Elsevier Editorial System(tm) for Graphical Models
Manuscript Draft

Manuscript Number:

Title: Biharmonic Fields and Mesh Completion

Article Type: SI: Geometric Modeling

Keywords: mesh repair; hole filling; bi-harmonic scalar fields; shape optimization

Corresponding Author: Dr. Pere Brunet, Dr.

Corresponding Author's Institution: Univ. Politècn. de Catalunya

First Author: Oscar Argudo

Order of Authors: Oscar Argudo; Pere Brunet, Dr.; Antoni Chica; Alvar Vinacua

Abstract: We discuss bi-harmonic fields which approximate signed distancefields. We conclude that the bi-harmonic field approximation can be apowerful tool for mesh completion in general and complex cases. Wepresent an adaptive, multigrid algorithm to extrapolate signeddistance fields. By defining a volume mask in a closed region boundingthe area that must be repaired, the algorithm computes a signeddistance field in well-defined regions and uses it as anover-determined boundary condition constraint for the biharmonic fieldcomputation in the remaining regions. We discuss this approximation in practical examples in the case of triangular meshes resulting from laser scan acquisitions which require massive hole repair. We conclude that the proposed algorithm is robust and general, being able to deal with complex topological cases.

Suggested Reviewers: Paolo Cignoni
p.cignoni@isti.cnr.it

Daniel Cohen-Or
dcor@tau.ac.il

Tao Ju
taoju@cse.wustl.edu

Peter Liepa
peter.liepa@gmail.com

Opposed Reviewers:

Biharmonic Fields and Mesh Completion

Oscar Argudo[a], Pere Brunet[a], Antoni Chica[a], Àlvar Vinacua[a]

[a] *Computer Science Department*
Universitat Politècnica de Catalunya

Abstract

We discuss bi-harmonic fields which approximate signed distance fields. We conclude that the bi-harmonic field approximation can be a powerful tool for mesh completion in general and complex cases. We present an adaptive, multigrid algorithm to extrapolate signed distance fields. By defining a volume mask in a closed region bounding the area that must be repaired, the algorithm computes a signed distance field in well-defined regions and uses it as an over-determined boundary condition constraint for the biharmonic field computation in the remaining regions. We discuss this approximation in practical examples in the case of triangular meshes resulting from laser scan acquisitions which require massive hole repair. We conclude that the proposed algorithm is robust and general, being able to deal with complex topological cases.

Keywords: Model repair, Thin-plate energy, Volumetric methods

1. Introduction

Repairing mesh holes in very large geometric models is nowadays still a challenge. Mesh completion algorithms should be automatic, stable and robust, to cope with a huge number of unrepaired regions having unpredictable topologies.

The motivation of this paper came from a Cultural Heritage project aiming at the acquisition and 3D reconstruction of the entrance of the Ripoll Monastery in Spain. The monument is also known as the "Portalada", and is the main Romanic sculpture in Catalonia, dating back to the 12^{th} Century. The total size of the mesh after registration was in the vicinity of 173 Mtriangles, presenting 14622 holes of different sizes. More details on the project can be found in [1]. Holes and cracks were automatically detected through a search for cycles of border edges, and simple holes could be repaired by projecting their boundary onto a suitable plane and triangulating the projection without adding Steiner points. Classical mesh repair algorithms, however, were not able to handle large holes or holes with a complex boundary, as discussed later in the paper.

Email addresses: oargudo@cs.upc.edu (Oscar Argudo), pere@cs.upc.edu (Pere Brunet), achica@cs.upc.edu (Antoni Chica), alvar@cs.upc.edu (Àlvar Vinacua)

In this paper, we present a novel algorithm that solves and repairs complex hole configurations without any user intervention. Our approach is based on several ideas. We compute a suitable approximation of the signed distance field to the mesh in a volumetric data structure, in the vicinity of a hole. Other authors have used diffusion equations or other variational formulations to obtain smooth signed distance fields (except on a set of null-measure manifolds). We choose to find a 3D thin-plate solution of the bi-harmonic equation to guarantee C^1 continuity with the mesh around the repaired area. The repairing surface is then obtained as the zero-isosurface of the scalar bi-harmonic field. The algorithm starts by computing a volume mask which allows the computation of the signed distance field to the mesh around the hole in the domain of the mask, and uses this distance field to over-constrain the boundary conditions of the bi-harmonic equation. A discrete approximation of the bi-harmonic field is obtained by solving a quadratic optimization problem with a multigrid adaptive solver. One of the remarkable features of our approach is that it is able to successfully address complex topologies with disconnected mesh regions in a robust way. As far we know, this is the first automatic solution for hole repair in very large triangle meshes that provides a smooth blending at the boundaries.

