



UNIVERSITÀ  
DEGLI STUDI  
FIRENZE

## FLORE

# Repository istituzionale dell'Università degli Studi di Firenze

### **Tactile 3D bas-relief from single-point perspective paintings: a computer based method**

Questa è la Versione finale referata (Post print/Accepted manuscript) della seguente pubblicazione:

*Original Citation:*

Tactile 3D bas-relief from single-point perspective paintings: a computer based method / R.Furferi; L. Governi; Niccolò Vanni; Yary Volpe. - In: JOURNAL OF INFORMATION AND COMPUTATIONAL SCIENCE. - ISSN 1548-7741. - ELETTRONICO. - 11(16):(2014), pp. 5667-5680. [10.12733/jics20104135]

*Availability:*

The webpage <https://hdl.handle.net/2158/868319> of the repository was last updated on

*Published version:*

DOI: 10.12733/jics20104135

*Terms of use:*

Open Access

La pubblicazione è resa disponibile sotto le norme e i termini della licenza di deposito, secondo quanto stabilito dalla Policy per l'accesso aperto dell'Università degli Studi di Firenze (<https://www.sba.unifi.it/upload/policy-oa-2016-1.pdf>)

*Publisher copyright claim:*

La data sopra indicata si riferisce all'ultimo aggiornamento della scheda del Repository FloRe - The above-mentioned date refers to the last update of the record in the Institutional Repository FloRe

(Article begins on next page)

# Tactile 3D Bas-relief from Single-point Perspective Paintings: A Computer Based Method<sup>\*</sup>

Rocco Furferi<sup>\*</sup>, Lapo Governi, Niccoló Vanni, Yary Volpe

*Department of Industrial Engineering, University of Florence, Italy*

---

## Abstract

Visually impaired and blind people are more excluded from equal access to the world's visual culture, thus being often unable to achieve concrete benefits of art education or to experience the life enhancing power of art. Despite the fact that all over the world several initiatives based on the interaction with sculptures and tactile three-dimensional reproductions or architectural aids on scale have been devised, a “sensory barrier” between blind people and pictorial artworks still exists, obviously due to their intrinsic 2D format. Translation of paintings into tactile models in the form of bas-reliefs is recognized to be one of the most effective ways for breaking down such a barrier, thus allowing a tangible improvement of the blind people's accessibility to artworks. In the recent years, computer aided technologies for automatic or semi-automatic translation of paintings into tactile models dramatically have been devised all over the world, thus improving blind people's accessibility to artworks. Inspired by some of these impressive works, this paper describes a user-driven methodology for the semi-automatic generation of tactile 3D models starting from paintings. Particularly devoted to the reconstruction of pictorial artworks characterized by single-point perspective, the proposed method relies on an interactive Computer-based modelling procedure. The method has been tested on two masterpieces of the Italian Renaissance period.

*Keywords:* Blind People; Tactile Bas-relief; 3D Computer Based Modelling; Single-point Perspective; Haptic Exploration

---

## 1 Introduction and Research Aims

Whilst early as the eighties much National legislation affirmed the need to remove the architectural barriers in public buildings, only very recently the importance of the “sensory barrier” has been taken into account. This is defined as the inability to freely perceive the artworks through forms of direct or indirect interaction, with respect to the people who are in need of special care [1, 2].

The access to three-dimensional representations has been the first degree of interaction for visually impaired people in museums. Nowadays, numerous initiatives based on the interaction with sculptures and tactile three-dimensional reproductions or architectural aids on scale have

---

<sup>\*</sup>Project supported by the Tuscany Region of Italy (PAR-FAS 2007-2013-Call identifier 1.1.a.3).

<sup>\*</sup>Corresponding author.

*Email address:* [rocco.furferi@unifi.it](mailto:rocco.furferi@unifi.it) (Rocco Furferi).

been developed. Many of these reproductions are manually realized by artists using materials like clay, plaster, wood, bronze, etc. Only in few cases some advanced technological systems like 3D scanning [3]-[5] and Rapid Prototyping (RP) [6] have been used, especially to improve the 3D reproduction of sculptures.

Obviously, cultural heritage in the form of two-dimensional art (e.g. paintings) cannot be directly reproduced into a 3D model but it needs to be translated into an appropriate “object” to be touched. In almost every scientific work dealing with this subject, the common criterion is to find a method for translating paintings into one of the two following representations: “tactile diagrams” and “bas-reliefs”. Tactile diagrams are not the exact relief reproduction of visual images: rather, they are translations of visual images into a tactile language and are generally used in conjunction with verbal narratives that guides the user through the diagram in a logical and ordered manner [7, 8]. Unlike tactile diagrams, bas-relief representation of paintings convey 3D information in a more “realistic” way by improving depth perception and, as demonstrated in recent studies [9], this representation is the most “readable” and meaningful for blind and visually impaired people. Unfortunately, the manufacture of such tactile media is a complex task, and to date entails the work of specialized artists (sculptors) thereby drastically reducing the production rate of such representations and the total number available.

