amcs

# A COMPARISON OF HOLE–FILLING METHODS IN 3D

Emiliano Pérez [a,*], Santiago Salamanca [a], Pilar Merchán [a], Antonio Adán [b]

[a] Industrial Engineering School
University of Extremadura, Avda. Elvas, 06006 Badajoz, Spain
e-mail: {emilianoph,ssalamanca,pmerchan}@unex.es

[b] Computer Science School
University of Castilla–La Mancha, Paseo de la Universidad 4, 13071 Ciudad Real, Spain
e-mail: antonio.adan@uclm.es

This paper presents a review of the most relevant current techniques that deal with hole-filling in 3D models. Contrary to earlier reports, which approach mesh repairing in a sparse and global manner, the objective of this review is twofold. First, a specific and comprehensive review of hole-filling techniques (as a relevant part in the field of mesh repairing) is carried out. We present a brief summary of each technique with attention paid to its algorithmic essence, main contributions and limitations. Second, a solid comparison between 34 methods is established. To do this, we define 19 possible meaningful features and properties that can be found in a generic hole-filling process. Then, we use these features to assess the virtues and deficiencies of the method and to build comparative tables. The purpose of this review is to make a comparative hole-filling state-of-the-art available to researchers, showing pros and cons in a common framework.

**Keywords:** survey, 3D polygonal models, repairing meshes, hole-filling, restoration algorithms.

## 1. Introduction

Recently, there has been a growing interest in the polygonal representation of three dimensional objects in many applications and, more particularly, in triangular meshes. One of the main reasons is that graphic hardware is currently highly specialized in the representation and processing of polygons. Also, RGB-D sensors provide the key for the resolution of real time applications using 3D data (Wilkowski *et al.*, 2016). Polygonal meshes provide good visualization, flexibility and simplicity in computer vision and computer graphics applications. Additionally, they greatly facilitate the design, processing, transmission, animation and interaction of 3D models in virtual scenes.

Polygonal models can be obtained from direct or reverse engineering processes. Direct engineering techniques use free-shape and CAD modelers to create virtual models, whereas reverse engineering methods build models from data provided by 3D sensors. In both cases different types of mesh defects arise. Thus, in designed (or virtual) models we can usually find gaps,

self-intersections, degenerated elements and singularities in the geometry. Nevertheless, the most common defects in digitized models are noise, holes, topology noise and aliasing. Of these, holes and gaps are perhaps the most important flaws, and have been studied in depth in recent years. Typically, gaps and holes have different meanings. A gap is defined as an empty region between two set of triangulated surface patches, the boundary of the gap being two disconnected chains of edges. In contrast, a hole is a missing area within a triangulated mesh, and the boundary is usually one or several edge loops. It is important to point out that this paper focuses only on hole-repairing.

Holes are well known defects in meshes generated from digitization tasks, for example, when using laser scanners. In this case, some parts of the object can be occluded by others and the scanner is unable to reach certain regions. These missing data generate empty areas in the mesh.

In general, when a polygonal mesh is created, geometrical and topological requirements are imposed. The former guarantee that polygons precisely represent the outer surface of the object. Therefore, the polygonal

---

*Corresponding author

surface is required to be closed, intersections-free and without singular points. Meeting these requirements is particularly important in engineering and manufacturing, where solid objects are required for numerical computations (finite element analysis) and for production tasks (rapid prototyping). Moreover, topological demands require that all pieces of the polygonal model follow connectivity rules and that errors, such as redundant handles and disconnected patches, be avoided. Such errors introduce extra complexity to the model and make further parametrization and segmentation tasks much more complicated. In particular, topological features, such as handles and connected components, have to be preserved.

In this paper, we discuss how to repair a polygonal mesh in which some holes have been detected. So far a multitude of hole-filling algorithms have been extensively tested but as yet there is no general solution for all situations and cases. Despite this, we can establish a set of general criteria that a robust hole-filling algorithm should satisfy (Podolak and Rusinkiewicz, 2005):

1. It should produce a watertight mesh without a self-intersection mesh.

2. It should be able to process holes of arbitrary shapes and sizes.

3. It should avoid changes, approximations or resampling of the original data. It is essential that the starting data remain unchanged throughout the process of reconstruction, as they are real data physically measured by sensors.

4. It should be able to incorporate user constraints in order to allow the selection of multiple topologically distinct solutions.

5. It should be able to process large and high resolution scanned meshes.

6. It should be able to differentiate real digitized surfaces from those created by means of a filling process. This requirement is particularly important in applications like manufacturing, study and restitution of cultural heritage pieces from their digital models, etc.

**1.1. Motivation.** The purpose of this paper is twofold. First, we provide a review of the most relevant hole-filling works and highlight their importance, the context in which each method is applied and the results obtained. This review is structured according to a classification based on the type of 3D representation that the method uses. Secondly, we provide a comparative analysis and evaluate a set of parameters of all the described methods.

Other surveys of interest that deal with polygonal mesh repairing exist. The papers by Ju (2009) and Attene *et al.* (2013) are two of the most interesting works. In these, the authors review the general problem of errors in meshes in a wider sense. These articles are very useful to consult the variety of mesh repairing methods since they present an organized classification of the kinds of errors (with their corresponding definitions) and provide some available techniques. Specifically, Ju (2009) collects algorithms that fix geometric errors with a methodological perspective, and Attene *et al.* (2013) distinguish between algorithms that fix local connectivity flaws, global topology issues, geometric errors, or a combination of the aforesaid.

In contrast to these valuable works, we propose a different survey on the following points:

1. This article provides a structured and extended overview of the specific problem of hole repairing in polygonal meshes. Our review gives a detailed explanation of each particular method with the aim of providing essential information to other researchers in the same area. We present a brief summary containing the algorithmic essence, the main contributions and the limitations of each method.

2. We make a comparison between all the referenced methods taking into consideration multiple aspects. We enumerate the features to be compared and generate a large comparison-table. Additionally, despite the risk in doing so, we make an assessment of the quality of each technique with respect to each particular feature. This comparison gives an idea of how versatile and suitable each solution is and which technique is appropriate to be applied under a particular set of conditions and applications. This kind of comparison has never been presented in the area of mesh repairing.

The survey was organized taking into account a classification which is not based on the input model, but on the intermediate representation of this model used by the algorithm. We distinguish between methods based on the polygonal representation (Section 2), on parametric representations (Section 3) and on volumetric representations (Section 4). We also include other nonclassical methods in Section 5. Section 6 is devoted to the comparison of all previous methods. Conclusions are presented in the last section.

## 2. Methods based on the polygonal representation

Methods based on the polygonal representation are usually local approaches, in the sense that the mesh is

manipulated only in the vicinity of the hole, whereas the rest of the mesh remains unaltered. This group encompasses a large part of algorithms devoted to hole-filling. Here, the search and detection of the hole's boundaries is one of the most important stages in the whole process. To begin with, hole identification can be carried out by seeking the triangles that have at least one non-shared edge. Then, the edges are joined in order to obtain cycles, and each cycle corresponds to the boundary of the hole.

Barequet and Sharir (1995) proposed a fundamental work on the treatment of holes and gaps with non-trivial boundaries. Their method uses a geometric hashing technique to identify and bridge boundary parts that have a similar shape. A partial curve matching technique, adapted from computer vision, is firstly used to identify matching border portions. Then, a consistent set of matched candidates is chosen and stitched together. Finally, the remaining holes are identified. The method closely follows the dynamic programming technique of Klincsek (1980).

Another important work to be highlighted is the one by Liepa (2003). This approach is influenced by Barequet and Sharir's method with respect to the triangulation process. Here, the author describes a method to fill holes in unstructured triangular meshes by interpolating the shape and density of the surrounding mesh. To perform triangulation, the proposal uses a weighted function that takes into account the area and angle of the triangles. This method can deal with arbitrary holes in arbitrary meshes and holes with islands. *Meshlab*, the well-known software in the 3D community, includes a filling hole section which is based on a variation of the Liepa (2003) method. An example of the results obtained by using *Meshlab* can be observed in Fig. 1. The mesh shown in this figure will be used several times throughout the paper to run available demos.

Other early methods tackle the problem directly (brute-force based methods). For example, the hole-filling process proposed by Wei *et al.* (2010) is carried out in three steps. The first one consists of a hole triangulation by means of a function that optimizes the triangulation angle. Then, a subdivision is iteratively applied to match the size of the hole's faces with the ones in the hole's surroundings. Finally, a Laplacian filter is applied to smooth the surface. The ability of the method to deal with large holes is remarkable.

The approach proposed by Brunton *et al.* (2009) also performs a simple hole triangulation. In this case, the mesh is firstly folded, and then a filling process in the 2D polygonal mesh where the hole lies is carried out. Vertices which are close to the hole's boundary are taken to apply the unfolding process. In order to avoid self-intersections during this process, the movement of the vertices is limited. After unfolding the mesh, the
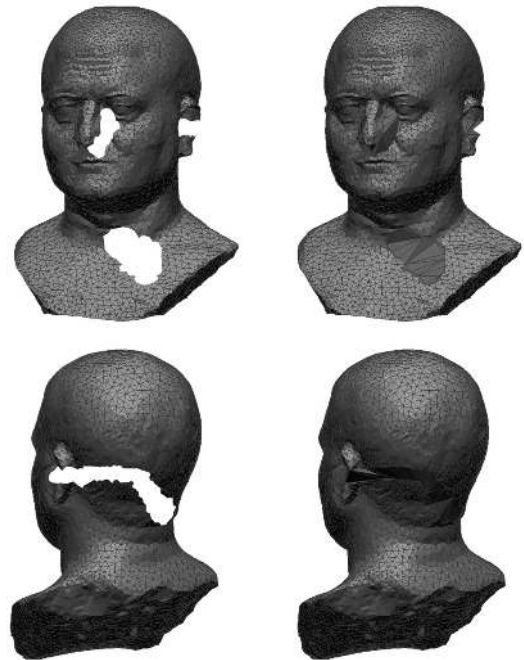


Fig. 1. Hole-filling example using the *Meshlab* software. Based on a variation of the method presented by Liepa (2003).
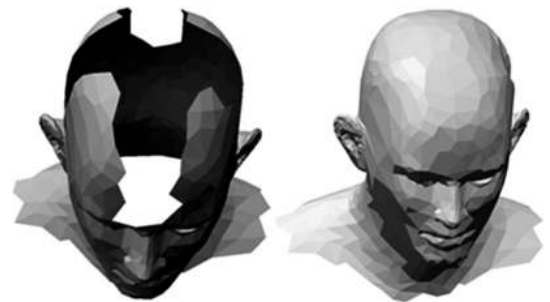


Fig. 2. Example of hole-filling by unfolding the mesh. The method presented by Brunton *et al.* (2009).

hole's boundary becomes a simple flat polygon which is triangulated using a limited Delaunay triangulation algorithm. This triangulated patch is finally embedded in the 3D model, and a refinement stage is performed. The interior vertices of the patch are positioned so that they approximate a minimum energy surface (MES). Two energy functions are used: a discrete equitable function of energy (Kobbelt *et al.*, 1998) and a Laplacian energy function (Sorkine and Cohen-Or, 2004). Figure 2 shows an example of this technique. The unfolding process clearly limits the applicability of this method.

