Interactive tools for quality enhancement in 3D modelling from reality

Mauro Ruggeri, Paulo Dias, Vítor Sequeira, João G.M. Gonçalves[*]

European Commission-Joint Research Centre, TP 270, 21020 Ispra (VA), Italy

Abstract. This paper presents a set of algorithms to improve the quality of triangulated 3D models representing real environments. The objective of the work is to improve the details of the 3D models (acquired using laser range sensors), which the automated 3D reconstruction procedures are not able to describe. The approach followed in this work is to apply specific geometric algorithms either on the entire model or in particular regions to correct data imprecision in respect to the real word. This includes edge straightening, constraint plane flattening and other general corrections. The techniques presented are based on the automatic/semi-automatic edge detection either on the triangulated model or on the combination of registered video and range images. This is followed by the mesh modification to accommodate the edge straightening. Finally, the paper proposes an extension of the data correction concepts to surfaces by introducing a planar flattening algorithm. These are the foundations for the further, more abstract, surfaces correction.

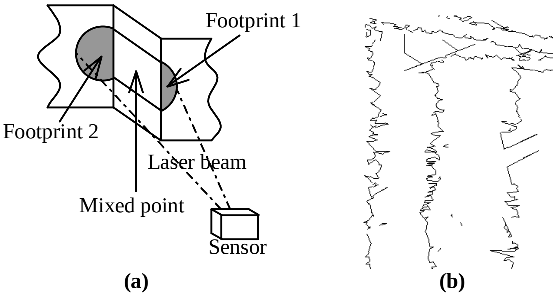
Keywords: 3D Reconstruction, Edge extraction, Mesh optimisation, Scene interpretation.

1 Introduction

Many virtual and augmented reality computer applications require accurate 3D representations of real environments. The most efficient techniques used to build realistic 3D model are based on laser range sensors combined with texture colour images, [4,5,6]. In order to make the acquired models more suited for computer graphics applications, they are described as triangular meshes. Many details of the real environment are not, however, perfectly described. This is due to the following factors:

a) distance measurements;
b) spatial resolution (limited by the laser footprint and scanner resolution);
c) triangulation procedure.

One of the most important reasons for measurement errors is that the laser footprint is not an ideal point but more like a disc; this produces discontinuities near the boundary of the surfaces and a straight line may appear like a sawtooth or a jagged line (see Figure 1b). This problem is known as "the mixed point problem" (Figure 1a). In fact, the distance measured by the laser sensor does not reflect the real, but rather a combination of the distances to both parts of the footprint. The effect of this imprecision is to produce a very worn out look especially in modelling of non-made environments (e.g. architectural) where straight or geometrically perfect edges abound.



(a) **(b)**

Figure 1: The mixed point problem (a) and its effect on a rectilinear edges (b).

The scope of the quality enhancement techniques is to correct, in the most efficient way, all the discrepancies from reality existing in the scanned 3D model. The first problem to be treated is the edge straightening which is characterized by the:

1. detection of the real edge from the different acquired datasets;
2. recognition of the correspondent edge in the triangulated 3D model;
3. effective straightening of the edge by re-triangulating the original mesh.

---

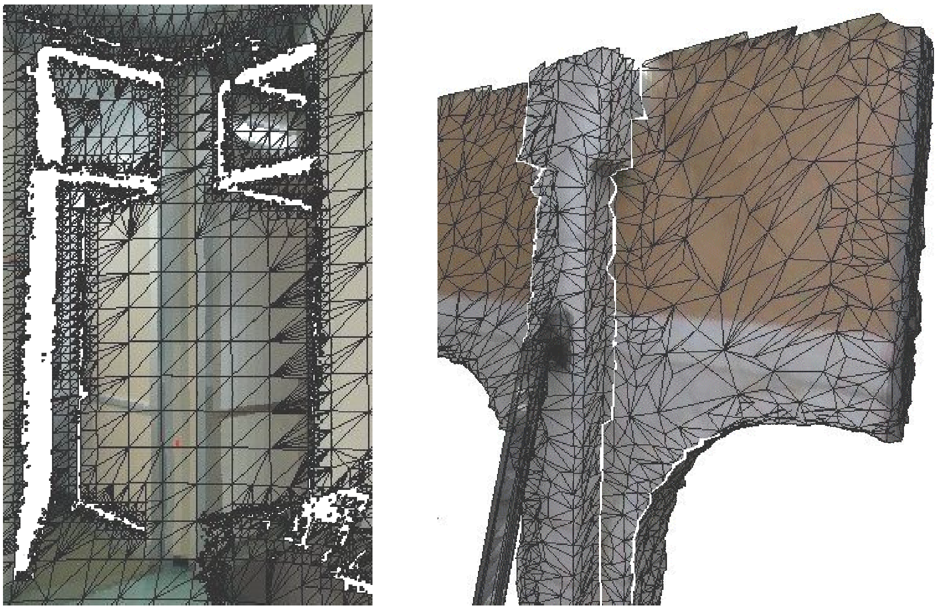
[*] E-mail: {mauro.ruggeri, paulo.dias, vitor.sequeira, joao.goncalves}@jrc.it

2 Edge straightening

Edges are an important visual cue about the shape of objects [1] and are essential to reach realistic representations of environments especially in the fields of architecture, design, culture heritage, and entertainment applications (i.e. basically in all non-made environments). Due to the problems described, a triangulated 3D model can have different type of edges to be straightened. This depends on the particular conformation of the surfaces from which the edges have been generated. In particular this paper will consider two of the most frequent types of edges: *jump edges* and *roof edges*. Both are assumed to correspond to straight lines.