From this point of view, computer aided technologies for automatic or semi-automatic translation of paintings into tactile models could dramatically improve blind people’s accessibility to artworks. An interesting attempt in introducing CAD-related technologies is proposed in [10] where a system capable of producing replicas of handmade tactile paintings using a 3d scanner is described. However, the input of the proposed system is still a physical bas-relief realised by sculptors. A number of algorithms, coming from computer vision science, exist to cope with the issue of computer aided bas-reliefs reconstruction. In some of these, e.g. [11] and [12] just to cite a few, the bas-relief is obtained starting from 3D surface models. Nonetheless, such a family of approaches is not suitable for reconstructing paintings where the input is an image since it requires, at least, a series of intermediate steps to be adapted.

Several authors investigate the problem of relief reconstruction from single images [13]-[19]. For instance, image embossing [20] is a common approach for recovering a digital relief from single image; embossing an image is a computer graphics technique in which each pixel of an image is replaced either by a highlight or a shadow, depending on boundaries on the original image. Unfortunately, the obtained result is a fake relief since it only visually resembles a bas-relief with incorrect depth reconstruction (due to the algorithm based on image gradient computation).

In a recent paper [21] an approach for estimating the height map from single images representing Brick and Stone Reliefs (BSR) has been proposed. The method proved to be adequate for restoring BSR surfaces by using a height map estimation scheme consisting of two levels: the bas-relief, referring to the low frequency component of the BSR surfaces, and the high frequency detail. However, as declared by the authors, due to the assumption of BSR priors, the method is not satisfactory when the relief to be generated complicatedly varies along the relief height.

One of the most recent methods for creating tactile paintings is the one proposed by [22]. In such a work the high resolution image of the painting to be translated into bas-relief is manually processed in order to 1) Extract the painted subject’s contours, 2) To identify semantically meaningful areas and 3) To assign appropriate height values to each area. One of the results consists of “a layered depth diagram made of a number of individual shapes cut out of flexible sheets of constant thickness, which are glued on top of each other to resemble a painting” [22].

This method is required to assign height relations for each area regardless to the information coming from scene perspective. While this approach may be extremely useful in case perspective related information is lacking in the considered paintings, it is not the best option in case of perspective paintings. In fact, the reconstructed scene could be geometrically inconsistent, thus causing misperception by blind people.

Inspired by some of the above cited relevant works, this paper aims to describe a new methodology for the semi-automatic generation of tactile 3D models starting from paintings characterized by single-point perspective typical of Italian Renaissance. More in particular, the goal of the proposed work is to provide a schematic three-dimensional representation of the scene depicted in the artwork, by identifying the subjects and the elements composing the scene itself and assigning them a proper depth. Since, as described below, the semi-automatic procedure requires user interaction, a GUI has been implemented.

In the presented methodology, depth variation takes into account perspective rules; for instance, two objects, whose real-world distance is assigned along the viewing direction, are represented onto two separate layers that are relevantly spaced in case objects are close to the observer or minimally spaced if they are far away from him/her.

In Section 2 the reconstruction methodology is presented and in Section 3 reconstruction examples are depicted. In Section 4 a few concluding considerations are drawn and possible future developments are briefly hinted at.

## 2 Methodology

With the aim of “translating” a two dimensional, single-point perspective, pictorial artwork into a tactile bas-relief, a methodology based on 7 phases has been developed:

1. Segmentation and classification of the identified elements;
2. Vanishing point identification;
3. Plane classification;
4. Characterisation of the *main plane* and greyscale gradient;
5. “Touching planes” and hierarchical sorting;
6. Characterisation of “non-touching planes”;
7. Greyscale image to 3D model transformation.

The algorithms used in the proposed methodology have been implemented in the MATLAB<sup>®</sup> programming environment according to an Object Oriented Programming (OOP) paradigm.

### 2.1 Segmentation and Categorisation of Identified Elements

#### 2.1.1 Segmentation Procedure

The first task consists of analysing the acquired image (see Fig. 1 (a)) in order to identify the different objects represented in the scene, such as human figures, architectural elements etc. Note that, since the final goal of the proposed methodology is to obtain a three-dimensional model made of planar regions, when identifying the different objects it is necessary to take into account



Fig. 1: (a) Acquired image of “*The Healing of the Cripple and the Raising of Tabitha*” fresco by Masolino da Panicale in the Brancacci Chapel (Church of Santa Maria del Carmine in Florence, Italy). (b) Segmented image (different colour represents different segment/region)

the type of plane which will be later used to represent the object itself.

The procedure of identifying scene objects, called segmentation, consists of partitioning the digital image into multiple regions. Over the years, a huge number of segmentation techniques have been proposed [23, 24]; since segmentation approaches are not the main focus of this work (but rather a tool necessary to image pre-processing), it is performed by means of the developed GUI, where the objects outlines are detected using the interactive livewire boundary extraction algorithm [25]. A unique numeric label is then assigned to each area delimited by a closed boundary. In case the region is not simply-connected (i.e. it features holes) each hole represents a separate object with a distinct label.

The matrix  $\mathbf{C}$ , size  $n \times m$ , resulting from the interactive algorithm consists of a set of regions whose values represent the region label. Every region is, therefore, represented by a set of pixels as shown in Fig. 1 (b).

The regions are separated from matrix  $\mathbf{C}$  by transforming them into a set  $\mathbf{I}$  of images  $\mathbf{I}_i$  where each image retains the same  $n \times m$  dimensions and it is expressed as follows:

$$\begin{cases} \mathbf{I}_i = 1 & \forall (n, m) | \mathbf{C} = 1 \\ \mathbf{I}_i = 0 & otherwise \end{cases} \quad (1)$$

where  $i = [1, 2, \dots, k]$  with  $k = \text{maximum number of identified regions}$ .