Our approach was also inspired by [2]. However, as discussed in the next Section, they used the heat diffusion equation which results in a C^0 field. To reach C^1 continuity with the hole border regions, they included a heuristic step in each iteration by adding a heat source term after each diffusion step. We claim that in order to obtain general volume-based solutions for the mesh repair problem, the heat equation is not sufficient. Instead, we show that bi-harmonic fields do provide acceptable results.

The main contributions of the presented algorithm are:

- A volumetric mesh repair algorithm, based on the local computation of a bi-harmonic field approximation of the signed distance field. A key ingredient in the solution is the pre-computation of a volume mask to delimit a region where the signed distance is well-defined, and can be used to over-constrain the boundary conditions of the bi-harmonic equation overt the whole domain.

- A discrete solution of the bi-harmonic equation based on a quadratic optimization, with a multi-grid adaptive repair algorithm.

- A fully automatic algorithm for massive hole repair in very large triangle meshes, including the detection of mesh areas with holes, their repair and post-processing.

- The ability to handle complex hole topologies having islands and general shapes.

The next two Sections present an overview of the prior art and of our proposed algorithm. Section 4 details the generation of the initial distance field

and our bi-harmonic field approximation, while Section 5 discusses the acceleration and adaptive solutions that have been implemented. The last two Sections present and discuss the results and detail the conclusions of the paper.

2. Previous work

The need to repair surface models has always been present regardless of their source. Since the general repair problem is inherently ill-posed (due to ambiguities), production in this subfield has been ongoing. The first algorithms developed to address this challenge attempted to characterize all possible defects (gaps, holes, self-intersections, degeneracies, loss of sharp features, noise, topology problems, ...). Various surveys detail these problems and many of the contributions that have been proposed to solve them until now [3, 4].

Partly because of this divide and conquer approach, partly because of the lack of the computing resources needed for alternative strategies, the first solutions focused on repairing meshes directly. Liepa [5] proposed a step-based algorithm in which the holes were identified, triangulated, the resulting mesh patch refined, and finally, smoothed. More recently Bac et al [6] introduced a similar method where refining the mesh and minimizing bending energy were alternated. Both methods share their inability to deal with holes with "islands".

Another group of approaches to the problem used a different approach. Solutions known as volumetric methods start from a model to be repaired (a mesh or point cloud) and transform it into a volume where multiple defects can be treated in a common way. Volumetric methods guarantee surfaces free of self-intersections. The main differences within this group of methods come from the process used to transform the input model into a volume and the algorithm applied to complete the repair if the first step is insufficient.

Some of these techniques build an octree to represent the surface and use it to help in the repair of the entry model. Ju in [7] computes signs for the vertices of the octree nodes consistently with the intersections between the triangles of the mesh and the edges of the octree. These signs serve later to extract an isosurface without defects, but that only approximates the original mesh. Instead, in [8] the user provides a maximum tolerance error for the corresponding octree construction ensuring that the final mesh does not deviate beyond this distance. This approach excels at repairing CAD models, but is not applicable to models resulting from a scan, since the holes can be hundreds of times greater than the average edge size.

Other techniques compute a signed distance field. Davis et al [2] calculate an initial signed distance field near the surface, over which they alternate the application of a low-pass filter with a composition operator. This solution resembles a simulation of heat diffusion. The patches produced interpolate the edges of the hole adequately but they do not achieve normal continuity. Nooruddin et al [9] throw rays in multiple directions counting the number of intersections with the model. Using the parity rule on the resulting data they compose a distance field that represents the repaired model, but they require that all defects are treated simultaneously which may be inefficient for larger objects. Other