Far from the traditional 3D triangulation techniques, there are other new ideas, such as the frontal advance technique of Zhao *et al.* (2007). The frontal advance approach generates an initial closure of the hole in the mesh. Afterwards, triangles forming the initial

Fig. 3. Complex hole-filling using the normal estimation technique. The method presented by Zhao *et al.* (2007).

patch are modified by estimating the appropriate normal vectors. Finally, the three coordinates of each new vertex are repositioned by the resolution of Poisson equations (which are based on the appropriated normals and the hole's boundary). Figure 3 illustrates some results of this method in complex holes.

Predictive methods are also incorporated as new strategies to be considered within the hole-filling world. This is the case of Wang and Hung (2012), who consider two stages: surface hole-filling and system grey prediction adjustment. In the filling stage, the method detects and rebuilds the holes' boundaries in the mesh, so that they are converted into simple convex polygonal holes. Then, the boundary points are projected onto a plane and the hole is filled by means of a Delaunay procedure (Fang and Piegl, 1995). The prediction model based on existing data forecasting methods is used to identify the future dynamic situation of each element within a given series. Here the normal vector prediction and the angle prediction are considered. This method yields good results for not too big and narrow holes. A particular case of this method is the one by Wang and Oliveira (2007). The method has a point cloud as input and generates an intermediate representation consisting of a triangular mesh on which an early hole boundary detection is applied. After that, an MLS (moving least squares) interpolation technique (Lancaster and Salkauskas, 1981) is applied on a set of boundary neighbor points.

Most of these local approaches follow the same stages: hole detection, rough filling and refinement. The results are usually good in smooth free forms, but these methods are unable to retrieve complex missing shapes with linear features. In the case of the work presented by Pernot *et al.* (2006), the contribution is in the refinement stage. The objective here is to minimize the curvature variation between the surrounding and the inserted mesh. To overcome it, the proposed method allows the user to manually specify additional constraints.

Some polygonal-processing-based methods have the ability to preserve the characteristics of the surface in the neighborhood of the hole. There exist simple techniques which, after triangulating the hole, apply a bilateral noise removal filter, thus maintaining the surface characteristics in the surrounding of the hole (Hu *et al.*, 2012). This kind of method works properly on industrial pieces.

Other methods are devoted to recovering sharp areas and lineal structures, which are typical in CAD models. For example, Wang *et al.* (2012) present a method to fill holes in meshes with abrupt changes. To recover the missing parts, three main steps are carried out. First, vertices around the hole are extracted and classified into different feature sets. Then, an algorithm matches these sets to construct the missing feature curves and divides the original hole into several simple sub-holes. Finally, the sub-holes are filled by the modified advancing front method (Wang *et al.*, 2011). This technique does not work in free-form objects in which it is very difficult to find characteristic vertices. Figure 4 illustrates some results of this method.

A simple idea which also yields good results in concave and sharp areas can be found in the paper by Li *et al.* (2008). Nevertheless, it does not work with very big holes in free-form objects. Here, the approach incrementally splits a complex hole into simpler ones, respecting the 3D shape of its boundary and its neighborhood. Each resulting simple hole is then filled with planar triangulation methods. This division is based on the curvature associated with each vertex of the hole's boundary. Starting from the vertex with least curvature, vertices are added in an iterative process, and the resulting set is further adjusted to a second degree bivariate polynomial.

Semiautomatic approaches, such as that by Ngo and Lee (2013), let the user modify the position of crest points detected in the mesh. These salient surface characteristics help to find feature points in the holes and their vicinities. The system performs a feature line interpolation over the holes and divides large complex holes into smaller and more planar ones. For each of these simple holes, triangulation and 3D mapping follows.

Zhao *et al.* (2006) address the problem of reconstructing the salient regions. This method uses crest lines to build salient features and, following this, the topology of the hole and crest lines are rebuilt by means of triangulation and region growing algorithms, respectively.

## 3. Methods based on parametric representations

As mentioned, holes usually come from missing surface parts, which have been caused by a variety of sources. Therefore, it is advisable to fill them by inserting new polygons after detecting closed pools of boundary edges.
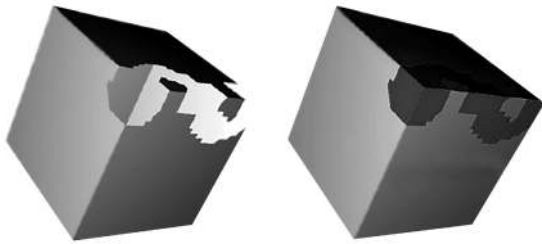
Fig. 4. Hole-filling method that follows lineal structures in sharp shapes. The method presented by Wang *et al.* (2012).

Nevertheless, other techniques can infer the missing geometry through surface parametric representations. Thus, although the polygonal representation is the most often used methods, there are other non-polygonal techniques which use implicit functions to represent surfaces. These techniques offer several advantages such as easy computation for interpolation and extrapolation of surfaces, easy management of large data sets and fast methods for fitting and evaluating. Another important advantage is that they generate mesh portions that smoothly fit into the original mesh.

Radial basis functions (RBFs) are used as a surface interpolation method in several techniques. Two good examples here are the ones by Branch *et al.* (2006) and Wu *et al.* (2008). In the work of Branch *et al.* (2006) the torque of the curve that defines the hole contour is analysed. The idea is that holes belonging to the surface are smooth and regular (with low torque), while those generated by occlusion exhibit irregularities and have high torques. Points in the surroundings of the hole's boundary are then used for fitting a surface through interpolators of radial basis functions (RBF). Afterwards, an iterative process makes the hole smaller until it reaches a preset threshold. Furthermore, the reconstructed surface maintains the resolution of the original mesh. A limitation of the method is that it does not work efficiently with large holes.

The second referenced method (Wu *et al.*, 2008) detects the hole and identifies the boundary's vertices and nodes with a degree of vicinity of 2 or 3. These vertices are used as interpolation centers to define a local implicit surface which, in turn, serves to interpolate (RBF) the content of the hole. The last step of the algorithm is the integration of the calculated mesh portion with the rest of the mesh. Since the mesh patch in which the points have to be interpolated is larger than the hole, it will be necessary to identify the interior points to the hole. To this end, a projection of the mesh patch to a plane obtained from a principal components analysis is performed. This entails a limitation of the method. The polygon corresponding to the hole's boundary is also projected onto the same plane. Finally, 2D to 3D transformation is applied and the hole is

filled in the initial mesh.

NURBS and Bezier surfaces are also mathematical models that are used to generate and represent interpolated surfaces in holes. The works of Kumar *et al.* (2007) and Li *et al.* (2010) are two representative examples in which this kind of solution is applied.

The method by Kumar *et al.* (2007) begins by closing the hole with a triangular mesh. This initial connectivity is kept by points until the end of the process. Then, for each hole, six rings around the hole's boundary are calculated and the set of splines which approach the rings are computed. These curves are used to obtain a set of eighteen NURBS surfaces which close the hole. The initial computed triangulation is projected on each NURBS and the final coordinates of the mesh are calculated as an average of all these projections. This algorithm can only be used on smooth surfaces.

Li *et al.* (2010) propose a method based on polynomial adjustment techniques using Bezier surfaces. First, the characteristics of the hole surroundings are detected and the types of curves that can fit these regions are established. Then, superficial curves that divide the hole into other simpler sub-holes are obtained. These surfaces are filled using hybrid Bezier-Lagrange patches. If the third order Bezier patch does not fit correctly, a further hole subdivision is then required and the procedure starts again.

It is worth mentioning that, in the previous methods, the holes do not explicitly appear in the resulting model because they are filled as the model is generated. For example, this occurs when a point cloud is approximated by NURBS. Therefore, the resulting model is created in advance without taking into account the holes, so that the filling process is done during the creation of the 3D model. That is, the holes, if they exist, are not dealt with.

In particular, in the point clouds merging stage, the union of all the partial views is treated as a set of unorganized 3D points which have to fit a continuous surface. Since there is no connectivity between the point clouds, the holes are conceptually equivalent to the space between the adjacent views, so that these methods fill holes during the reconstruction. Some of the existing methods interpolate original data using alpha shapes (e.g., Edelsbrunner and Mücke, 1994; Bajaj *et al.*, 1995), crusts (e.g., Amenta *et al.*, 1998; Dey *et al.*, 2001) or spheres (e.g., Bernardini *et al.*, 1999). However, the interpolation of data by continuous shapes may be sometimes unsuitable for noisy data and holes can then appear. In the work of Bernardini *et al.* (1999) this problem is solved by applying smoothing and merging processes between increasing spheres and taking care not to leave gaps between spheres in each iteration. RBF is also used directly over point clouds by Dinh and Turk (2001) as well as Carr *et al.* (2001). Here, the method performs a weighted sum of RBFs in order to obtain a

new global function, which finally generates the surface.

## 4. Methods based on signed distance functions and volumetric representations

In general, highly distorted or inconsistent meshes with multiple types of defects should be fixed through global approaches. Methods based on volumetric representations can be classified as global methods in the hole-filling field. The word "global" is used here in the sense that the approach is not applied in the vicinity of each particular hole. Commonly, volumetric models are employed in the synthesis, manipulation, and rendering of objects, and stored as a volume buffer of voxels. In hole restoration, volumetric solutions are typically based on a complete remeshing of the input or imply some intermediate data structure different from a polygonal mesh.

One of the earliest approaches can be found in the work of Murali and Funkhouser (1997). These authors propose an approach to construct consistent representations of the solid object modeled by an arbitrary set of polygons. To do so, the method follows three steps:

(a) spatial subdivision: the space is partitioned into a set of polyhedral cells and an adjacency graph is built (in which each node represents a convex polyhedron and each link represents a convex polygon);

(b) determination of solid regions: the algorithm computes whether each cell is solid or not, based on the properties of its links and neighbors;

(c) model generation: the output is a polygonal description of all links in the adjacency graph that represents the boundaries between cells that are solid and cells that are not solid, consistently orienting all polygons away from solid cells.