### 2.1.2 Plane Typologies Definition

As stated above, each raster image  $\mathbf{I}_i$  describes an image object; in the present work, it has been chosen to model each object as a plane by identifying 4 types of planes: *frontal planes*, *horizontal planes*, *vertical planes* and *oblique planes*. For the classification of these types of planes it is necessary to define an appropriate Reference Coordinate System (RCS) in the virtual 3D scene depicted in the analysed image. The RCS is defined as follows:

- $x$  axis, lying on the image plane and parallel to the horizon (pointing rightwards);
- $y$  axis, lying on the image plane and perpendicular to the horizon (pointing upwards);
- $z$  axis perpendicular to the image plane (according to the right hand rule).

The RCS origin is taken in the bottom left corner of the image plane. According to defined RCS, the above mentioned types of planes are defined as follows:

- *frontal planes* are the ones parallel to the image plane, whose geometry is not modified by the perspective view;
- *horizontal planes* are the ones perpendicular to the image plane and whose normal is parallel to the  $y$  axis;
- *vertical planes* are the ones perpendicular to the image plane and whose normal is parallel to the  $x$  axis;
- *oblique planes* are all the remaining planes, not belonging to the previous three categories. This last type of planes is not taken into account in the present method; further investigation shall be pursued in future studies.

In a single-point perspective view, the geometry of the *horizontal*, *vertical* and *oblique planes* changes (i.e. a *vertical plane* of rectangular form becomes trapezoidal). A visual representation of these types of planes is shown in Fig. 2.

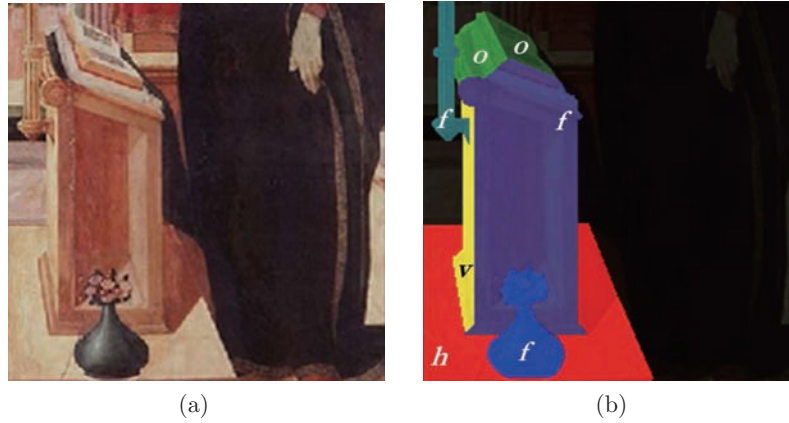


Fig. 2: (a) Detail of “Annunciation” painted by Filippino Lippi (Galleria dell’Accademia museum, Florence, Italy). (b) Visual representation of *frontal planes* (f), *horizontal planes* (h), *vertical planes* (v) and *oblique planes* (o)

## 2.2 Vanishing Point Identification

In this phase, the vanishing point coordinates on the image plane ( $xy$ ) are computed. For this purpose, with reference to the virtual 3D scene, the simplest way consists of identifying a pair of parallel lines, actually converging in the perspective view and receding to the horizon towards the vanishing point  $V$ . Once the line equations are expressed in homogeneous coordinates [26] it is possible to straightforwardly obtain the  $V$  coordinates ( $x_V, y_V$ ). The horizontal line through  $V$  in the picture plane is the horizon  $l_h$ . Let’s, then, the vertical line through  $V$  call  $l_v$ .

## 2.3 Plane Classification

As explained in Section 2.1.2, three types of planes are defined. In this section, the procedure to classify each region contained in images  $\mathbf{I}_i$  will be described. The first step to be performed is entirely manual and consists of selecting the regions representing *frontal planes*. Afterwards, since the  $V$  coordinates have been computed, it is possible to define an automatic procedure to classify at least a subset of the remaining *vertical* and *frontal planes*. For each image  $\mathbf{I}_i$ , the intersections

between the region contained in  $\mathbf{I}_i$ , and the horizon line are sought. If at least one intersection is found, the plane type is necessarily *vertical* and, in particular, is *vertical left* if placed to the left of  $V$  and *vertical right* if placed to the right of  $V$ .

This statement can be justified by observing that, at this stage of the procedure, only *horizontal* and *vertical planes* remain and that no *horizontal plane* can cross the horizon line (they are entirely above or below it). Actually the only *horizontal plane* (i.e. parallel to the ground) which may “intersect” the horizon line is the one passing through the view point (i.e. at the eye level) and whose representation degenerates in the horizon line itself.

If no intersections with the horizon line are found, intersections between the plane and the line passing through the vanishing point and perpendicular to the horizon are sought. If an intersection of this kind is found the plane type is necessarily *horizontal* (*upper horizontal* if above  $V$ , *lower horizontal* if below  $V$ ). This statement can be justified analogously to the previous one. If no intersections with either the horizontal line or with its perpendicular are found, the automatic classification of the plane is not possible; as a consequence, it is requested to manually specify the type of the plane via direct user input. The possible scenarios that may occur during plane classification procedure are shown in Fig. 3.