A jump edge is created when the laser sensor scans surfaces with depth discontinuities, such as, the boundary of a door (Figure 2a).

A roof edge is obtained when the laser sensor scans a connection of surfaces with different normal orientation, such as, the boundary of a squared column (Figure 2b).

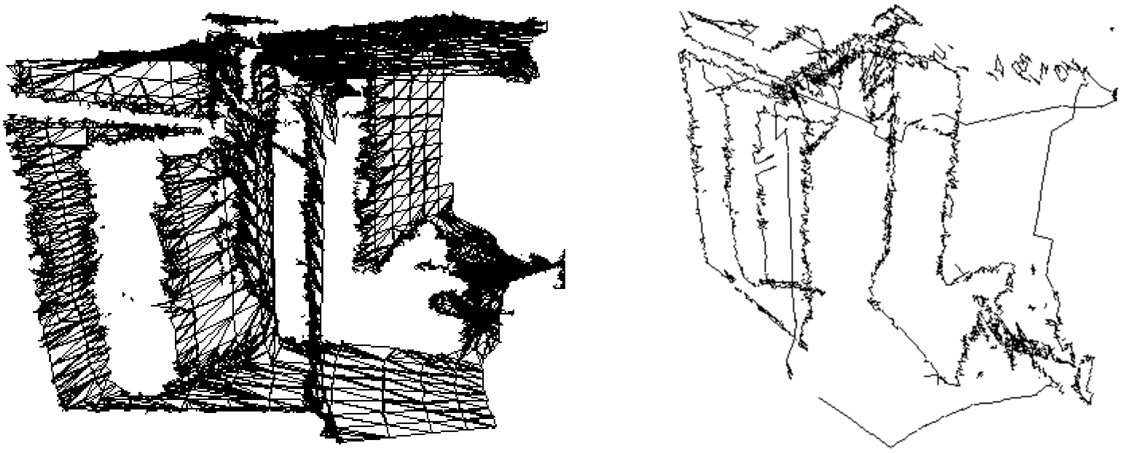


(a) (b)

Figure 2: Example of jump edge (a) and roof edge (b).

2.1 Edge detection

The first step in the straightening procedure is the detection of 3D edges in the 3D model. Jump edges are relatively easy to detect as they are shared by only one triangle and a suitable data structure (optimising the navigation through the topology of the model) makes the whole operation quick and efficient. Figure 3 presents the results of the jump edge detection procedure from a laboratory test scene.



(a) (b)

Figure 3: The original triangulated model (a) and the results of the jump edge detection (b) in a laboratory scene.

The roof edges detection consists in checking the angle between the triangles sharing the edges. If the angle is higher than a user given threshold, the edge is considered as a valid roof edge, otherwise the triangles are assumed to be coplanar.

Detecting the 3D edges in the model is not sufficient since the straightening algorithm need a *hypothetic edge* line segment as approximation of the real edge. This *hypothetic edge* can be supplied manually (see section 2.2) using a friendly and interactive tool or automatically detected.

The automatic procedure uses high-resolution digital images registered with the range images during the texture mapping phase. First, straight lines are detected in the digital images (in white in Figure 4a). Secondly, the 2D intensity and 3D model edges are compared based on the camera model to select the 3D edge points (in white in Figure 4b) with high probability to lie on a line (in black in Figure 4b). For each line, the hypothetic edge line is then computed as the orthogonal linear regression of all the 3D points [2]. Figures 4c and 4d show the original jagged 3D edges (in light grey) and the computed hypothetic 3D edges (in black).