One of the best techniques in this line is the one presented by Davis *et al.* (2001). Here, an implicit distance function defined in the vicinity of the hole is calculated. This function is subjected to a diffusion process that extends the surface along the volume. This is an easy-to-implement method which generates surfaces that do not intersect with each other. It is efficient for large holes and high resolution meshes as depicted in Fig. 5. Another distance function based approach is proposed by Sagawa and Ikeuchi (2008). This method has a set of range images as input and initially classifies inside or outside voxels, depending on the input's normals. In order to merge all range images, the authors use a signed distance field which is taken as an intermediate representation. A surface is created to fill a hole by iteratively updating the distance field, while making an effort to maintain continuity with the starting range images. Since this method can be applied to a distance



Fig. 5. Hole-filling method which applies volumetric diffusion. The method presented by Davis *et al.* (2001).

field with an adaptive resolution, it is assumed to work efficiently in large holes and environments with high curvature. Both methods, by Davis *et al.* (2001) as well as Sagawa and Ikeuchi (2008), require oriented inputs.

Variants of the former method can be found in the works of Guo *et al.* (2006) and Caselles *et al.* (2008). As in the case of Davis *et al.* (2001), these methods also use volumetric data, so that the surface is represented as the zero level-set of a function $u$, and then minimize an energy functional which integrates a power of the mean curvature of the level sets of $u$. In contrast to the work of Davis *et al.* (2001), these approaches use a system of coupled anisotropic (geometric) partial differential equations which are applied only at the holes and their neighborhood. This permits the surface to be geometrically extended into the hole. Both methods work in large holes and always produce a smooth surface. Worth noting is their ability to preserve the surface's features.

The strategy followed by Podolak and Rusinkiewicz (2005) as well as Nooruddin and Turk (2003) changes the input mesh model representation for a volumetric model. In the first paper (Podolak and Rusinkiewicz, 2005), a decomposition of the space in atomic volumes is proposed. The process is divided into two steps. In the first one, a cube containing the input mesh is generated, and then it is partitioned in atomic volumes. A volume is atomic if it does not intersect with polygons of the mesh. From this formulation, each atomic volume will be either within the output mesh (inner volume) or outside thereof (external volume). Output model is defined as the union of the interior volumes, which ensures that the resultant object is watertight, without holes. In the second step, a smoothing process is applied to make the underlying atomic volumes structure less visible. Some results in very large holes can be seen in Fig. 6. Nooruddin and Turk (2003) convert a polygonal model into a volumetric representation. They handle models with holes, double walls and intersecting parts. One benefit of converting the input polygonal model into a volume is that these can easily repair a number of degeneracies. The resultant process does not allow filling holes in areas with high curvature.

Ju (2004) presents a method which also considers

Fig. 6. Hole-filling of a torus using 3D space decomposition in atomic volumes. The image on the far left shows the original torus that must be repaired. The other two images are two different solutions produced by the algorithm. The method presented by Podolak and Rusinkiewicz (2005).

the idea of inside/outside volumes, but using an octree grid. The proposal follows four stages: boundary cycle detection, boundary cycle patching, sign generation in octree nodes, and surface reconstruction. This technique produces a closed surface and divides the space into disjointed internal and external volumes. It is able to efficiently process large models containing millions of polygons as well as reproduce sharp features in the original geometry. The method by Bischoff *et al.* (2005) also uses the octree data structure but in a different manner. Of the six steps in their method, we will highlight the one concerning the representation model. The method defines an adaptive octree, in which each cell stores references to the triangles that it intersects with. Following this, a sequence of morphological operations to the cells is carried out to determine the topology of the restored surface. An extension of Ju's dual contouring algorithm (Ju *et al.*, 2002) guarantees that the restored surface has a proper manifold topology. This technique does not impose restrictions on the input mesh, resamples the original model and preserves all the important geometric features, such as sharp corners and edges. The author has developed a software called *PolyMender*, available on his webpage, which allows polygonal models to be repaired using this method. *Poly-Mender* is suitable for repairing CAD models and gigantic polygonal model. Alternatively, it can also be used to generate a signed volume from any polygonal models. Some results of this technique are shown in Fig. 7.

A recent method described by Kumar and Shih (2012) extends the algorithm of Kumar *et al.* (2007) (Section 3). Here a hybrid approach surface and volume based technique is proposed. First, the surface based technique of Kumar *et al.* (2007) is applied and then the resulting surface mesh is converted into a volumetric representation. For simplicity, the proposal uses a Cartesian grid to represent the data. Voxelization is only performed in non-intersected regions and near the surface defects. To complete the missing parts of voxelized geometry, a diffusion equation is applied. As a result, and following a marching cubes technique, a set of points (zero-set in the voxelization) is available with its pre-computed normals. Finally, the method uses a *Poisson surface reconstruction* (Kazhdan *et al.*, 2006) algorithm
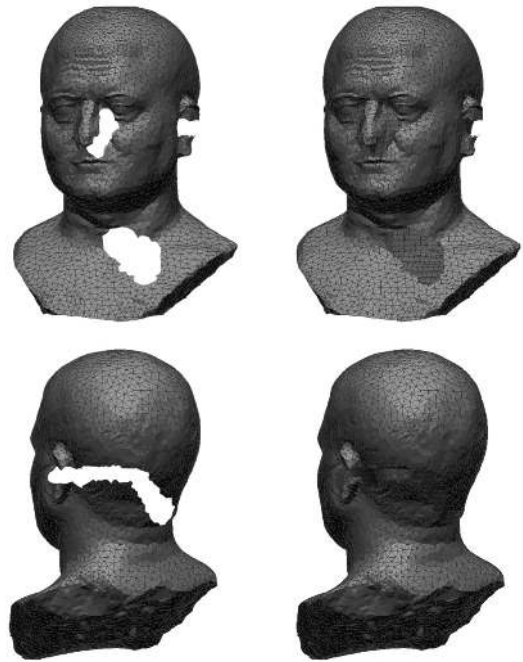


Fig. 7. Hole-filling example using the *PolyMender* software which generates a signed volume from the input mesh. The method presented by Ju (2004).

and obtains the final watertight mesh.

Contrary to the above methods, in the approach proposed by Curless and Levoy (1996) hole-filling is implicitly done during model creation. The approach processes separately each view of the scene and then carries out an integration phase in which the holes are filled without explicit delimitation. Each view is converted into a signed distance function whose zero level is the observed surface. The distance functions are then merged, the set of null values are extracted and the corresponding surface is defined. To fill the holes, the method firstly marks as empty the region of 3D space that lies along the line of sight between the scanner and the meshes. Afterwards, the boundary of this region is extracted and the filling process is performed by creating a surface which limits the maximum region of space that is consistent with the set of views, thus ensuring that the resultant mesh is watertight.

Finally, it is worth mentioning the work developed by Paulsen *et al.* (2010). This method creates a new oriented 3D data with consistent normal directions. A signed distance field based over the oriented point set is then computed by means of a second order energy minimization. This method is similar to the one by Jakobsen *et al.* (2007). In the next step, a Markov random field (MRF) based regularization method is applied to the distance field. First, the prior and observation models are formulated as an energy function that has to be minimized. In addition, multi-scale methods re-estimate distances in

between the regularization steps. Finally, the Bloomenthal polygonizer (Bloomenthal, 1994) is used to extract the isosurface from the MRF regularized distance field, and the resulting mesh is optimized by solving linear systems based on Laplacians (Botsch and Kobbelt, 2004). This method works very well in small/medium holes and is able to preserve simple and smooth features.

## 5. Other methods

**5.1. 2D image based methods.** 2D image based methods are those in which one or several images of the scene are used in any stage of the hole-filling process. These methods usually start by detecting and filling holes through typical polygonal approaches and, after that, the image is used to refine or correct earlier results. Some of these solutions are briefly explained in the following paragraphs.

The solution proposed by Brunton *et al.* (2007) starts with initial hole-filling by using a geometric interpolation technique. This interpolation performs a hole triangulation without adding interior points. To achieve a resolution similar to the rest of the mesh, the triangulation is refined by adding interior points (Delaunay's triangulation). At this point in the process, photographs of the object, taken during the scanning stage, are used to deform the filled surface. After the photograph matching process, the same information is used to formulate an energy minimization problem based on photo-consistency and Laplacian smoothing.

A combination of global surface adjustment techniques and texture synthesis techniques is presented by Breckon and Fisher (2005). As usual, in the first stage, the underlying surface is completed by using simple geometric techniques, such as those proposed by Dell'acqua and Fisher (2002), Castellani *et al.* (2002) or Stulp and Fisher (2001). In the second stage, the surface's texture is extended from the viewed portion toward the previously created or completed surface. To do this, an adaptation of non-parametric 2D texture synthesis techniques, proposed by Efros *et al.* (1999), is used. Based on the spread of knowledge from visible portions toward invisible ones, the method completes the surface with a realistic appearance. This process is governed by the geometric constraint imposed by the filling done in the first part of the approach. Although the obtained filling is not an accurate reconstruction of the hole, it is acceptable to define a coarse shape of the entire object (Fig. 8).

The method of Pérez *et al.*(2008; 2012) extends the Roth and Black image inpainting algorithm (Roth and Black, 2005) to fill holes in 3D. This algorithm uses the idea of image coding to learn the parameters of Markov random fields (MRFs). A 3D partial view containing the hole is projected onto a plane to obtain the corresponding
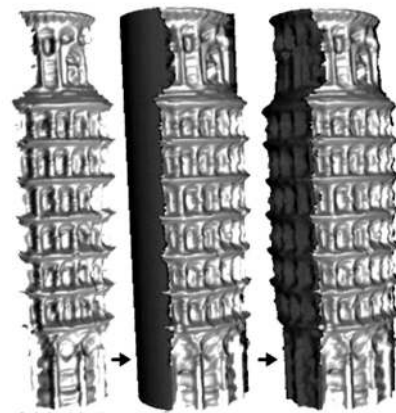


Fig. 8. Structured surface completion of a model of Pisa Tower by applying texture synthesis. The method presented by Breckon and Fisher (2005).

2D range image. The image restoration algorithm is then applied to the range image, in which empty areas correspond to 3D holes. Once the filling process is finished, the inverse transformation 2D to 3D is performed and the surface is repaired. This is a method that provides very good results with holes of different shapes and sizes. Nevertheless, it has the disadvantage that the hole must provide a univocal projection on the plane. Figure 9 illustrates a hole-filling example after this method has been used.