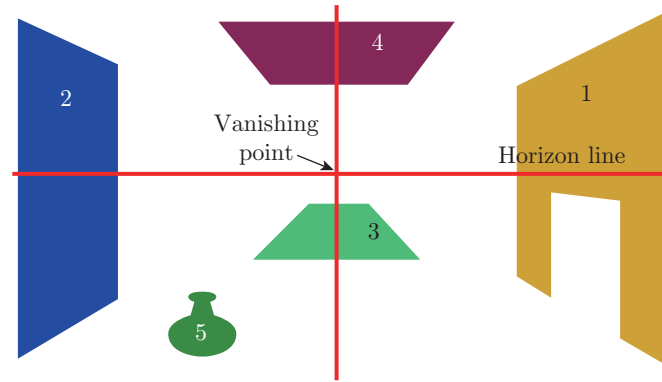


Fig. 3: Possible plane classification scenarios: plane type is automatically assigned to planes 1, 2, 3, 4; plane 5 has to be manually characterised via direct user input

## 2.4 Greyscale Gradient and Characterisation of the *Main Plane*

Among the identified *horizontal planes* it is necessary to define the *main plane* corresponding to the ground (or floor) of the virtual 3D scene; this plane will be used to obtain the  $z$  coordinate (height) of the planes directly “touching” it (e.g. trees, building walls, human figures).

As mentioned in Section 1, the surface of the final 3D model is obtained in the form of a height map image. According to this kind of representation, the  $z$  coordinate is bijectively represented by a grey value: a black value represents the background ( $z = 0$ ), whereas a white value represents the foreground, i.e. the virtual scene element nearest to the observer ( $z = 1$ ).

A greyscale gradient may be represented by a linear graduated ramp extending between two grey levels. Each different grey value will represent a different height  $z$ , as stated before. The gradient slope  $\mathbf{S}_{grad}$  is expressed as follows:

$$\mathbf{S}_{grad} = \frac{\mathbf{G}_0}{|p_0 - p_{vanishing}|} \quad (2)$$

where  $\mathbf{G}_0$  is the starting grey value,  $p_0$  is the gradient starting point coordinate ( $p \equiv x$  for *vertical planes*,  $p \equiv y$  for *horizontal planes*) and  $p_{vanishing}$  is the vanishing point coordinate ( $p \equiv x$  for *vertical planes*,  $p \equiv y$  for *horizontal planes*).

Referring to a digital image,  $p_0$  and  $p_{vanishing}$  coordinates are expressed in pixels (integer numbers) while the starting grey value  $\mathbf{G}_0$  is represented with a double-precision floating point value varying in the range  $[0 - 1]$ .

In order to assign the correct height to the pixels belonging to the *main plane*, a greyscale gradient can therefore be applied on the entire plane, from the foreground to the background. In this approach, the *main plane* (ground) lies below the horizon line and extends throughout the entire scene, as a consequence the foreground is white ( $\mathbf{G}_0 = 1$ ) and corresponds to the lower pixels, while the background is black ( $\mathbf{G}_0 = 0$ ) and corresponds to the pixels lying on the horizon line (see Fig. 4).

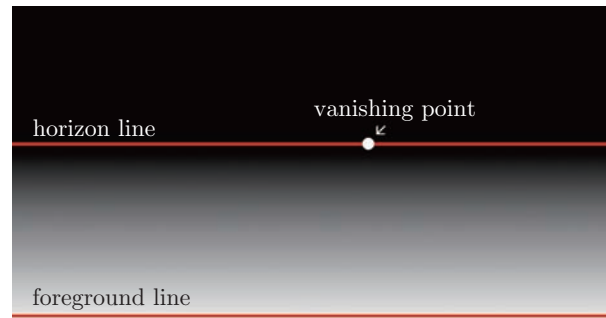


Fig. 4: Representation of the gradiented general *main plane*, extending between the foreground and the horizon (background) lines

Obviously, when *horizontal planes* other than the main one are considered, this statement is not valid. In such cases it is necessary to determine a suitable  $\mathbf{G}_0$  value in order to apply the greyscale gradient. In this case some additional information is needed; in particular it proves to be useful to analyse geometrical relationships among the planes as described in the next section.

## 2.5 “Touching Planes” and Hierarchical Sorting

In the 3D virtual scene, some planes (which, from now on, are called “touching planes”) are recognizable as physically in contact to other ones. These planes share one or more visible contact points: some examples may include a human figure standing on the floor, a picture on a wall, etc. With reference to a couple of touching planes, the first one is called *master plane* while the second one is called *slave plane*. The difference between *master* and *slave planes* is related to the hierarchical sorting procedure described below. Since contact points between two touching planes should share the same grey value (i.e. same height), considering the *main plane* and its touching ones (*slaves*) it is possible to obtain the grey value of a contact point, belonging to a *slave plane*, directly from the main one (*master*). This can be done once the *main plane* has already been gradiented. From the inherited grey value  $\mathbf{G}_{contact}$ , the starting grey value  $\mathbf{G}_0$  for the *slave plane* is obtained as follows, in the case of *frontal* or *vertical/horizontal planes* respectively:

a. *frontal planes*

$$\mathbf{G}_0 = \mathbf{G}_{contact} \quad (3)$$



b. *vertical and horizontal planes*

$$\Delta_{grad} = \frac{\mathbf{G}_{contact}}{|p_{contact} - p_{vanishing}|} \quad (4)$$