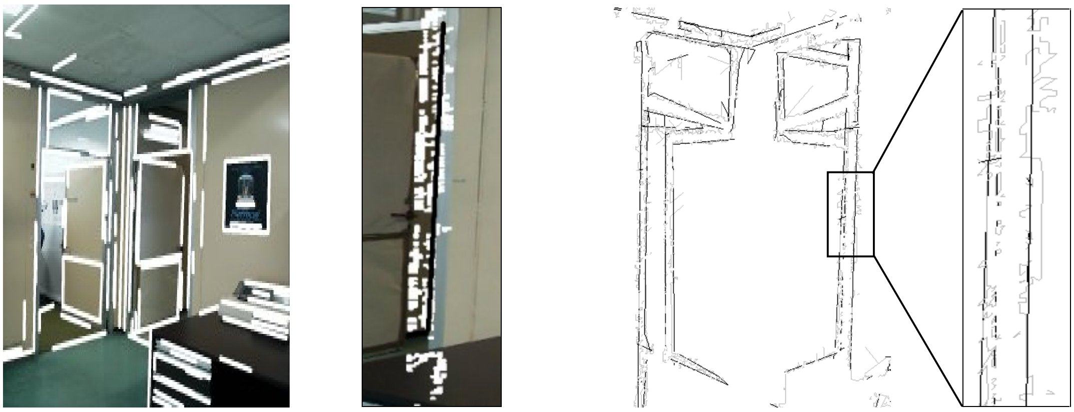

(a) (b) (c) (d)

Figure 4: The automatic hypothetic edge detection procedure (see text for details).

2.2 *Association between hypothetic edge and 3D model*

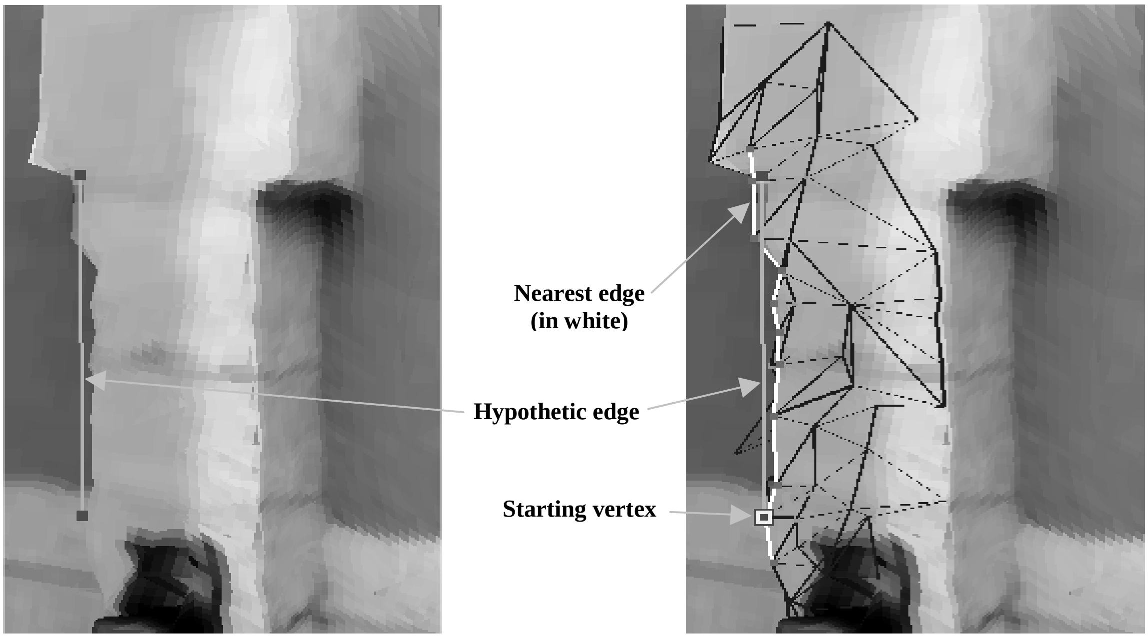
When a hypothetic 3D edge has been detected (manually or by mean of the automatic procedure described in previous section), a specific straightening procedure is applied to the triangulated 3D mesh. In order to choose the best straightening method is necessary to analyse the topology of the surrounded 3D mesh and extract all information needed for changing the triangulation. The topology analysis includes an initialisation step to speed up the computation and consists of the following operations:

- Set up the geometric parameters describing the candidate edge to be straightened.
- Bound the computation area (Figure 5b) in order to optimise and speed up the algorithm.
- Find the *nearest edge* (white path in Figure 5b).
- Selection of the best straightening method.

The *nearest edge* is the connected 3D edge path closest to the hypothetic edge in the triangulated 3D model. It represents the real 3D sawtooth edge to be straightened as well as the starting geometric 3D entity for the further straighten algorithms. The steps to find the nearest edge are:

1. Find the nearest triangle vertex: *Starting vertex* (squared point in Figure 5b).
2. Extracts iteratively the triangle edge that is nearer to the hypothetic edge and is connected to the previous edge.
3. Stop when the extracted connected edge length (distance between the 3D orthogonal projections of its extremes on the hypothetic edge line) exceeds the hypothetic edge length (Figure 5b).

Figure 5 shows the result of the above procedure on the 3D model mesh. Figure 5a shows the selected hypothetic edge and Figure 5b the result of the initialisation procedure. In Figure 5b the isolated 3D triangle mesh are represented in black and the nearest detected 3D edge in white.



(a) (b)

Figure 5: Selection of edge to be straightened in the model (see text for details).