Lui and Gu (2013) also employ an inpainting technique, inspired by 2D image restoration. The contributions of this paper are twofold. First, the method considers the representation of a Riemann surface using its conformal factor $\lambda$ and the mean curvature H. Given these scalar functions, $\lambda$ and H, the associated Riemann surface can be reconstructed by solving the Gauss–Codazzi equation. Second, a novel surface inpainting technique by inpainting the scalar functions $\lambda$ and H is proposed. The method is tested on synthetic data, 3D human face data and MRI-derived brain surfaces. Experimental results demonstrate that the algorithm can effectively inpaint holes in surfaces and restore the incomplete 3D surface models, following lineal structures in smooth surfaces.

**5.2. Context-based methods.** As is known, in context-based learning the system learns through the actual and practical experience, avoiding mere theoretical and heuristic strategies. In the hole-filling field, instead of making *a priori* assumptions and calculations about the parts to be filled, the system can analyze existing surfaces by looking for known surface patterns or patterns that are repeated in the mesh. These patterns are later used to fill the holes. This is the core of context-based methods.

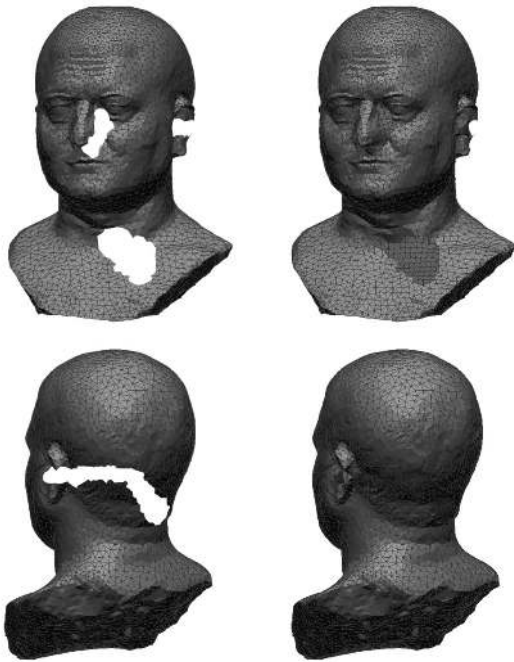The method presented by Sharf *et al.* (2004) is able

Fig. 9. Filling method which applies 2D image restoration techniques. The method presented by Pérez *et al.* (2008; 2012).



Fig. 10. Use of context information in order to fill the hole in a head. The method presented by Harary *et al.* (2014).

to identify surface patterns. In the first stage, the selected hole is coarsely filled. Then, refinement is carried out by fitting the surface-pattern on it. The given mesh pattern is fitted to the mesh through the ICP technique together with a small elastic deformation algorithm. To discretize the space, they use an octree structure, which allows them to manage low and high levels of detail in the mesh throughout the whole process. The results seem to be good for medium size holes but are limited by the relation between the data sampling density and the detail frequency. To capture fine structural details, the cell must be small enough with respect to the detail size. Another restriction is that the result of the surface completion procedure can only contain copies from the example set. If no appropriate examples exist, the match might be erroneous.

The approach presented by Vichitvejpaisal and Kanongchaiyos (2014) has the disadvantage that it works only on surfaces with strong geometric variations, being ineffective on smooth surfaces. This method handles surfaces with relief patterns (near-regular patterns, irregular patterns and stochastic patterns). In order to decompose the model into two parts: the coarse mesh (the low-frequency part of the surface) and the relief mesh (the high-frequency part of the surface), a multi-resolution approach is used. First, the hole of the coarse mesh is smoothly filled, and then the relief pattern is transferred to this hole following the idea of the example-based framework of texture synthesis of Wei *et al.* (2009). It
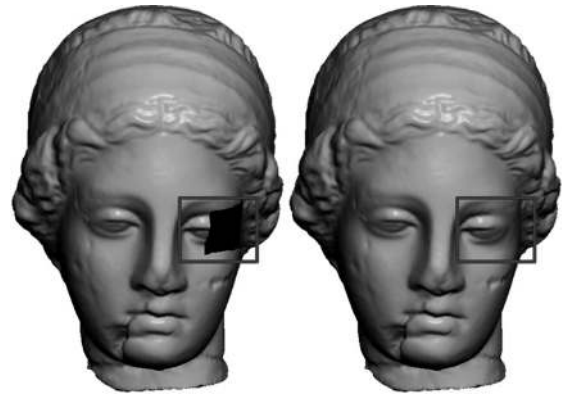
is worth mentioning that the quality of the coarse mesh determines the quality of the filled surface. Thus, if the mesh is not smooth enough, the relief pattern may not be easily detectable.

Harary *et al.* (2014) introduce a context-based algorithm to synthesize geometry that is similar to the remainder of the input mesh. This algorithm goes further by imposing a coherence objective. A synthesis is coherent if every local neighborhood of the filled hole is similar to some local neighborhood of the input mesh. This requirement avoids undesired features such as those that can occur in a context-based completion. For each target region, several candidate source patches are found using a multi-scale signature. From there, the inserted surface is iteratively refined to minimize coherence error (Fig. 10). A limitation of this method is that the completion might be smoother than desired when the hole is large and there are not enough features on its surroundings. In addition, this method is slower than others.

## 6. Comparison of hole-filling methods

After making a review of the main hole-filling methods, our intention is to establish a comparison among them. As is known, authors do not follow a particular pattern when they present their approaches and experimental results. Thus, we found authors who give complete information (they even offer the code and demos to check their approach), and those who only provide visual evidence of the results and do not quantitatively evaluate the method. Therefore, it is extremely difficult to establish comparisons among the techniques referenced in this paper.

On the other hand, of all the previously described methods, we have mentioned (Section 3) some which do not identify holes. As such, we do not consider them purely hole-filling methods and we do not include them in the comparison. We are aware that all kinds

of comparisons become a controversial issue in which a variety of reasonable opinions ovvur.

In Table 1 all the methods used in the comparison are listed with an associated ID number. Hereinafter, we refer to these methods using their corresponding IDs.

Depending on each specific application, input, shape and type of hole, several factors should be taken into account before choosing the most appropriate technique. We first considered some general properties for comparison, as for example: the size of the holes that the algorithm is able to fill, the ability of the method to fill sharp or irregular areas, its efficiency in CAD models, the data/mesh requirements, the computational cost of the method, whether the method deals with multiple boundary loops, etc. Nevertheless, we made an effort to draw out other interesting aspects from each hole-filling process. Tables 2 and 3 summarize up to 19 properties of the methods, each with several possibilities. In some cases we include the acronym NR, meaning that the characteristic is 'not reported' by the authors or it is not possible to infer this property from the information contained in the paper. A brief description of the features considered in the comparison follows.

F1 *Type of method.* We distinguish (i) methods based on polygonal representation; (ii) methods based on parametric representations; (iii) methods based on signed distance functions and volumetric representations; (iv) 2D image based methods; (v) context-based methods.

F2 *Object.* This refers to the kind of object which the method is applicable to: free-form shapes (**F**) and polyhedral shapes (**P**).

F3 *Hole size.* The size is measured with respect to the total area of the object. Thus, we distinguish between big (>3%) (**B**) and small (<3%) (**S**).

F4 *Shape of the hole boundary.* Here we have regular (**R**) (short narrow area, regular area) or irregular (**I**).

F5 *Multiple boundary loops.* We identify the methods which deal with more than a single boundary loop. The assessment is **Yes** or **No**.

F6 *Curvature of the filled area and its surroundings.* The curvature can be low/medium (**L**) or high (**H**).

F7 *Curvature gradient of filled area.* This property concerns the curvature variation of the filled region. Thus, it can be low gradient/low (**L**), which means smooth surface, or high gradient/high (**H**), which signifies high curvature frequency.

F8 *Sensitivity to noise.* We identify the papers that argue or demonstrate that the method works under noisy conditions. As far as we know, none of the methods

included in Table 2 have been tested under noise; nevertheless, two of them, nos. 29 and 34, state that the method would work under noisy conditions. Others state that if the input is too noisy the filling process might fail (no. 32). The assessment here is **Yes** (which means the method works) or **No**.

F9 *Distortion.* This analyzes whether the method introduces distortions around the hole's boundary or on the whole surface of the object (**Yes** or **No**).

F10 *Model requirements.* Some methods only work if certain properties of the model are verified. We distinguish here between **Yes** or **No**. Usual requirements are **Rs**: high resolution of the mesh or voxel model (no. 32); **Rg**: regularity of the model (for example, in no. 8 homogenization of the edge lengths is required); **P**: shape of the patches; **OCM**: oriented connected manifold (an example can be found in no. 24).

F11 *Raw point-sampled data requirements.* This property is evaluated as **Yes** or **No** and depends on particular demands in the data acquisition stage. For example, some methods need repetition of patterns in the piece to be dealt with (nos. 33 and 34), others require color information to be included in the data (no. 29), etc.

F12 *High complexity of the method.* We take into account the complexity of the algorithms from beginning to end. Since this information is not provided by the majority of the authors, we only evaluate the complexity from the theoretical point of view by assessing the asymptotic behavior of the algorithms. We give a simple **Yes** or **No**. We are aware that rating complexity in a binary manner is always risky. For example, rating no. 21 'Yes' and no. 25 'No' could be debatable.

F13 *Computational time.* High (**H**) or acceptable (**A**). This classification has been made from a qualitative point of view after the observation of the results presented in each article. In order to offer quantitative values of time, we have added a comparison which can be observed in Table 6. To build this table, we computed the relation between the time taken to fill the mesh and the number of vertices of the input mesh for the methods whose articles report this information. It must be pointed out that the values shown in this table depend on many factors (resolution of the input mesh, pre-processing step, complexity of the algorithm, etc.), and so they should be taken as illustrative.

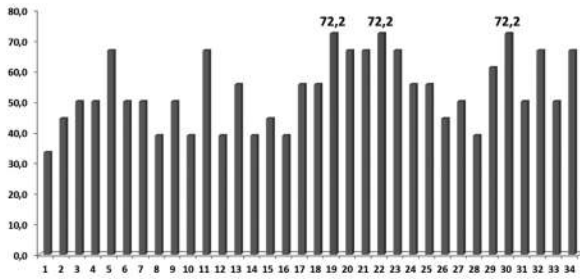F14 *Demo.* There is a demo or the hole-filling algorithm code is available on a specific website (**Yes** or **No**).

Fig. 11. Graph representing the percentage of positive aspects for each method.

F15 *Evaluation.* We here give information about how the method has been evaluated. We consider two possibilities: qualitative (**Ql**) (visual) and quantitative (**Qn**) evaluation of the method.