$$\mathbf{G}_0 = \mathbf{G}_{contact} + (|p_{contact} - p_0| \cdot \Delta_{grad}) \quad (5)$$

where  $p_{contact}$  is the contact point coordinate ( $p \equiv x$  for *vertical planes*,  $p \equiv y$  for *horizontal planes*). In light of that, the greyscale gradient has to be applied firstly to the *main plane* so that it is possible to determine the  $\mathbf{G}_0$  value for its *slave planes*. In fact, a totally similar procedure can be used to deal with planes touching other (non-main) ones. In other words, once plane slaves to the main one have been gradiented, it is possible to find the  $\mathbf{G}_0$  values for their own *slave planes* and so forth. More in detail, assuming the *master-slave* relationships are available, a hierarchical sorting of the touching planes can be performed. This is possible by observing that the relations between each touching plane can be unambiguously represented by a *non-ordered rooted tree*, in which the *main plane* is the tree root node while the other touching planes are the tree leaf nodes. By computing the depth of each leaf (i.e. distance from the root) it is possible to sort the planes according to their depth.

The coordinates of the contact point  $p_{contact}$  for a pair of *touching planes* are obtained in different ways depending on their type.

**Frontal planes touching the main plane:** for the generic  $i^{th}$  *frontal plane* touching the *main plane*, the contact point can be approximated, most of the times, by the lowest pixel  $h_b$  of the region represented in the  $\mathbf{I}_i$  image. Accordingly, the *main plane* pixel in contact with the considered region, is the one below  $h_b$  so that the entire region inherits its grey value. In some special cases, for instance when a scene object is extended below the ground (e.g. a well), the contact point between the two planes may not be the lowest pixel of  $\mathbf{I}_i$ . In this case it is needed to specify one contact point via user input.

**Vertical planes touching the main plane:** for the generic  $i^{th}$  *vertical plane* (preliminarily classified) touching the *main plane* it is necessary to determine the region bounding trapezoid. This geometrical construction allows to assign the correct starting grey value ( $\mathbf{G}_0$ ), for the *vertical plane* gradient, even if the plane has a non-rectangular shape (trapezoidal in perspective view). This is a common situation, also shown in Fig. 5.

By observing Fig. 5 (b), it is clear that the leftmost pixel of the vertical planar region to be gradiented is not enough in order to identify the point at the same height on the *main plane* where to inherit  $\mathbf{G}_0$  value from. In order to identify such a point, it is actually necessary to compute the vertexes of the bounding box as follows.

The four lines  $l_M, l_m, l_{LL}, l_{RR}$  delimiting the bounding box (expressed in homogeneous coordinates) are determined by defining:

- $M$  as the intersection point between the cluster and the line passing through  $V$ , tangent to the cluster and having the maximum slope ( $\frac{y_p}{x_p}$ );
- $m$  as the intersection point between the cluster and the line passing through  $V$ , tangent to the cluster and having the minimum slope;
- $L$  as any of the leftmost pixels of the cluster;
- $\hat{L}$  as any point taken in the same column as  $L$  (same  $x$  coordinate, any  $y$  coordinate);
- $R$  as any of the rightmost pixels of the cluster;
- $\hat{R}$  as any point taken in the same column as  $R$  (same  $x$  coordinate, any  $y$  coordinate).

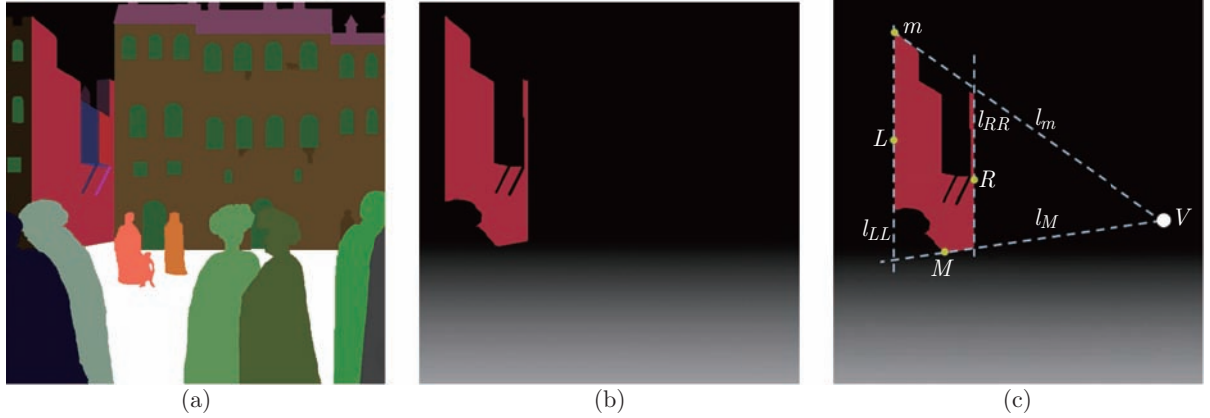


Fig. 5: (a) Detail of the segmented image of “*The Healing of the Cripple and the Raising of Tabitha*”. (b) Example of *vertical left plane* touching the *main plane*. (c) Example of bounding box for the same *vertical left plane*

so that:

$$l_M = M \times V \mid \frac{y_M}{x_M} = \max \left( \frac{y_p}{x_p} \right) \forall p \in \mathbf{I}_i \quad (6)$$

$$l_m = m \times V \mid \frac{y_m}{x_m} = \min \left( \frac{y_p}{x_p} \right) \forall p \in \mathbf{I}_i \quad (7)$$

$$l_{LL} = L \times \hat{L} ; \hat{L} = (x_L, \forall y, 1) \quad (8)$$

$$l_{RR} = R \times \hat{R} ; \hat{R} = (x_R, \forall y, 1) \quad (9)$$

An example of bounding trapezoid for a *vertical left plane* is shown in Fig. 5 (c) and the necessary point results from the intersection of edges  $l_M$  and  $l_{LL}$ .