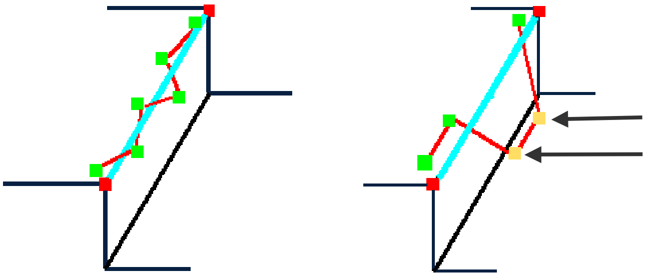
2.3 Straightening methods

The selection of the most convenient straightening technique is fundamental to reach a more realistic result, as well as, to keep the topological information of the acquired 3D model. Two different straightening methods have been used:

1. projection straightening;
2. re-triangulation straightening.

Projection straightening consists in a 3D orthogonal projection of the *nearest edge* vertices directly into the hypothetic edge line, provided by the user or computed by the algorithm in section 2.1. Re-triangulation straightening performs a new triangulation of the mesh around the nearest edge.

These two techniques follow different approaches and are strictly depending on the 3D model mesh topology. Figure6a shows that a simple projection of the nearest edge vertices would result in a lost of important depth information (the step would disappear). Thus, is fundamental to base the selection of the straightening method on the mesh topology to increase the quality of the 3D model to reduce the complexity of the computation.



Using the projection straightening the depth information in these points would be lost.

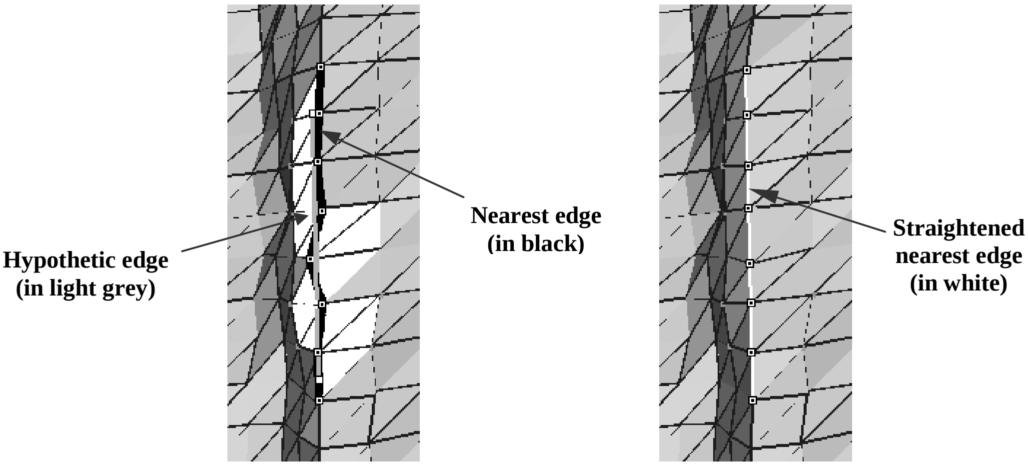
(a) (b)

Figure 6: Edge straightening: (a) projection, (b) re-triangulation.

The best straightening method is automatically estimated by isolating the closest connected triangle path to the hypothetic edge and that shares the nearest edge. If this path presents relevant oscillation in respect to the hypothetic edge (Figure 7a) the projection straightening will be chosen. Otherwise, the re-triangulation straightening (Figure 8a) is chosen.

2.3.1 Projection straightening

During projection straightening all vertices belonging to the nearest edge are projected, (with a 3D orthogonal projection) directly into the hypothetic edge line. The result of this operation can be seen in the Figure 7. Figure 7a shows as squared points the vertices to be projected into the hypothetic edge and in black the computed nearest edge. Figure 7b shows the result of the straightening. The white triangles in the Figure 7a identify the closest connected triangle path to the hypothetic edge presenting oscillations.



(a) **(b)**

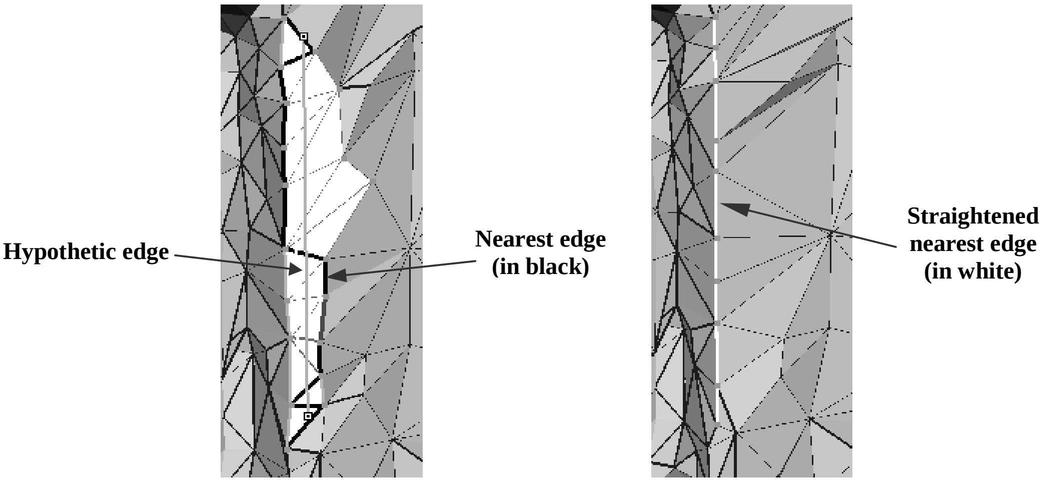
Figure 7: Projection straightening: before (a) and after projection (b).