F16 *Linear structures.* The method is able or not to deal with holes located inside a surface linear pattern **Yes** or **No**). Some examples of linear structures are an eyebrow on a face, a canal, a track or the edge of a polyhedron.

F17 *Pre-process.* Some approaches require a data preprocessing stage to be carried out before applying the specific hole-filling algorithm; for example, cleaning (nos. 10 and 30) or deletion of badly oriented triangles (no. 8). Another example can be found in no. 4, where, in order to reduce the occurrence of self-intersections, Steiner points are added to the mesh. The assessment is **Yes** or **No**.

F18 *Post-process.* Sometimes post-processing tasks (smoothing(**S**), refinement (**Rf**), repairing (**Rp**)) are necessary. We evaluate this feature as **Yes** or **No**.

F19 User intervention (**Yes** or **No**). This characteristic refers to the fact that the user could take part in algorithm execution. In some papers the intervention of the user is suggested.

Tables 2 and 3 present all these features, using the numbering established in Table 1. Needless to say, despite the diversity of techniques, there is no universal filling method valid for all situations. Note the large variety of holes: large, small, regular/irregular, holes with high/low curvature, holes located at corners, edges or sharp areas, holes with interior islands, etc.

One way to compare methods is to give a qualitative score for each feature and put all the methods together in a table. Tables 4 and 5 collect the assessment of the 19 properties for 34 approaches in which some checkmarks appear. The assessment is very simple: for each feature we add one tick for each positive aspect. There are cases in which this rule must be clarified. For example, for Feature 2 (F2), we only add a tick if the method runs for

Table 1. List of methods numbered.

| ID number | Method |
|---|---|
| 1 | Barequet and Sharir (1995) |
| 2 | Liepa (2003) |
| 3 | Wei *et al*. (2010) |
| 4 | Brunton *et al*. (2009) |
| 5 | Zhao *et al*. (2007) |
| 6 | Wang and Hung (2012) |
| 7 | Wang and Oliveira (2007) |
| 8 | Pernot *et al*. (2006) |
| 9 | Hu *et al*. (2012) |
| 10 | Wang *et al*. (2012) |
| 11 | Li *et al*. (2008) |
| 12 | Ngo and Lee (2013) |
| 13 | Zhao *et al*. (2006) |
| 14 | Branch *et al*. (2006) |
| 15 | Wu *et al*. (2008) |
| 16 | Kumar *et al*. (2007) |
| 17 | Li *et al*. (2010) |
| 18 | Murali and Funkhouser (1997) |
| 19 | Ju (2004) |
| 20 | Bischoff *et al*. (2005) |
| 21 | Davis *et al*. (2001) |
| 22 | Guo *et al*. (2006) |
| 23 | Caselles *et al*. (2008) |
| 24 | Sagawa and Ikeuchi (2008) |
| 25 | Podolak and Rusinkiewicz (2005) |
| 26 | Nooruddin and Turk (2003) |
| 27 | Kumar and Shih (2012) |
| 28 | Brunton *et al*. (2007) |
| 29 | Breckon and Fisher (2005) |
| 30 | Pérez *et al*. (2008) |
| 31 | Lui and Gu (2013) |
| 32 | Sharf *et al*. (2004) |
| 33 | Vichitvejpaisal and Kanongchaiyos (2014) |
| 34 | Harary *et al*. (2014) |

both types. Thus, when in a feature the method covers more than one case or aspect, we consider that it is more versatile and we give it a tick. This mainly occurs in features F1, F2 and F15, although it can be also verified in F3, F4, F5, F6, F7.

Some of the ticks are obvious, as in F8, F9, F12 and F13. Of course, for the rest of the features we make reasonable assumptions. For example, we consider that techniques which deal with big holes (F3), irregular hole contours (F4), high curvature gradients (F7), without model and data requirements (F10 and F11), with available demos (F14) and a complete evaluation report (F15), without pre- and post-processing stages (F17 and F18) should be positively evaluated. As regards feature F6, it is assumed that flat surfaces are easily filled, whereas irregular or high-curvature zones are more difficult to manage. The evaluation of feature F19, concerning user intervention, might be the subject of debate. Although fully automatic hole-filling solutions
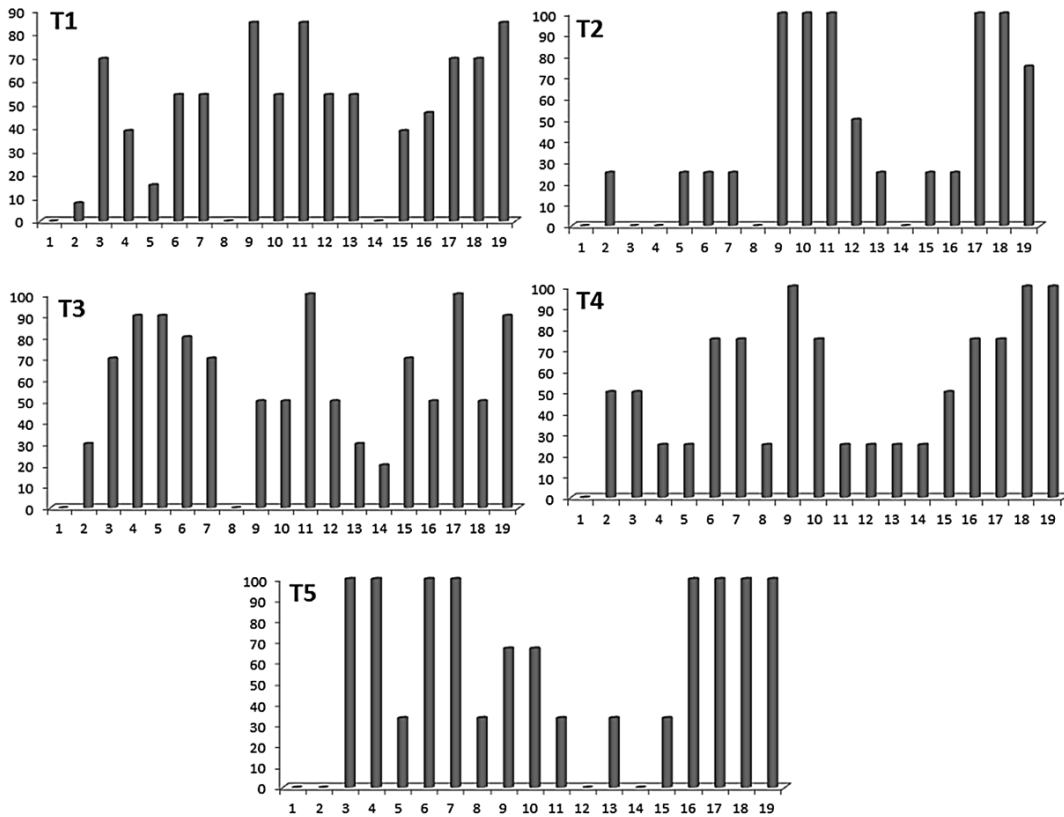
Fig. 12. Mean percentages versus the features for each type of method.

have been investigated for years, the truth is that this is a difficult goal which has not yet been achieved. In fact, one of the future lines of research lies in incorporating a human expert in the filling process. In any case, the methods without user intervention are rated as positive in Table 3.

An illustrative graph is added in Fig. 11. It shows the percentages of ticks (i.e., positive aspects) of each of the 34 approaches. As mentioned above, we can only state that, under our particular evaluation procedure, the algorithms which have accumulated a higher number of positive aspects can be considered versatile or complete solutions for the hole-filling problem. Nevertheless, we leave readers to draw their own conclusions about the suitability of each method for each particular application and circumstances.

In Fig. 12 the mean percentages versus the features for each type of method (F1) are presented. From this figure the respective strengths and lacks per type can be extracted. Some comments of note are as follows.

- In general, Features F8 (sensitivity to noise) and F14 (demo) are seldom satisfied in all the methods, whereas most of them satisfy F19 (user intervention not needed).

- Type T1 achieves maximum percentages in F3 (hole

size), F9 (distortion) and F11 (raw point-sampled data requirements), and has very low values for F2 (object) and F5 (multiple boundary loops).

- Type T2 seems to be effective for F9 (distortion), F10 (models requirements), F11 (raw point-sampled data requirements), F17 (pre-process) and F18 (post-process) but not for features F3 (hole size), F4 (shape of the holes boundary) and F5 (multiple boundary loops).

- Type T3 clearly satisfies features F4 (shape of the hole's boundary), F5 (multiple boundary loops), F6 (curvature of the filled area), F11 (raw point-sampled data requirements) and F17 (pre-process).

- Type T4 stands out in features F9 (distortion) and F18 (post-process). The rest of the features are above 25%.

- Type T5 has the highest score (100%) for many features (F3, F4, F6, F7, F16, F17, F18 and F19) and lowest score (0%) for F2 (object), F12 (high complexity of the method) and F14 (demo).

The average of the mean percentages is illustrated in Fig. 13. As can be seen, there are no meaningful distances between all methods. The highest values, above

Table 2. Features for comparing methods 1–9.

| Method | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| | Type | Object shape | Size | Hole contour | Multiple boundaries | Curv. | Curv. gradient | Noise | Distortion | Model requirements |
| 1 | 1 | P | S | R | No | L | L | – | No | Yes |
| 2 | 1 | F | B | I | Yes | H | L | – | No | Yes, OCM |
| 3 | 1 | F | B | I | No | L | L | – | No | Yes, OCM |
| 4 | 1 | F | B | I | No | H | H | – | No | No |
| 5 | 1 | F | B | I | No | H | H | – | No | No |
| 6 | 1 | F | B | R | No | L | L | – | No | No |
| 7 | 1 | F | B | R | Yes | L | L | – | No | No |
| 8 | 1 | F | B | R | No | H | H | – | No | Yes, OCM |
| 9 | 1 | F/P | B | R | No | H | L | – | No | Yes, OCM |
| 10 | 1 | P | B | R | No | L | H | – | No | Yes, OCM |
| 11 | 1 | F | S | I | No | H | H | – | No | No |
| 12 | 1 | P | S | R | No | L | H | – | No | No |
| 13 | 1 | P | S | R | No | H | H | – | No | No |
| 14 | 2 | F | S | R | No | L | L | – | No | No |
| 15 | 2 | F | B | R | No | L | L | – | No | No |
| 16 | 2 | F | S | R | No | L | L | – | No | No |
| 17 | 2 | F/P | S | R | Yes | H | H | – | No | No |
| 18 | 3 | P | B | R | No | H | H | – | No | No |
| 19 | 3 | F | B | I | Yes | H | H | – | No | No |
| 20 | 3 | F/P | B | I | Yes | H | H | – | No | No |
| 21 | 3 | F | S | I | Yes | H | H | – | No | Yes, OCM, Rs |
| 22 | 3 | F/P | B | I | Yes | H | H | – | Yes | Yes, Rs |
| 23 | 3 | F/P | B | I | Yes | H | L | – | No | No |
| 24 | 3 | F | B | I | Yes | L | H | – | No | Yes, OCM |
| 25 | 3 | F | B | I | Yes | H | L | – | No | Yes, OCM |
| 26 | 3 | F | B | I | Yes | L | L | – | Yes | No |
| 27 | 3 | F | B | I | Yes | H | H | – | No | Yes |
| 28 | 4 | F | S | R | No | L | L | – | No | No |
| 29 | 4 | F/P | B | R | No | H | H | Yes | No | No |
| 30 | 4 | F/P | B | I | Yes | H | H | – | No | Yes, OCM |
| 31 | 4 | F | B | R | No | H | H | – | No | No |
| 32 | 5 | F | B | I | Yes | H | H | No | No | No |
| 33 | 5 | F | B | I | No | H | H | – | Yes | Yes, Rs |
| 34 | 5 | F | B | I | No | H | H | Yes | No | No |

60%, correspond to Type 3 (volumetric representations) and Type 5 (context-based methods). The rest of the types are around 50%.