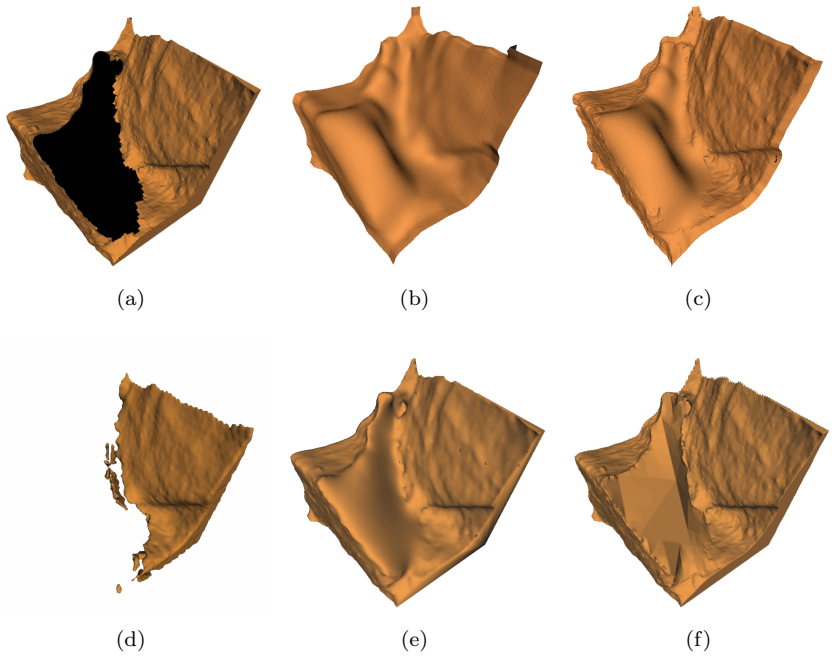


Figure 1: The results of some previous algorithms on one of our test cases. a) shows the original hole to fix, b) the iso-surface computed by our algorithm, and c) the final mended mesh we obtain. Figure d) the result of Nooruddin and Turk [9], e) the result of Liepa [5], and f) the result of Ju [7]. Notice that these last three do not achieve C^1 continuity at the boundary.

authors attempt to calculate an initial field, extending the distance values obtained using some interpolation technique, and the help of an expert user in the case of complex shapes. In particular, Masuda [10] used quadratic functions to perform this extension of the distance field. Brunet et al [11] also use an iterative algorithm that alternates smoothing steps –based on B-splines– and fitting steps, but their method requires an initial triangulation of the detected holes, which could adversely affect the outcome.

These previous art often focuses on closed meshes with some imperfections, and therefore try to complete all boundaries. Thus, a direct comparison with our framework is not possible. In order to compare them, we have completed one of our sample holes with a portion of its bounding box, and run those algorithms on the resulting closed —but defective— mesh. The results are shown in Figure 1. In this example, Nooruddin and Turk [9] delete a portion of the original mesh (Figure 1(d)), Liepa [5] does not blend smoothly at the boundary (Figure 1(e)), and Ju [7] produces triangles that are too large (Figure 1(f)), and all three solutions join the boundary with only C^0 continuity.

There are also algorithms that get rid of the requirement of using signed distances in order to handle point clouds directly. Hornung et al [12] pro-

pose immersing the point data in a voxelization so an unsigned distance field can be computed. The resulting distance values can be interpreted as inverse-likelihoods of the voxel being visited by the surface. Feeding this information into a min-cut algorithm yields a classification of the voxels' faces. This algorithm has the potential of producing a manifold boundary from the vertices alone, but does not respect the initial data points. Instead, it resamples the data at the frequency of the highest resolution used.

As the above methods have a tendency to use smooth patches to fill holes surrounded by complex patterns, another set of proposals attempt to reproduce the original shape of the model –in a plausible way–. One possibility is to apply principles similar to those used in texture synthesis and fill the holes with pieces of geometry that fit in from the model itself. Sharf et al [13] use local implicit approximations in order to measure the similarity between two patches. These patches are put in place using first a rigid transformation followed by an iterative closest point procedure with non-rigid transformations. The underlying assumption in all of these methods is that an appropriate shape continuation can be inferred from the acquired sample points only, using generic smoothness or self-similarity priors for missing parts of the model. Kraevoy et al [14] use instead template-based completion techniques. Their method computes a mapping between the incomplete input mesh and the template to correctly close gaps and holes.

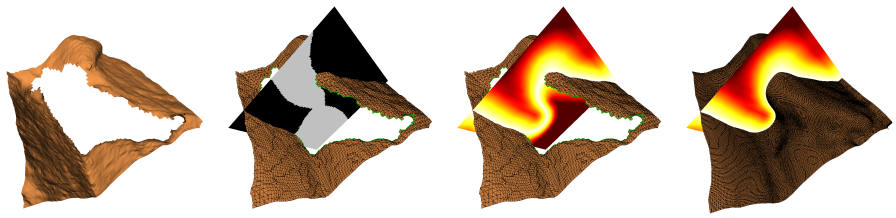
An advantage of our algorithm is the fact that we can minimize the distance from the computed to the original mesh. This allows us to use pieces of the resulting isosurface to close the holes (as it is done in [11]), while preserving the original geometry.

3. Overview

Our approach is aimed at repairing and completing very complex meshes in a robust and automatic way. As mentioned, our work was initially motivated by the case presented in [1], where a mesh of 173 million triangles with more than 14,000 holes had to be repaired. Cases from real applications like [1] with very large acquired meshes require fully automatic and reliable algorithms. In our opinion, volumetric schemes are well suited to fulfill these requirements.