2.3.2 Re-triangulation straightening

Re-triangulation straightening changes the triangulation of the mesh in the neighbourhood of the nearest edge. The fundamental steps of this procedure are:

1. Isolate a connected triangle path (done during the selection of the straightening method).
2. 3D orthogonal projection of the triangles path vertices on the hypothetic edge line segment.
3. Delete triangle path.
4. Adjust triangles near the path.

The white triangle strip in Figure 8a identifies the closest connected triangle path to the hypothetic edge that will be deleted. In respect to the triangle path shown in the Figure 7a, this one does not present any oscillations. Figure 8b shows the mesh after re-triangulation.



(a) **(b)**

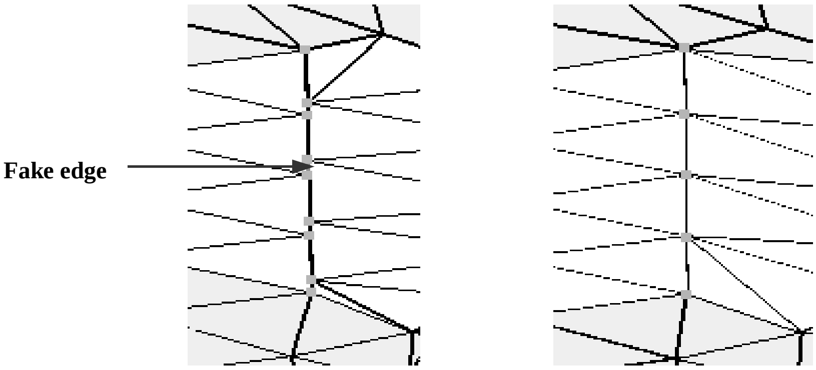
Figure 8: Re-triangulation straightening: before (a) and after algorithm (b).

The last step of the straightening procedure is the most important because adjusts the topology of the mesh just changed. In fact steps 2 and 3 can introduce many triangulation anomalies depending on the conformation of the mesh.

The principal triangulation problems handled are introduced below:

- non-anchored triangles (Figure 9);
- inverted surface normal triangles (Figure 10);
- point concentrated triangles (Figure 11).

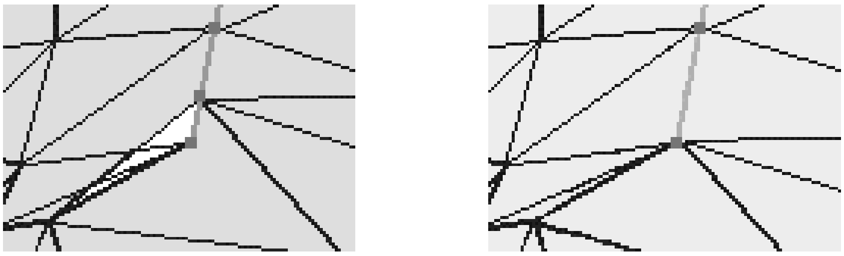
The non-anchored triangle problem occurs when the mesh contains vertices belonging to a *fake edge*. A fake edge does not create any triangle (Figure 9a); hence it generates topology anomalies in the mesh that should not be univocally represented. To avoid this problem the triangles have to be aligned by collapsing the fake edge vertices.



(a) **(b)**

Figure 9: Non-anchored triangles: before (a) and after alignment (b).

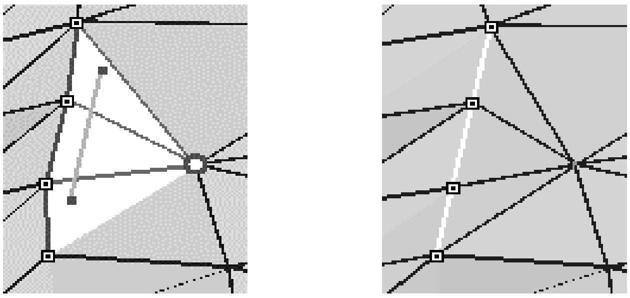
When a triangle has the surface normal inverted it is necessary to flip it by changing the order of its vertices, however this procedure is not always enough. In fact, as in the case of the Figure 10a, when this type of triangle contains a start or an end edge point a simple flipping produces an unwanted edge intersection with consequent mesh anomalies. To correct this problem, such triangle is deleted and vertices sharing the connected triangle path are collapsed (Figure 10b).