Finally, Fig. 14 summarizes mean percentages per type versus feature. Values have been scaled between 0 and 1 to represent this information as a grey image. A bright pixel means that most of the methods of the same type tick the corresponding feature. For example, the majority of methods T3 are able to deal with irregular hole contours, which correspond to feature F4.

## 7. Conclusions

This article presents a review of a wide variety of hole-filling techniques that have been proposed in journals and conferences. Although several surveys dealing with mesh repairing have been presented in
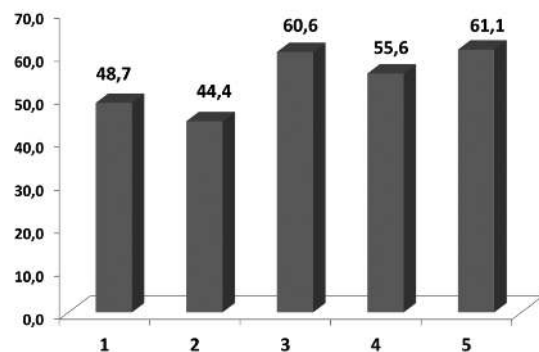
Fig. 13. Averages of the mean percentages per type.

the past, this paper focuses specifically on the problem of hole-filling. The survey has been organized taking into account a classification which distinguishes between

Table 3. Features for comparing methods 11–19.

| Method | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
|---|---|---|---|---|---|---|---|---|---|
| | Data require-ments | High complexi-ty | Time | Demo | Evalu-ation | Linear struc-tures | Pre-process process | Post-process process | User inter-vention |
| 1 | No | No | A | No | Qn | No | Yes | No | No |
| 2 | No | No | NR | Yes | Ql | NR | No | Yes:S,Rf | No |
| 3 | No | No | A | No | Qn | NR | No | Yes: S, Rf | No |
| 4 | No | Yes | A | No | Ql | NR | Yes: Steiner points | Yes: Rf | No |
| 5 | No | Yes | A | No | Qn | NR | No | No | No |
| 6 | No | Yes | A | No | Qn | NR | No | No | No |
| 7 | Yes | Yes | A | No | Qn | NR | No | No | No |
| 8 | No | Yes | NR | No | Ql | Yes | Yes | No | Yes |
| 9 | No | No | NR | No | Ql | Yes, Limited | No | Yes: Rf | No |
| 10 | No | Yes | NR | No | Ql | Yes | Yes: Cleaning | No | No |
| 11 | No | No | A | No | Ql | Yes | No | No | No |
| 12 | Yes | No | NR | No | Ql | Yes | No | No | Yes |
| 13 | No | No | NR | No | Ql | Yes | No | No | No |
| 14 | No | No | NR | No | Ql | NR | No | No | No |
| 15 | No | Yes | A | No | Qn | NR | No | No | No |
| 16 | No | No | NR | No | Ql | No | No | No | No |
| 17 | No | Yes | NR | No | Ql | Yes | No | No | Yes |
| 18 | No | No | H | No | Qn | No | No | No | No |
| 19 | No | No | H | No | Qn | Yes | No | No | No |
| 20 | No | Yes | H | No | Qn | Yes | No | Yes:S | No |
| 21 | No | Yes | H | Yes | Qn | Yes, Limited | No | No | No |
| 22 | No | Yes | A | No | Qn | Yes | No | No | No |
| 23 | No | Yes | NR | No | Ql | Yes, Limited | No | No | No |
| 24 | No | Yes | A | No | Qn | NR | No | Yes: S | No |
| 25 | No | No | A | No | Qn | No | No | Yes: S | Yes |
| 26 | No | No | NR | Yes | Ql | NR | No | Yes: S, Rp | No |
| 27 | No | No | NR | No | Ql | NR | No | Yes: Rp | No |
| 28 | Yes | No | NR | No | Qn | No | No | No | No |
| 29 | Yes | Yes | H | No | Ql | Yes | No | No | No |
| 30 | No | Yes | NR | No | Qn | Yes, Limited | Yes: Cleaning | No | No |
| 31 | Yes | Yes | NR | Yes | Ql | Yes, Limited | No | No | No |
| 32 | No | Yes | H | No | Ql | Yes, Limited | No | No | No |
| 33 | Yes | Yes | A | No | Ql | Yes, Limited | No | No | No |
| 34 | Yes | Yes | H | No | Qn | Yes | No | No | No |
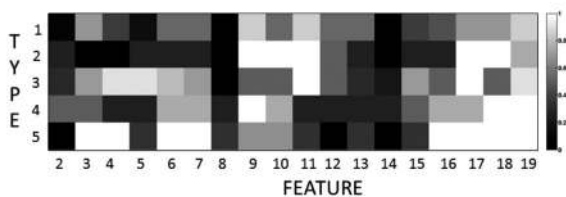


Fig. 14. Integrated image containing the mean percentages per type versus features.

methods based on polygonal representations, those based on parametric representations and the ones based on volumetric representations. We have also included other non-classical methods in a separate section.

Contributions of the paper are addressed to give a more detailed explanation of each method in its context and, more importantly, to provide their general comparison.

In order to obtain the comparative Tables 1 and 2, we have used the information which appears in the respective papers and we have also evaluated other interesting aspects. Hence, up to 19 meaningful characteristics have been analyzed in 34 methods. Aspects such as the size and shape of the holes that the algorithm is able to fill, the ability of the method to fill sharp or flat areas, the response to lineal structures (specifically in CAD models), complexity, special requirements, sensitivity to noise, and other features have been considered. A qualitative comparison is also established in Tables 3 and 4. For each feature, we have marked the method that has a good or positive performance. Thus, the readers themselves can

Table 4. Result of the voting process for features 1-10

| No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|-----|------|----------------|------|------------------|------------------------|-----------|----------------------|-------|------------|------------------------|
| | Type | Object shape | Size | Hole contour | Multiple boundaries | Curvature | Curvature gradient | Noise | Distortion | Model requirements |
| 1 | 1 | – | – | – | – | – | – | – | – | – |
| 2 | 1 | – | ✓ | ✓ | ✓ | ✓ | – | – | – | – |
| 3 | 1 | – | ✓ | ✓ | – | – | – | – | ✓ | – |
| 4 | 1 | – | ✓ | ✓ | – | ✓ | ✓ | – | ✓ | ✓ |
| 5 | 1 | – | ✓ | ✓ | – | ✓ | ✓ | – | ✓ | ✓ |
| 6 | 1 | – | ✓ | – | – | – | – | – | ✓ | ✓ |
| 7 | 1 | – | ✓ | – | ✓ | – | – | – | ✓ | ✓ |
| 8 | 1 | - | ✓ | – | – | ✓ | ✓ | – | ✓ | – |
| 9 | 1 | ✓ | ✓ | – | – | ✓ | – | – | ✓ | – |
| 10 | 1 | – | ✓ | – | – | – | ✓ | – | ✓ | – |
| 11 | 1 | – | – | ✓ | – | ✓ | ✓ | – | ✓ | ✓ |
| 12 | 1 | – | – | – | – | – | ✓ | – | ✓ | ✓ |
| 13 | 1 | – | – | – | – | ✓ | ✓ | – | ✓ | ✓ |
| 14 | 2 | – | – | – | – | – | – | – | ✓ | ✓ |
| 15 | 2 | – | – | – | – | – | – | – | ✓ | ✓ |
| 16 | 2 | – | – | – | – | – | – | – | ✓ | ✓ |
| 17 | 2 | ✓ | – | – | ✓ | ✓ | ✓ | – | ✓ | ✓ |
| 18 | 3 | – | ✓ | – | – | ✓ | ✓ | – | – | ✓ |
| 19 | 3 | – | ✓ | ✓ | ✓ | ✓ | ✓ | – | – | ✓ |
| 20 | 3 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | – | – | ✓ |
| 21 | 3 | – | – | ✓ | ✓ | ✓ | ✓ | – | ✓ | – |
| 22 | 3 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | – | – | – |
| 23 | 3 | ✓ | ✓ | ✓ | ✓ | ✓ | – | – | ✓ | ✓ |
| 24 | 3 | – | ✓ | ✓ | ✓ | – | ✓ | – | ✓ | – |
| 25 | 3 | – | ✓ | ✓ | ✓ | ✓ | – | – | ✓ | – |
| 26 | 3 | – | – | ✓ | ✓ | – | – | – | – | ✓ |
| 27 | 3 | – | – | ✓ | ✓ | ✓ | ✓ | - | ✓ | – |
| 28 | 4 | – | – | – | – | – | – | – | ✓ | ✓ |
| 29 | 4 | ✓ | ✓ | – | – | ✓ | ✓ | ✓ | ✓ | ✓ |
| 30 | 4 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | – | ✓ | – |
| 31 | 4 | – | – | – | – | ✓ | ✓ | – | ✓ | ✓ |
| 32 | 5 | – | ✓ | ✓ | ✓ | ✓ | ✓ | – | ✓ | ✓ |
| 33 | 5 | – | ✓ | ✓ | – | ✓ | ✓ | – | – | – |
| 34 | 5 | – | ✓ | ✓ | – | ✓ | ✓ | ✓ | ✓ | ✓ |



Fig. 15. Maximum averages of positive percentages.

mentioned in this paper and proposed types. Several interesting graphs regarding mean percentages versus the features for each type of method and average of the mean percentages are included and discussed in the last part of the document.

