid face folding. Using vector $\boldsymbol{n}_{dir,12}$ as a guide we search for candidate connecting vertices and mark the best candidates as *STRONG* (Section 3.1). To avoid later triangle intersections we refine and exclude *STRONG* connections that cross-intersect, following a circular direction. We repeat this procedure for the inverse connection examination, from $H_{2,n}$ to $H_{1,m}$ (Fig 3b).

We next examine whether a connecting point in the set of $h_{1,m}$(i) has $h_{1,m}$(i) in its corresponding connection set and if such candidate connections exist they are additionally marked as *BIDIRECTIONAL* (Fig 3c) . Retaining only the connections marked as *STRONG* and *BIDIRECTIONAL* simultaneously, excluding any cross-intersections among them, and removing any other candidates, we create connections firstly from each unconnected point of $H_{1,m}$ to points of $H_{2,n}$, by interpolating indexes pointing to points $H_{2,n}$ given by *STRONG* and *BIDIRECTIONAL* connections (Section 3.2).

The inverse procedure is performed for any remaining unconnected points of $H_{2,n}$ to points of $H_{1,m}$. Now we have connected every point in each outline with a point on the matching outline avoiding intersections between them (Fig 3d).

3.1 Finding and Characterizing Candidate Connections

For each pair of points $h_{1,m}$(i), $h_{2,n}$(j), we compute the angle $\theta_{i,j}$ between vectors $\boldsymbol{n}_{dir,12}$ and $\boldsymbol{n}_{i,j} = h_{2,n}$(j) - $h_{1,m}$(i). If the angle lies within an error margin of [-e, e] degrees, we consider point $h_{2,n}$(j) as a connection candidate. If for any point of $H_{1,m}$ a candidate cannot be found we adaptively expand the error margin and redo the search. Now each point $h_{1,m}$(i) is linked to a set of points of $H_{2,m}$. We sort each point set by increasing absolute angle value $|\theta_{i,j}|$ and consider the candidate connection with the smallest angle as *STRONG* candidate (Fig 3a).

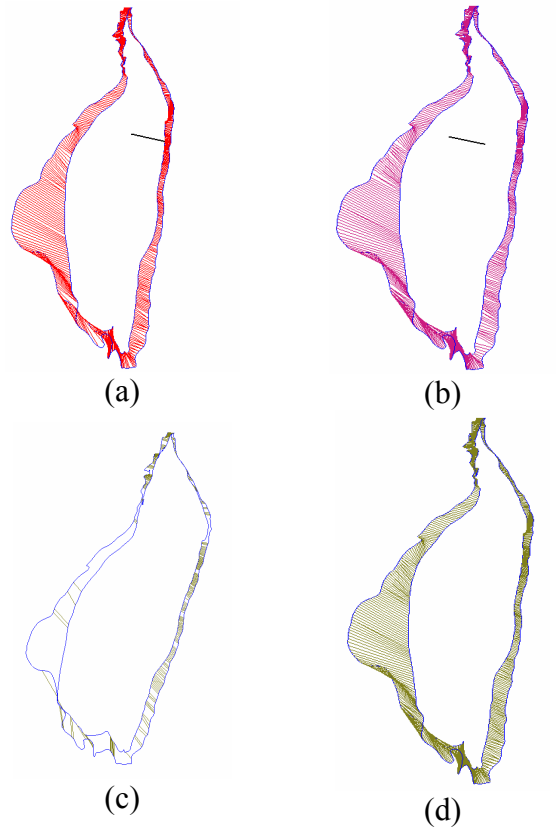


Fig 3. (a) *Strong* connections from $H_{1,m}$ to $H_{2,n}$. The vector $\boldsymbol{n}_{dir,12}$ is rendered in black. (b) *Strong* connections from $H_{2,n}$ to $H_{1,m}$. (c) Refined *Strong* and *Bidirectional* connections. (d) Final connections.

3.2 Point Connection Finalization

We create sets of adjacent unconnected points of $H_{1,m}$, $R_{1,i}$ = {$h_{1,m}(R_{1,i}(1))$,… $h_{1,m}(R_{1,i}(N)$}, where N is the number of points in the set. Points in set $R_{1,i}$ are bounded by points $h_{1,m}(R_{1,i}(1)$-1) and $h_{1,m}(R_{1,i}(N)$+1) which are connected to points $h_{2,n}$(k) and $h_{2,n}$(l) respectively. Another set $R_{2,j}$ = { $h_{2,n}$(k),…, $h_{2,n}$(l)} is created and each point in set $R_{1,i}$ is connected to a point in $R_{2,j}$, and vice versa, by interpolating between them.