(a) **(b)**

Figure 10: Inverted surface normal triangle: before (a) and after correction (b).

The last problem arises when the continuous connected triangle path is made of point concentrated triangles represented by those triangles sharing a common vertex as shown in the Figure 11a. In such a case the mesh becomes more difficult to handle and the normal straightening procedure could cause triangulation anomalies (e.g., in the case of Figure 11a, the concentration of the mesh increase in one point). Thus these types of triangles require a particular attention and the checking of this problem can be fundamental for the topology of the resulting mesh. In the example of Figure 11 the algorithm performs a simple projection of the squared points in to the hypothetic edge (Figure 11b).



(a) **(b)**

Figure 11: Point concentrated triangle: before (a) and after correction (b).

2.4 Experimental results

The above techniques have been implemented as part of an interactive tool for quality enhancement of the triangulated 3D models. This tool aims to give to the user, a friendly, efficient and assisted way to apply geometric and topological corrections to the 3D models. To reach these objectives the user interface has been designed to accommodate the user correction operations. In fact to make a correction, the user only indicates the type of action and the region where to focus in the model. After these simple operations the correction algorithm automatically starts. In the case of edge straightening after the hypothetic edge is selected the algorithm automatically suggests the best straightening method.

Figure 12 shows the result of a jump edge projection straightening. Figure 12a shows the mesh before the algorithm is applied and Figure 12b illustrates the mesh after the projection straightening procedure.

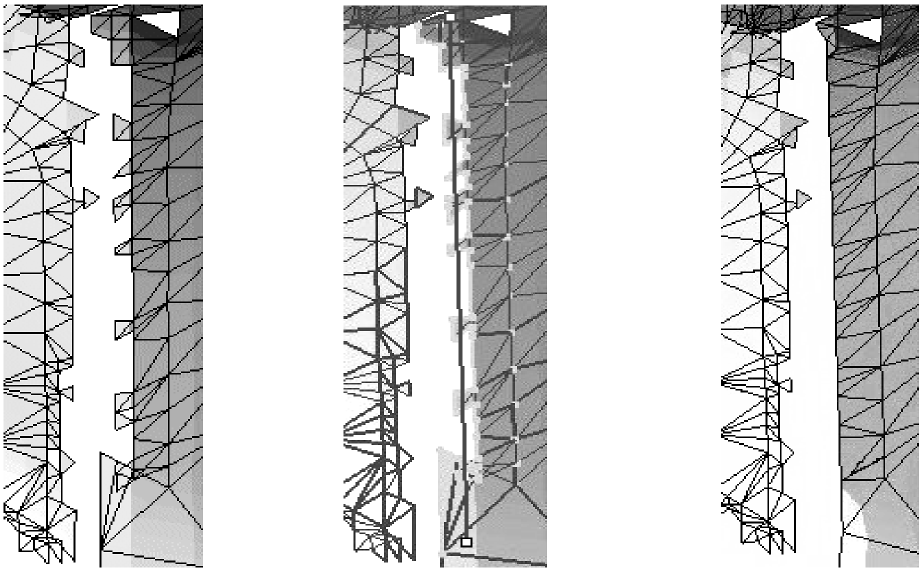

(a) (b) (c)

Figure 12: Jump edge projection straightening: original edge (a), jump edge detected (b) and jump edge straightened (c).

Figure 13 shows the result of the re-triangulation straightening applied on a higher part of the column represented in the Figure 2b. The highlighted triangle strip identifies the connected triangle path that will be deleted.