As mentioned before, the hole-filling problem is not a trivial one and has no general solution. Consequently, what we find are efficient solutions for large or small holes, methods that work in holes with high or low curvature, approaches which are suitable for holes located at corners, in edges or sharp areas, whereas others run in smooth and plane areas. The question is that capturing the whole surface of an object can involve an inaccurate and time-consuming process in which the lack of data is always present. Therefore, in order to build automatic realistic 3D models, the hole-filling field still needs to be developed and improved in the future. From

evaluate the goodness and versatility of each approach. Additionally, in order to provide a general idea of the main strengths and limitations that characterize the hole-filling field, we have analyzed these features in the methods

Table 5. Result of the voting process for features 11–19.

| No. | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
|-----|----|----|----|----|----|----|----|----|----|
| | Data requirements | High complexity | Time | Demo | Evaluation | Linear structures | Preprocess process | Postprocess process | User intervention |
| 1 | ✓ | ✓ | ✓ | – | ✓ | – | – | ✓ | ✓ |
| 2 | ✓ | ✓ | – | ✓ | – | – | ✓ | – | ✓ |
| 3 | ✓ | ✓ | ✓ | – | ✓ | – | ✓ | – | ✓ |
| 4 | ✓ | – | ✓ | – | – | – | – | – | ✓ |
| 5 | ✓ | – | ✓ | – | ✓ | – | ✓ | ✓ | ✓ |
| 6 | ✓ | – | ✓ | – | ✓ | – | ✓ | ✓ | ✓ |
| 7 | – | – | ✓ | – | ✓ | – | ✓ | ✓ | ✓ |
| 8 | ✓ | – | – | – | – | ✓ | – | ✓ | – |
| 9 | ✓ | ✓ | – | – | – | ✓ | ✓ | – | ✓ |
| 10 | ✓ | – | – | – | – | ✓ | – | ✓ | ✓ |
| 11 | ✓ | ✓ | ✓ | – | – | ✓ | ✓ | ✓ | ✓ |
| 12 | – | ✓ | – | – | – | ✓ | ✓ | ✓ | – |
| 13 | ✓ | ✓ | – | – | – | ✓ | ✓ | ✓ | ✓ |
| 14 | ✓ | ✓ | – | – | – | – | ✓ | ✓ | ✓ |
| 15 | ✓ | – | ✓ | – | ✓ | – | ✓ | ✓ | ✓ |
| 16 | ✓ | ✓ | – | – | – | – | ✓ | ✓ | ✓ |
| 17 | ✓ | – | – | – | – | ✓ | ✓ | ✓ | – |
| 18 | ✓ | ✓ | – | – | ✓ | – | ✓ | ✓ | ✓ |
| 19 | ✓ | ✓ | – | – | ✓ | ✓ | ✓ | ✓ | ✓ |
| 20 | ✓ | – | – | – | ✓ | ✓ | ✓ | - | ✓ |
| 21 | ✓ | – | – | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 22 | ✓ | – | ✓ | – | ✓ | ✓ | ✓ | ✓ | ✓ |
| 23 | ✓ | – | – | – | – | ✓ | ✓ | ✓ | ✓ |
| 24 | ✓ | – | ✓ | – | ✓ | – | ✓ | – | ✓ |
| 25 | ✓ | ✓ | ✓ | – | ✓ | – | ✓ | – | – |
| 26 | ✓ | ✓ | – | ✓ | – | – | ✓ | – | ✓ |
| 27 | ✓ | ✓ | – | – | – | – | ✓ | – | ✓ |
| 28 | – | ✓ | – | – | ✓ | – | ✓ | ✓ | ✓ |
| 29 | – | – | – | – | – | ✓ | ✓ | ✓ | ✓ |
| 30 | ✓ | – | ✓ | – | ✓ | ✓ | - | ✓ | ✓ |
| 31 | – | – | – | ✓ | – | ✓ | ✓ | ✓ | ✓ |
| 32 | ✓ | – | – | – | – | ✓ | ✓ | ✓ | ✓ |
| 33 | – | – | ✓ | – | – | ✓ | ✓ | ✓ | ✓ |
| 34 | – | – | – | – | ✓ | ✓ | ✓ | ✓ | ✓ |

Fig. 15, which shows the maximum average of positive percentages per type vs. feature, one can infer that, according to the assessment performed, none of the types is able to satisfy all the features. Thus, apart from features F8 (33%, sensitivity to noise) and F14 (25%, demo), features F2 (object shape), F12 (high complexity of the method) and F13 (computational time) have maximum percentages around 50%, which means that these are key aspects that should be addressed in future studies.

## Acknowledgment

## References

Amenta, N., Bern, M. and Kamvysselis, M. (1998). A new Voronoi-based surface reconstruction algorithm, *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'98), Orlando, FL, USA*, pp. 415–421.

Attene, M., Campen, M. and Kobbelt, L. (2013). Polygon mesh repairing: An application perspective, *ACM Computing Surveys* **45**(2): 15:1–15:33.

Bajaj, C., Bernardini, F. and Xu, G. (1995). Automatic reconstruction of surfaces and scalar fields from 3D scans, *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'95), Los Angeles, CA, USA*, pp. 109–118.

Barequet, G. and Sharir, M. (1995). Filling gaps in the boundary of a polyhedron, *Computer-Aided Geometric Design* **12**(2): 207–229.

Table 6. Comparison taking into account the relation between the time taken to fill the mesh and the number of vertices of the input.

| Method | Time (s) per vertex |
|--------|---------------------|
| 1 | $3.34 \times 10^{-2}$ |
| 3 | $4.23 \times 10^{-2}$ |
| 4 | $3.46 \times 10^{-3}$ |
| 5 | $6.66 \times 10^{-7}$ |
| 6 | $8.47 \times 10^{-6}$ |
| 7 | $1.30 \times 10^{-3}$ |
| 11 | $3.33 \times 10^{-5}$ |
| 15 | $4.51 \times 10^{-3}$ |
| 18 | $1.92 \times 10^{-1}$ |
| 19 | $1.55 \times 10^{-4}$ |
| 20 | $2.93 \times 10^{-2}$ |
| 21 | $2.73 \times 10^{-6}$ |
| 22 | $9.11 \times 10^{-4}$ |
| 24 | $1.40 \times 10^{-3}$ |
| 25 | $2.31 \times 10^{-3}$ |
| 29 | $4.68 \times 10^{+4}$ |
| 32 | $4.27 \times 10^{-4}$ |
| 33 | $3.75 \times 10^{-2}$ |
| 34 | $5.26 \times 10^{-3}$ |

Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C. and Taubin, G. (1999). The ball-pivoting algorithm for surface reconstruction, *IEEE Transactions on Visualization and Computer Graphics* **5**(4): 349–359.

Bischoff, S., Pavic, D. and Kobbelt, L. (2005). Automatic restoration of polygon models, *ACM Transactions on Graphics* **24**(4): 1332–1352.

Bloomenthal, J. (1994). *Graphics gEMS IV*, Academic Press Professional, Inc., San Diego, CA, pp. 324–349.

Botsch, M. and Kobbelt, L. (2004). A remeshing approach to multiresolution modeling, *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, SGP'04, Orlando, FL, USA*, pp. 185–192.

Branch, J., Prieto, F. and Boulanger, P. (2006). A hole-filling algorithm for triangular meshes using local radial basis function, *Proceedings of the 15th International Meshing Roundtable, Birmingham, AL, USA*, pp. 411–431.

Breckon, T. and Fisher, R. (2005). Non-parametric 3D surface completion, *Proceedings of the 5th International Conference on 3-D Digital Imaging and Modeling (3DIM'05), Ottawa, Ontario, Canada*, pp. 573–580.

Brunton, A., Wuhrer, S. and Shu, C. (2007). Image-based model completion, *Proceedings of the 6th International Conference on 3-D Digital Imaging and Modeling (3DIM'07), Montreal, Quebec, Canada*, pp. 305–311.

Brunton, A., Wuhrer, S., Shu, C., Bose, P. and Demaine, E. (2009). Filling holes in triangular meshes by curve unfolding, *Proceedings of the 2009 IEEE International Conference on Shape Modeling and Applications (SMI'09), Beijing, China*, pp. 66–72.

Carr, J., Beatson, R., Cherrie, J., Mitchell, T., Fright, W., McCallum, B. and Evans, T. (2001). Reconstruction and representation of 3D objects with radial basis functions, *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'01), Los Angeles, CA, USA*, pp. 67–76.

Caselles, V., Haro, G., Sapiro, G. and Verdera, J. (2008). On geometric variational models for inpainting surface holes, *Computer Vision and Image Understanding* **111**(3): 351–373.

Castellani, U., Livatino, S. and Fisher, R. (2002). Improving environment modelling by edge occlusion surface completion, *1st International Symposium on 3D Data Processing Visualization and Transmission (3DPVT 2002), Padova, Italy*, pp. 672–675.

Curless, B. and Levoy, M. (1996). A volumetric method for building complex models from range images, *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'96), New Orleans, LA, USA*, pp. 303–312.

Davis, J., Marschner, S., Garr, M. and Levoy, M. (2001). Filling holes in complex surfaces using volumetric diffusion, *Proceedings of the 1st International Symposium on 3D Data Processing, Visualization and Transmission, Padova, Italy*, pp. 428–438.

Dell'acqua, F. and Fisher, R. (2002). Reconstruction of planar surfaces behind occlusions in range images, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(4): 569–575.

Dey, T., Giesen, J. and Hudson, J. (2001). Delaunay based shape reconstruction from large data, *Proceedings of the IEEE 2001 Symposium on Parallel and Large-Data Visualization and Graphics (PVG'01), San Diego, CA, USA*, pp. 19–27.

Dinh, H. and Turk, G. (2001). Reconstructing surfaces using anisotropic basis functions, *Proceedings of the 2001 International Conference on Computer Vision (ICCV'01), Vancouver, British Columbia, Canada*, pp. 606–613.

Edelsbrunner, H. and Mücke, E. (1994). Three-dimensional alpha shapes, *ACM Transactions on Graphics* **13**(1): 43–72.

Efros, A., Efros, A. and Leung, T. (1999). Texture synthesis by non-parametric sampling, *Proceedings of the 2001 International Conference on Computer Vision (ICCV'01), Vancouver, British Columbia, Canada*, pp. 1033–1038.