**Other touching planes:** for every other type of *touching plane* whose *master plane* is not the *main plane* it is necessary to specify both the *master plane* and one of the contact points shared between the two. This task is performed via user input since there is no automatic method to univocally determine the contact point. The planes which are not visibly in contact, also called “non-touching planes”, are characterised according to the strategy provided in the following section.

## 2.6 Characterisation of “Non-touching Planes”

Some elements (i.e. planes) of the painting cannot be viewed as physically touching other elements. This is the case, for example, of birds, angels and cherubs suspended above the ground, who do not have any “physical” contact point with other elements (albeit having adjacent pixels). Other examples include planes whose contact point(s) is not visible in the scene because, for instance, it is hidden by a foreground element. For such planes it is not possible to find a  $z$  height unless the starting grey value is manually specified. The  $\mathbf{G}_0$  value is therefore obtained by choosing, on the gradiented image, any point having the desired height. When  $\mathbf{G}_0$  is assigned to a non-touching plane it is straightforward to apply the greyscale gradient to the plane, eventually obtaining the definitive greyscale image (height map)  $\mathbf{I}_{final}$ , shown in Fig. 6.



Fig. 6: Final greyscale heightmap of “*The Healing of the Cripple and the Raising of Tabitha*”

## 2.7 Greyscale Image to 3D Model

Once the final greyscale image is complete it is “translated” into a 2.5D point cloud which, in its turn, originates a fully three-dimensional solid model. The 2.5D point cloud is obtained by defining a set of  $(x, y, z)$  coordinates:  $x$  and  $y$  values are directly represented by the 2D image pixel coordinates;  $z$  values (height) are represented by the grey values, picked at  $\mathbf{I}_{final}(x, y)$  points. As previously stated, the background is represented by  $z = 0$ , whereas for the foreground, near the observer, by  $z = 1$ .

The resulting 2.5D point cloud, corresponding to the greyscale image depicted in Fig. 6, is shown in Fig. 7.

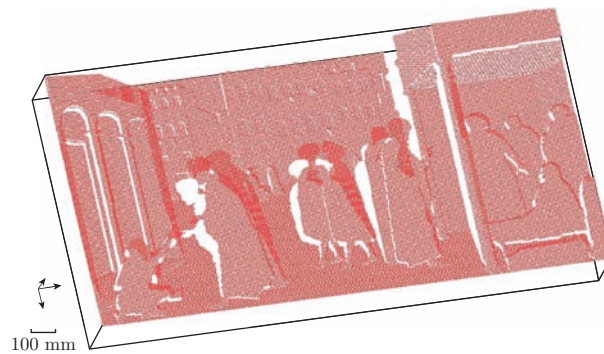


Fig. 7: Point cloud 2.5D representation of “*The Healing of the Cripple and the Raising of Tabitha*”

Based on the obtained 2.5D point cloud, it is possible to build a faceted surface model (triangular mesh). Numerous algorithms to tessellate point clouds are presented in literature. In this paper, the triangular mesh is obtained by performing the Delaunay triangulation-based method [27] proposed in [28].

In order to build the final bas-relief solid model, some post-production operations on the mesh are required. First of all it is necessary to scale the mesh so to obtain the desired dimensions, according to a set of general, which have been investigated by the authors in a previous work and which are not discussed here [9]. The primary recommendations to be taken into account, however, are:

- the height of relief needs to be at least 0.4 mm in order to be correctly perceived by the user (this requirement affects the final thickness of the bas-relief);
- the maximum width (or height) of the bas-relief needs to be such that the model can be

encompassed by the user's arms.

In the specific case study presented so far, an additional operation has been performed in order to further emphasize the foreground elements of the virtual scene and to prevent a disproportionate extension along the  $z$  direction.

The  $z$  dimension behind such foreground elements is shrunk, consequently creating a “*stacciato*”. The “*stacciato*” or “*schacciato*” is a sculpting technique, introduced and mastered for the first time by Donatello in the 15<sup>th</sup> century, in which the carved reliefs are shallow and limited in size. The use of this technique allows to represent the background elements in a limited  $z$  dimension and it is particularly adequate in a perspective representation. Once the model dimensions are adjusted, side and backward walls are added to the mesh, thereby shaping the final 3D bas-relief model (Fig. 8 (a)). Such a digital bas-relief can be used as input for several kinds of Computer Aided Machinery (CAM) or Rapid Prototyping (RP) systems (an example of a rapid prototyped model is shown in Fig. 8 (b)).