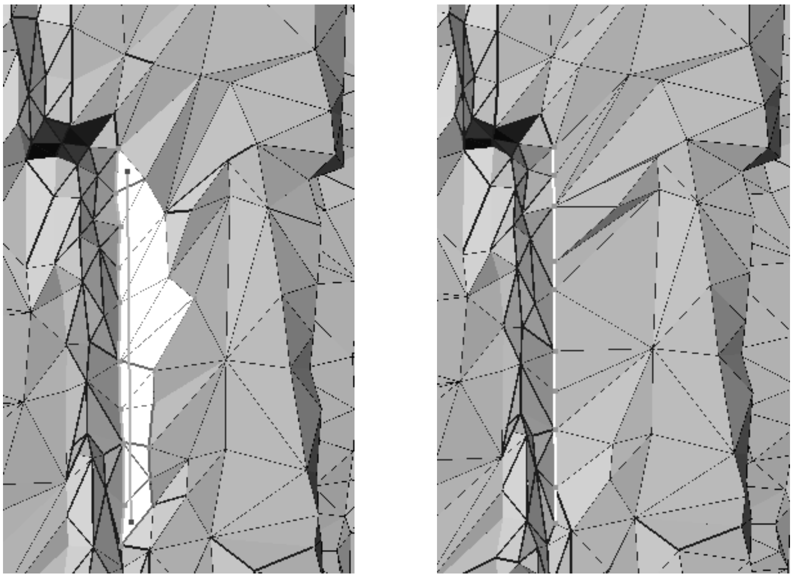
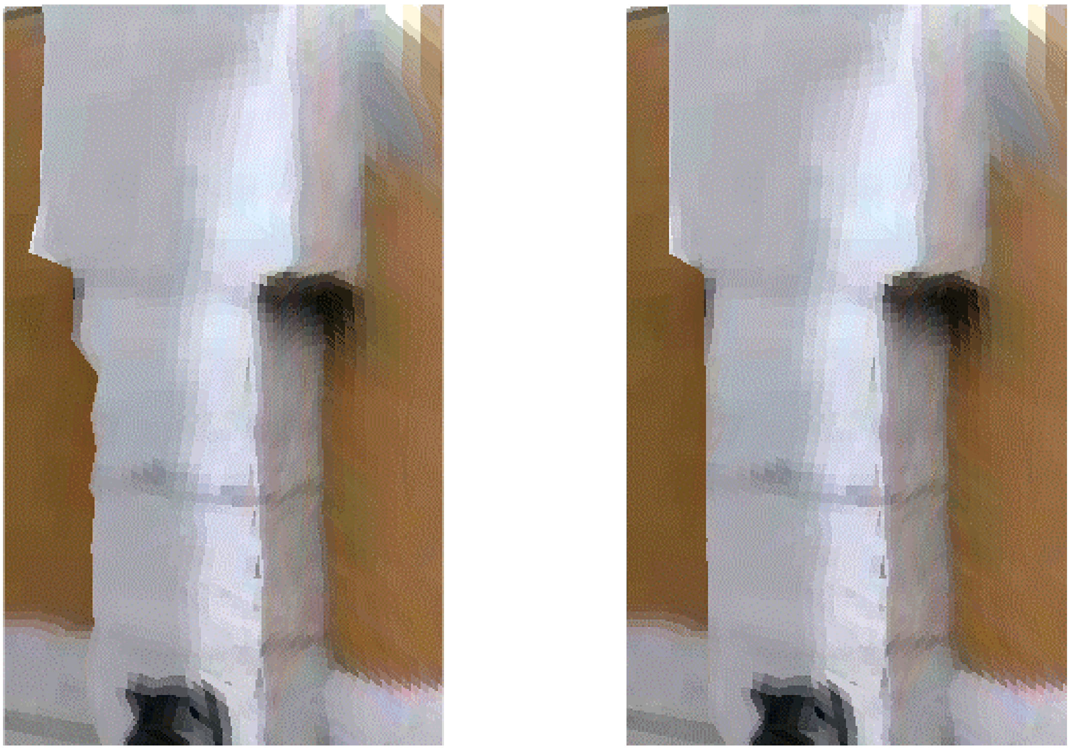

(a) (b)

Figure 13: Re-triangulation straightening: before (a) and after (b) the algorithm.

Finally, Figure 14 shows the re-textured version of the Figure 13 to illustrate the quality enhancement produced by the edge straightening in the modelling of an architectural scene.



(a) (b)

Figure 14: Edge straightening result: before (a) and after (b) the algorithm.

3 Surfaces correction

The next step in quality enhancement is to extend the concept of straightening to a general surface. In fact the edges treated above can be seen as the progenitor of a family of important visual cues. For example, when examining the column and the arcs of the Figure 2b, they seem corroded and irregular due to noise in the scanned data. In reality the column faces are four planes, the edges are straight lines and the arcs are regular. Moreover, having a mathematic surface eases the process of texture mapping and the acquired texture could represent more efficiently the real materials such in the example of Figure 15.

The first step of surface correction is to consider the easiest surface, the plane. An algorithm of plane recognition and flattening can be combined with the above edge straightening methods to define a more abstract object correction such as a plane corner. This is very useful in the case of columns, roof and floor corners, and in each case where there is a 3D object having a corner between planes, very frequent in indoor environments.

The main idea is to locate constrains in the model, based on the hypothetic edge that define the corner, to fit surfaces around it into a *planes corner object*. These constrains could be set like the hypothetic edge or by looking for the nearest bounding edges. When the bounding edges are isolated the triangles vertices between constrains are fitted on the best plane, computed by using the orthogonal regression planar fitting [3].