4 Original Outline Vertex Connection

Each point of the smooth outlines $H_{1,m}$ and $H_{2,n}$, is linked to a set of vertices of the original outlines $O_{1,m}$ and $O_{2,n}$ respectively. Using the information provided by the connections between points of the smooth outlines we are able to connect sets of vertices of $O_{1,m}$ to sets of vertices of $O_{2,n}$ in the following manner :

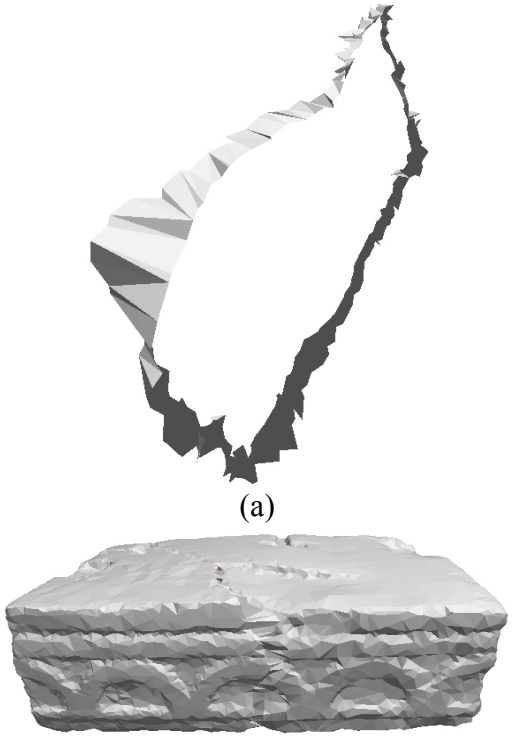
Starting from a connection between vertices $h_{1,m}$(i) and $h_{2,n}$(j), we create two sets R'_1 and R'_2 of vertices by adding the sets of vertices linked to $h_{1,m}$(i) and $h_{2,n}$(j). We examine adjacent vertices to $h_{1,m}$(i) and $h_{2,n}$(j) in order to find existing connections pointing to either of these points. If any are found, we add their linked vertex sets to R'_1 and R'_2 accordingly, either at their start or end, depending on the direction that the adjacent connecting vertex was found. For example, if point $h_{2,n}$(j+1), with linked vertex set $Vs_{2,n}$(j), was also connected to $h_{1,m}$(i), the new set would be equal to $R`_{new2} = \{R'_2, Vs_{2,n}(j)\}$. Operating on sets R'_1 and R'_2, we connect each vertex of R'_1 to a vertex of R'_2, and vice versa, by interpolating between them, as was done for connecting points of sampled outlines in the previous step. The procedure continues until all connections from $H_{1,m}$ to $H_{2,n}$ have been processed. For any remaining unconnected vertices of $O_{1,m}$ and $O_{2,n}$ we work in a similar manner, using interpolation between already connected vertices, to create new connections, primarily for $O_{1,m}$ and then for $O_{2,n}$.

We have thus achieved the connection of every vertex of $O_{1,m}$ to a vertex of $O_{2,n}$ and vice versa and are now ready to proceed with the construction of our surface patch.

5 Surface Patch Construction

The construction of our surface patch begins by inserting an edge for every vertex connection created in the previous step. These connections along with the edges derived by adjacent vertices of the original outlines represent a surface consisting of quads and triangles. To give the surface a unified form, each quad is split into two triangles.

Our surface patch preserves the boundaries between the fractured surfaces and their adjacent ones, a desired feature as the adjacent surfaces are not fractured and their geometry must remain intact. It also connects the matching objects based on the information derived from their geometry (Fig 4b).



(a) (b)

Fig. 4. (a) The triangulated surface patch. (b) The surface patch applied to repairing the fractured objects.

6 Conclusions and Further Work

The proposed surface patches preserve object continuity, geometry and detail. In most cases they offer a fairly attractive complement for the fractured objects as their original geometry remains intact and only undesired geometry, such as fractured surfaces, is removed. The simplicity and speed of the proposed method is ideal for mending matching broken artefacts once their proper alignment has been determined. The lesser the distance between them the more coherent the surface appears with respect to the matching parts. In addition, the sharper the common edges along different surfaces are, the more accurate the segmentation results, which further improve surface continuity. As presented in our experiments, the method performs well for any form of object, independently of the noise a surface may possess (Fig. 5).

However its purpose is not to reproduce fine details along the geometry of objects as in hole-filling or surface in-painting techniques, thus a comparison between them would be imprecise. This is the goal of our future

research. We intend to further improve the surface patches by extracting features of adjacent non-fractured surfaces and experimenting with different types of patches, such as Bezier patches.

7 Acknowledgements

Financial support from the Hellenic General Secretariat for Research and Technology (GSRT) under the PENED programme is acknowledged.

8 References

[1] G. Papaioannou, E. A. Karabassi, On the Automatic Assemblage of Arbitrary Broken Solid Artefacts, *Image & Vision Computing*, 21(5), pp. 401-412, Elsevier, 2003.