Fig. 8: (a) Three-dimensional bas-relief model (meshed surface) obtained by a Delaunay triangulation. (b) Final rapid prototyped model (UV cured resin) obtained with PolyJet™ technology (size 240 mm × 460 mm). Images are relative to the same case study “*The Healing of the Cripple and the Raising of Tabitha*”

### 3 Case Studies

The reconstruction methodology, implemented using MATLAB® programming language (with the exception of the post-processing operations on the mesh-see Section 2.7-which has been carried out with PolyWorks® V12), has been applied on two Italian Renaissance masterpieces:

1. “*The Healing of the Cripple and the Raising of Tabitha*” fresco painted by Masolino da Panicale (Brancacci Chapel in Church of Santa Maria del Carmine, Florence, Italy).
2. “*Annunciation*” painted by Filippino Lippi (Galleria dell’ Accademia museum, Florence, Italy).

The first of these case studies has been extensively illustrated in the previous sections and the final outcomes are shown in Fig. 8 (a) and Fig. 8 (b). With regards to the second case study, the painting is shown in Fig. 9 (a) and the segmented image in Fig. 9 (b).



Fig. 9: (a) “*Annunciation*” painted by Filippino Lippi (Galleria dell’ Accademia museum, Florence, Italy). (b) Segmented image

It needs to be pointed out that *oblique planes*, such as the ones characterizing the bookstand, are approximated with *horizontal* or *vertical* ones, since a proper greyscale gradient cannot be correctly applied. The greyscale image and the final 3D model are shown in Fig. 10 (a) and Fig. 10 (b).

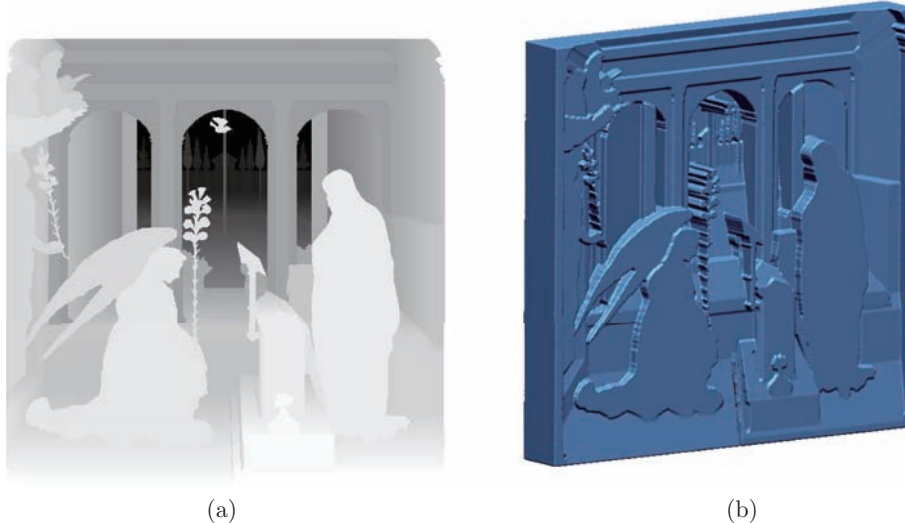


Fig. 10: (a) Final greyscale heightmap. (b) Three-dimensional bas-relief model (meshed surface) obtained by a Delaunay triangulation. Images are relative to the second case

## 4 Conclusions and Future Work

A new methodology for the semi-automatic generation of tactile 3D models starting from paintings characterized by single-point perspective has been presented. The proposed method, tested on two Italian Renaissance masterpieces, proved, by preliminary tests, to be robust and to require only little interaction from the user. The resulting 3D representation is meant to be a preliminary

one to be enriched by surface information, obtained by other suitable techniques, such as Shape-from-shading (SFS) [29], to be “added” on the planar regions identified by means of the above described procedure. Though the plane-wise representation is not the one to be ultimately sought, it has been shown in preliminary works by authors [30] that it is quite effective in enabling blind people to perceive spatial structure and scene elements. The proposed approach let us glimpse the possibility to simplify the task of translating 2D images into bas-reliefs, making this task accessible to a vast number of users and to a substantial decrease of costs. This will permit to increase the number of “translatable” artworks which, to date, have to be translated into bas-reliefs by hand sculpting. In addition to the above mentioned surface information obtainable from shading analysis, a straightforward improvement to this method is the forthcoming inclusion of two point perspective images, which leads to a broadening of the artwork range to be processed using the proposed method. Other future enhancements will encompass the automatic 3D characterization of *oblique planes* as well as geometrical primitives like cylinders, spheres and cones.

## Acknowledgement

The authors wish to acknowledge the valuable contribution of Prof. Antonio Quatraro, President of the Italian Union of Blind People (UIC) Florence (Italy), in helping the authors in the selection of artworks and in assessing the final outcome. The authors also wish to thank the Tuscany Region (Italy) for co-funding the T-VedO project (PAR-FAS 2007-2013), which originated and made possible this research, the *Fondo Edifici di Culto*-FEC of the Italian Ministry of Interior (Florence, Italy) and the Carmelite Community.