The algorithm minimizes the orthogonal distance error function using least squares. Given $\vec{\mathrm{N}} = (\vec{\mathrm{X}} - \vec{\mathrm{A}})$ the hyperplanar equation, where $\vec{\mathrm{N}}$ is a unit length normal to the hyperplane and $\vec{\mathrm{A}}$ is a point on it, the algorithm minimize the following energy function:

$$E(\vec{A},\vec{N}) = \vec{N}^t\left(\sum_{i=1}^{m}\vec{Y}_i\vec{Y}_i^t\right)\vec{N}^t = \vec{N}^t M(A)\vec{N}$$

$$\text{where, } M(A) = \begin{bmatrix} \sum_{i=1}^{m}(x_i-a)^2 & \sum_{i=1}^{m}(x_i-a)(y_i-b) & \sum_{i=1}^{m}(x_i-a)(z_i-c) \\ \sum_{i=1}^{m}(x_i-a)(y_i-b) & \sum_{i=1}^{m}(y_i-b)^2 & \sum_{i=1}^{m}(y_i-b)(z_i-c) \\ \sum_{i=1}^{m}(x_i-a)(z_i-c) & \sum_{i=1}^{m}(y_i-b)(z_i-c) & \sum_{i=1}^{m}(z_i-c)^2 \end{bmatrix}$$

The above algorithm increases significantly the quality of the model as shown in the Figure 15. The line resulting by the intersection of the corner planes just found can be matched with hypothetic edge direction line in order to estimate the error of a predefined edge detection procedure.

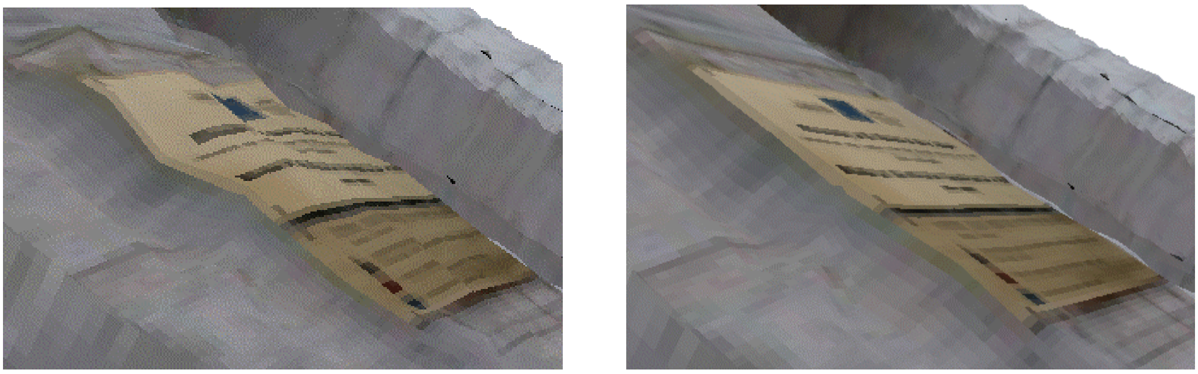

(a) **(b)**

Figure 15: Plane flattening: before (a) and after (b) the orthogonal regression planar fitting.

4 Conclusion and future work

The work presented in this paper has given good results using different scenes and models. To make the whole procedure more reliable, current work is underway to improve the efficiency of the automatic algorithms especially in the mesh optimisation algorithm.

The arguments treated in the paper are the foundations of a more general 3D quality enhancement system that has the scope to become more general and completely automatic. Future work will be oriented to deal with further quality enhancement problems such as:

- *Hole filling:* They can be created by dropouts in the range data due to highly reflective surfaces or by objects hiding parts of further away objects, also known as occlusions.
- *Removing Surface anomalies:* They can be created by the data integration procedure, and include duplicate polygons and vertices, inverted surface normal, unconnected meshes, etc.
- *Debris removal:* Small groups of triangles are generated as a result of insufficient scanning resolution or noisy scan data.
- *Spike flattening*: Noise in the range data can create spikes that are turned into little pyramids sticking out of planar surfaces. As they create shading discontinuities in an otherwise constantly lit surface they have a large distraction effect.

5 Acknowledgments

This research was funded by the EC CAMERA TMR (Training and Mobility of Researchers) network, contract number ERBFMRXCT970127.

6 References

[1] K. Choo, I.D. Yun and S.U. Lee, “Edge-Based Approach to Mesh Simplification”, in Second International Conference on 3-D Imaging and Modeling, Ottawa, Canada, 1999.

[2] P. Dias, V. Sequeira, J.G.M Gonçalves, F. Vaz – “Fusion of Intensity and Range Data for Improved 3D models”, ICIP01 - IEEE International Conference on Image Processing, Thessalonica, Greece, October 7-10, 2001

[3] D.H.Eberly, “3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics”, Morgan Kaufmann Publishers, September 2000.

[4] S. F. El-Hakim, C. Brenner, G. Roth, "A multi-sensor approach to creating accurate virtual environments", *ISPRS Journal for Photogrammetry and Remote Sensing*, Vol. 53(6), pp. 379-391, December 1998.

[5] V. Sequeira, K. Ng, E. Wolfart, J.G.M. Gonçalves, D. Hogg, "Automated Reconstruction of 3D Models from Real Environments", ISPRS Journal of Photogrammetry and Remote Sensing (Elsevier), vol. 54, pp. 1-22, 1999.

[6] I. Stamos, P. K. Allen, "3-D Model Construction Using Range and Image Data", IEEE International Conference on Computer Vision and Pattern Recognition, pages 531-536 (volume I), South Carolina, June 2000.