[2] Huang, Q., Flöry, S., Gelfand, N., Hofer, M., and Pottmann, H. 2006. Reassembling fractured objects by geometric matching. *In ACM SIGGRAPH 2006 Papers (Boston, Massachusetts, July 30 - August 03, 2006). SIGGRAPH '06. ACM Press*, New York, NY, 569-578

[3] Jonathan C. Carr, Richard K. Beatson, Jon B. Cherrie, Tim J. Mitchell, W. Richard Fright, Bruce C. McCallum, and Tim R. Evans. Reconstruction and representation of 3D patterned objects with radial basis functions. *SIGGRAPH 2001, Computer Graphics Proceedings*, pages 67.76. ACM Press / ACM SIGGRAPH, 2001.

[4] Davis, J., Marschner, S. R., Garr, M., and LevoyY, M. 2002. Filling holes in complex surfaces using volumetric diffusion. In Proceedings of the 1st International Symposium on 3D Data Processing Visualization and Transmission (3DPVT-02), IEEE Computer Society, Los Alamitos, CA, G. M. Cortelazzo and C. Guerra, Eds., 428.438

[5] Papaioannou G., E.A. Karabassi & T. Theoharis, "Segmentation and Surface Characterization of Arbitrary 3D Meshes for Object Reconstruction and Recognition", *IEEE Int. Conference on Pattern Recognition*, Barcelona, Spain, September 2000, pp. 734-737.

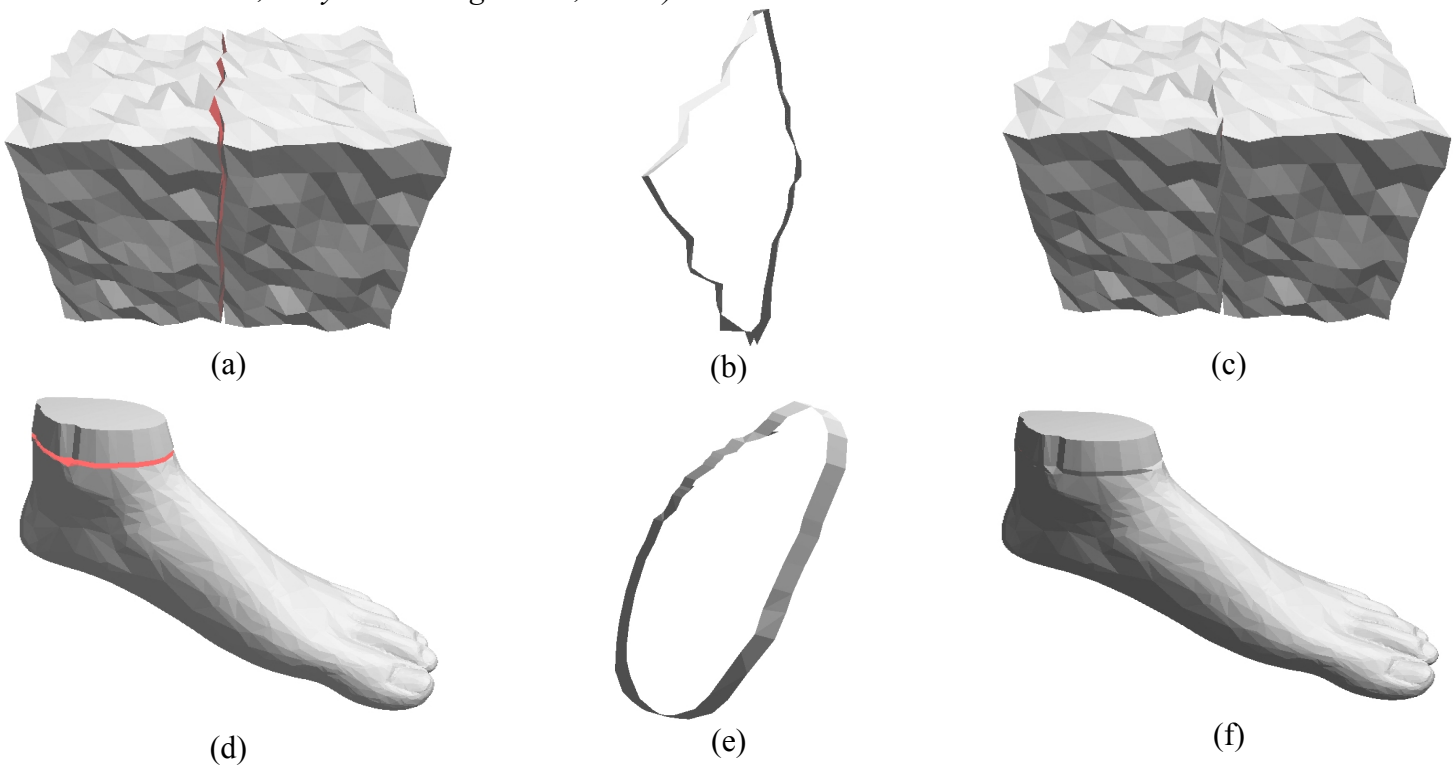


Fig. 5. Reconstruction procedure of a foot and noisy box objects. (a),(d) Aligned models. (b),(e) Surface patch. (c),(f) Repaired objects.