Fang, T.-P. and Piegl, L. (1995). Delaunay triangulation in three dimensions, *IEEE Computer Graphics and Applications* **15**(5): 62–69.

Guo, T.-Q., Li, J.-J., Weng, J.-G. and Zhuang, Y.-T. (2006). Filling holes in complex surfaces using oriented voxel diffusion, *Proceedings of the 5th International Conference on Machine Learning and Cybernetics, Dalian, China*, pp. 4370–4375 .

Harary, G., Tal, A. and Grinspun, E. (2014). Context-based coherent surface completion, *ACM Transactions on Graphics* **33**(1): 5:1–5:12.

Hu, P., Wang, C., Li, B. and Liu, M. (2012). Filling holes in triangular meshes in engineering, *Journal of Software* **7**(1): 141–148.

Jakobsen, B., Bærentzen, J. and Christensen, N. (2007). Variational volumetric surface reconstruction from unorganized points, *Proceedings of the 6th Eurographics/IEEE VGTC Conference on Volume Graphics, VG'07, Prague, Czech Republic*, pp. 65–72.

Ju, T. (2004). Robust repair of polygonal models, *Technical report*, Rice University, Houston, TX.

Ju, T. (2009). Fixing geometric errors on polygonal models: A survey, *Journal of Computer Science Technology* **24**(1): 19–29.

Ju, T., Losasso, F., Schaefer, S. and Warren (2002). Dual contouring of hermite data, *SIGGRAPH02, San Antonio, TX, USA*, pp. 339–346.

Kazhdan, M., Bolitho, M. and Hoppe, H. (2006). Poisson surface reconstruction, *Proceedings of the 4th Eurographics Symposium on Geometry Processing, SGP'06, Cagliari, Sardinia, Italy*, pp. 61–70.

Klincsek, G. (1980). Minimal triangulations of polygonal domains, *Annals of Discrete Mathematics* **9**: 121–123.

Kobbelt, L., Campagna, S., Vorsatz, J. and Seidel, H.-P. (1998). Interactive multi-resolution modeling on arbitrary meshes, *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'98), Orlando, FL, USA*, pp. 105–114.

Kumar, A. and Shih, A. (2012). Hybrid approach for repair of geometry with complex topology, *in* W. Quadros (Ed.), *Proceedings of the 20th International Meshing Roundtable*, Springer, Berlin/Heidelberg, pp. 387–403.

Kumar, A., Shih, A., Ito, Y., Ross, D. and Soni, B. (2007). A hole-filling algorithm using non-uniform rational b-splines, *Proceedings of the 16th International Meshing Roundtable, Seattle, WA, USA*, pp. 169–182.

Lancaster, P. and Salkauskas, K. (1981). Surfaces generated by moving least squares methods, *Mathematics of Computation* **37**(155): 141–158.

Li, G., Ye, X. and Zhang, S. (2008). An algorithm for filling complex holes in reverse engineering, *Engineering with Computers* **24**(2): 119–125.

Li, Z., Meek, D. and Walton, D. (2010). Polynomial blending in a mesh hole-filling application, *Journal of Computer-Aided Design* **42**(4): pp. 340–349.

Liepa, P. (2003). Filling holes in meshes, *SGP'03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, San Diego, CA, USA*, pp. 200–205.

Lui, L. and Gu, X. (2013). A conformal approach for surface inpainting, *Inverse Problems and Imaging* **7**(3): 863–884.

Murali, T. and Funkhouser, T.A. (1997). Consistent solid and boundary representations from arbitrary polygonal data, *Proceedings of the 1997 Symposium on Interactive 3D Graphics, I3D'97, Providence, RI, USA*, pp. 155–162.

Ngo, H.-M. and Lee, W.-S. (2013). Feature-first hole filling strategy for 3D meshes, *in* G. Csurka *et al.* (Eds.), *Computer Vision, Imaging and Computer Graphics. Theory and Applications*, Communications in Computer and Information Science, Vol. 274, Springer, Berlin/Heidelberg, pp. 53–68.

Nooruddin, F. and Turk, G. (2003). Simplification and repair of polygonal models using volumetric techniques, *IEEE Transactions on Visualization and Computer Graphics* **9**(2): 191–205.

Paulsen, R.R., Baerentzen, J.A. and Larsen, R. (2010). Markov random field surface reconstruction, *IEEE Transactions on Visualization and Computer Graphics* **16**(4): 636–646.

Pérez, E., Salamanca, S., Cerrada, C., Merchán, P. and Adán, A. (2012). Filling holes in manifold digitized 3D meshes using image restoration algorithms, *Proceedings of the IEEE Intelligent Vehicles Symposium Workshops, Madrid, Spain*.

Pérez, E., Salamanca, S., Merchán, P., Adán, A., Cerrada, C. and Cambero, I. (2008). A robust method for filling holes in 3D meshes based on image restoration, *in* S. Bourennane *et al.* (Eds.), *Proceedings of the 10th International Conference on Advanced Concepts for Intelligent Vision Systems, ACIVS'08*, Springer-Verlag, Berlin/Heidelberg, pp. 742–751.

Pernot, J.-P., Moraru, G. and Vron, P. (2006). Filling holes in meshes using a mechanical model to simulate the curvature variation minimization, *Computers & Graphics* **30**(6): 892–902.

Podolak, J. and Rusinkiewicz, S. (2005). Atomic volumes for mesh completion, *Proceedings of the 3rd Eurographics Symposium on Geometry Processing (SGP'05), Vienna, Austria*, p. 33.

Roth, S. and Black, M.J. (2005). Fields of experts: A framework for learning image priors, *IEEE Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA*, pp. 860–867.

Sagawa, R. and Ikeuchi, K. (2008). Hole filling of a 3D model by flipping signs of a signed distance field in adaptive resolution, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **30**(4): 686–699.

Sharf, A., Alexa, M. and Cohen-Or, D. (2004). Context-based surface completion, *ACM Transactions on Graphics* **23**(3): 878–887.

Sorkine, O. and Cohen-Or, D. (2004). Least-squares meshes, *Proceedings of the 2004 International Conference on Shape Modeling Applications (SMI'04), Genova, Italy*, pp. 191–199.

Stulp, F. and Fisher, R. (2001). Reconstruction of surfaces behind occlusions in range images, *Proceedings of the 3rd International Conference on 3-D Digital Imaging and Modeling (3DIM'01), Quebec City, Canada*, pp. 232–239.

Vichitvejpaisal, P. and Kanongchaiyos, P. (2014). Surface completion using Laplacian transform, *Engineering Journal* **18**(1): 129–144.

Wang, J. and Oliveira, M. (2007). Filling holes on locally smooth surfaces reconstructed from point clouds, *Image and Vision Computing* **25**(1): 103–113.

Wang, L.-C. and Hung, Y.-C. (2012). Hole filling of triangular mesh segments using systematic grey prediction, *Computer-Aided Design* **44**(12): 1182–1189.

Wang, X., Cao, J., Liu, X. and Li, B. (2011). Advancing front method in triangular meshes hole-filling application, *Journal of Computer-Aided Design and Computer Graphics* **23**(6): 1048–1054.

Wang, X., Liu, X., Lu, L., Li, B., Cao, J., Yin, B. and Shi, X. (2012). Automatic hole-filling of CAD models with feature-preserving, *Computers & Graphics* **36**(2): 101–110.

Wei, L.-Y., Lefebvre, S., Kwatra, V. and Turk, G. (2009). State of the art in example-based texture synthesis, *Eurographics 2009, Munich, Germany*, pp. 1–25.

Wei, M., Wu, J. and Pang, M. (2010). An integrated approach to filling holes in meshes, *Proceedings of the 2010 International Conference on Artificial Intelligence and Computational Intelligence, Las Vegas, NV, USA*, pp. 306–310.

Wilkowski, A., Kornuta, T., Stefańczyk, M. and Kasprzak, W. (2016). Efficient generation of 3D surfel maps using RGB-D sensors, *International Journal of Applied Mathematics and Computer Science* **26**(1): 99–122, DOI: 10.1515/amcs-2016-0007.

Wu, X., Wang, M. and Han, B. (2008). An automatic hole-filling algorithm for polygon meshes, *Journal of Computer-Aided Design and Applications* **5**(6): 889–899.

Zhao, M., Ma, L., Mao, Z. and Li, Z. (2006). Feature sensitive hole filling with crest lines, *in* L. Jiao *et al.* (Eds.), *Advances in Natural Computation*, Lecture Notes in Computer Science, Vol. 4222, Springer, Berlin/Heidelberg, pp. 660–663.

Zhao, W., Gao, S. and Lin, H. (2007). A robust hole-filling algorithm for triangular mesh, *The Visual Computer* **23**(12): 987–997.

**Emiliano Pérez** received the Ph.D. degree in software engineering and computer systems from Universidad Nacional de Educación a Distancia (UNED), Spain, in 2011. He is an adjunct professor at Universidad de Extremadura, Badajoz, Spain. He has worked as a researcher in 5 R&D projects and has made more than 20 contributions in conferences and journals. He focuses his research on 3D computer vision and virtual reality: pattern recognition, modeling, representation and reconstruction of 3D objects, 3D sensors applied on cultural heritage, and virtual simulation.

**Santiago Salamanca** received the M.Sc. degree in physics from Universidad Complutense de Madrid, Spain, in 1995 and the Ph.D. degree in industrial engineering from Universidad Nacional de Educación a Distancia (UNED), Madrid, in 2005. He has been an associate professor of systems engineering and automation since 2002 with Escuela de Ingenieras Industriales, Universidad de Extremadura, Badajoz, Spain. During this time, he has made more than 60 international technical contributions to prestigious journals and conferences/workshops. His research interests include pattern recognition, 3-D object modeling and representation, and 3-D sensors.

**Pilar Merchán** received the Ph.D. degree in industrial engineering from Universidad de Extremadura, Spain, in 2007. She started work as an assistant professor in 2000 at Universidad de Extremadura and has been an associate professor there since 2012. Her research is focused on the field of 3D computer vision: sensory systems for 3D vision, complex scenes segmentation and retrieval, 3D scene modeling and representation, and their application to cultural heritage. She has worked as a researcher in more than 20 R&D projects and has generated about 60 technical contributions in prestigious journals and conferences.

**Antonio Adán** is an associate professor (1990) at the University of Castilla–La Mancha (UCLM, Spain) with an accredited full professor level (2012). He is the leader of the 3D Visual Computing & Robotics Lab in UCLM. His research interests are in 3D object representation and recognition, 3D data processing, 3D sensors, automatic BIM with scanners and robot interaction in complex scenes. He has made more than 120 international technical contributions in prestigious journals and conferences.