## References

- [1] F. Candlin, Blindness, art and exclusion in museums and galleries, *International Journal of Art and Design Education*, 22(1), 2003, 100-110
- [2] K. Hetherington, Museums and the visually impaired: The spatial politics of access, *The Sociological Review*, 48(3), 2000, 444-463
- [3] M. Pieraccini, G. Guidi, C. Atzeni, 3D digitizing of cultural heritage, *Journal of Cultural Heritage*, 2(1), 2001, 63-70
- [4] G. Guidi, J. A. Beraldin, C. Atzeni, High-accuracy 3D modeling of cultural heritage: The digitizing of Donatello's, *IEEE Transactions on Image Processing*, 13(3), 2004, 370-380
- [5] S. Barone, A. Paoli, A. V. Rationale, 3D reconstruction and restoration monitoring of sculptural artworks by a multi-sensor framework, *Sensors*, 12(12), 2012, 16785-16801
- [6] L. Iuliano, P. Minetola, Rapid manufacturing of sculptures replicas: A comparison between 3D optical scanners, in: *CIPA 2005 XX International Symposium*, 2005, 384-389
- [7] P. K. Edman, *Tactile Graphics*, AFB Press, 1992
- [8] L. Thompson, E. Chronicle, Beyond visual conventions: Rethinking the design of tactile diagrams, *British Journal of Visual Impairment*, 24(2), 2006, 76-82
- [9] M. Carfagni, R. Furferi, L. Governi, Y. Volpe, G. Tennirelli, Tactile representation of paintings: An early assessment of possible computer based strategies, *Progress in Cultural Heritage Preservation, Lecture Notes in Computer Science*, 9 (2012), 261-270



- [10] S. Oouchi, K. Yamazawa, L. Secchi, Reproduction of tactile paintings for visual impairments utilized three-dimensional modeling system and the effect of difference in the painting size on tactile perception, in: K. Miesenberger, J. Klaus, W. Zagler, A. Karshmer (Eds.), *Computers Helping People with Special Needs*, Springer Berlin Heidelberg, 2010, 527-533
- [11] T. Weyrich, J. Deng, C. Barnes, S. Rusinkiewicz, A. Finkelstein, Digital bas-relief from 3D scenes, *ACM Transactions on Graphics (TOG)*, 26(3), 2007, 32
- [12] P. Cignoni, C. Montani, R. Scopigno, Computer-assisted generation of bas-and high-reliefs, *Journal of Graphics Tools*, 2(3), 1997, 15-28
- [13] P. Sturm, S. Maybank, A method for interactive 3D reconstruction of piecewise planar objects from single images, in: *10th British Machine Vision Conference (BMVC'99)*, 1999, 265-274
- [14] A. Criminisi, I. Reid, A. Zisserman, Single view metrology, *International Journal of Computer Vision*, 40(2), 2000, 123-148
- [15] D. Hoiem, A. A. Efros, M. Hebert, Automatic photo pop-up, *ACM Transactions on Graphics (TOG)*, 2005, 577-584
- [16] T. Hassner, R. Basri, Example based 3D reconstruction from single 2D images, in: *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, 2006, 15-15
- [17] E. Delage, H. Lee, A. Y. Ng, Automatic Single-image 3D Reconstructions of Indoor Manhattan World Scenes, *Robotics Research*, Springer Verlag, 2007
- [18] R. Furferi, L. Governi, M. Palai, Y. Volpe, 3D model retrieval from mechanical drawings analysis, *International Journal of Mechanics*, 5(2), 2011, 91-99
- [19] L. Governi, R. Furferi, M. Palai, Y. Volpe, 3D geometry reconstruction from orthographic views: A method based on 3D image processing and data fitting, *Computers in Industry*, 64(9), 2013, 1290-1300
- [20] Z. K. Huang, X. W. Zhang, W. Z. Zhang, L. Y. Hou, A new embossing method for gray images using Kalman filter, *Applied Mechanics and Materials*, 39 (2011), 488-491
- [21] Z. Li, S. Wang, J. Yu, K. L. Ma, Restoration of brick and stone relief from single rubbing images, *IEEE Transactions on Visualization and Computer Graphics*, 18(2), 2012, 177-187
- [22] A. Reichinger, S. Maierhofer, W. Purgathofer, High-quality tactile paintings, *Journal on Computing and Cultural Heritage*, 4(2), 2011, Article No. 5
- [23] D. Ziou, S. Tabbone, Edge detection techniques - An overview, *International Journal of Pattern Recognition and Image Analysis*, 8 (1998), 537-559
- [24] E. Nadernejad, S. Sharifzadeh, H. Hassanpour, Edge detection techniques: Evaluations and comparisons, *Applied Mathematical Sciences*, 2(31), 2008, 1507-1520
- [25] W. A. Barrett, E. N. Mortensen, Interactive live-wire boundary extraction, *Medical Image Analysis*, 1(4), 1997, 331-341
- [26] R. F. Riesenfeld, Homogeneous coordinates and projective planes in computer graphics, *IEEE Comput. Graph. Appl.*, 1(1), 1981, 50-55
- [27] L. Di Angelo, P. Di Stefano, L. Giaccari, A new mesh-growing algorithm for fast surface reconstruction, *Comput. Aided Design*, 43(6), 2011, 639-650
- [28] A. Bowyer, Computing Dirichlet tessellations, *The Computer Journal*, 24(2), 1981, 162-166
- [29] R. Zhang, P. S. Tsai, J. E. Cryer, M. Shah, Shape-from-shading: A survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8), 1999, 690-706
- [30] Y. Volpe, R. Furferi, L. Governi, G. Tennirelli, Computer-based methodologies for semi-automatic 3D model generation from paintings, *Int. J. Computer Aided Engineering and Technology*, 6(1), 2014, 88-112