The input of the algorithm is an unrepaired triangle mesh. Figure 2 shows the steps of our algorithm on a sample hole in the mesh reported in [1]. The operation of the algorithm is fully automatic and requires no special parameters to be set by the user to achieve these results. Our algorithm starts by detecting all border edges (triangle edges which do not belong to any other triangle). Holes are identified as closed loops of border edges. Every hole is then associated with its extended bounding box B_H, which is obtained by extending its bounding box by 20% in each coordinate direction. By computing these extended bounding boxes, we ensure that a sufficient part of the mesh around the hole is included in B_H.

The next steps of the algorithm, detailed below, are then repeated for each detected hole. It may occur that the extended bounding box B_H of a certain



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

Figure 2: Steps of our algorithm, illustrated with an example hole. The leftmost image shows the input mesh; the second image shows (a slice of) the mask; the third shows (the same slice of) the bi-laplacian smoothed distance field, and finally the last image shows the resulting isosurface used to fill-in the missing portion of the model. The details are more easily perceived in the accompanying video.

hole H contains other small holes. These secondary holes are also repaired as a byproduct of the repair process of H.

- A uniform grid is defined in the volume B_H, with a resolution similar to the size of the triangles in this extended bounding box. We have experimentally observed that a resolution of 128^3 is usually acceptable, although in some cases the optimal resolution may be 64^3 (for small holes) or 256^3, in the case of big holes.

- For all B_H grid vertices, two distances are computed: the unsigned distance to the mesh triangles in B_H and the unsigned distance to the border edges of the hole H. This is detailed in Section 4.1.

- A mask defining a partition in the discrete volume B_H is computed. The basic idea behind the mask concept is to split the discrete volume B_H into two disjoint regions R_D and R_B. As shown in Section 4.1, R_D is a closed region where a signed distance field to the mesh around the hole can be computed, while R_B is the region closer to the hole, where we want to smoothly extend the distance computed inside R_D. In what follows, we define the mask as the characteristic function of R_D, Figure 2(b).

- A signed distance field to the mesh around the hole is computed for every mesh vertex in R_D, as detailed in Section 4.1.

- A discrete bi-harmonic field is computed by solving a constrained optimization problem, see Section 4.2. The field is defined in B_H and the algorithm works by optimizing its smoothness. A key ingredient of the scheme is the use of over-constrained bi-harmonic boundary conditions. A plausible and C^1 continuous solution is obtained by constraining all grid vertices in R_D to be as close as possible to the pre-computed signed distance field values. By over-constraining the boundary conditions of the bi-harmonic function, we obtain a stable and robust solution, as discussed in Section 6, (see Figure 2(c)).

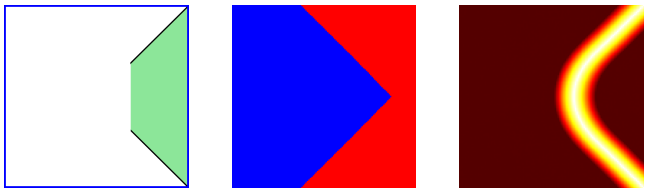


Figure 3: Given the geometry surrounding a hole (*left*) we compute a mask (*center*) by classifying a discrete volume's vertices into those closer to that geometry (R_D, colored in red) and those closer to the hole itself (R_B, colored in blue). This mask determines where to apply constraints.

- The computation of the bi-harmonic field is accelerated by using a multi-grid solver with adaptive refinement, as detailed in Section 5.
- Once the bi-harmonic field is computed, the hole is closed by blending the zero-isosurface of the field with the triangles in the vicinity of the border edges.

Observe that, unlike some previous approaches, our algorithm does not require an initial triangulation of the holes. The computation of the bi-harmonic field is driven by the mask M and the signed distance field in the region R_D.

4. Discrete bi-harmonic field

4.1. Initial distance field

We start by defining a uniform grid in the volume B_H of each mesh hole. The grid resolution is chosen to guarantee that cell sizes are of the order of the size of the triangles in B_H.

Then, a volume mask defining a region R_D in the discrete volume B_H is computed. Valid masks R_D are defined as a set of B_H grid vertices, with the property that all grid cells containing mesh triangles in B_H must be in R_D, and that the set of cells containing mesh triangles in B_H splits R_D in two disconnected regions R_D^+ and R_D^-: any connected path in R_D from any cell in R_D^+ to any cell in R_D^- mush go through a cell containing mesh triangles. Moreover, another desirable property that provides increased stability to the boundary conditions of the bi-harmonic equation is to have a mask with a high number of cells in R_D. R_D is the mask region, a closed region where the signed distance field to the mesh around the hole can be reliably computed, while R_B is the complement $B_H - R_D$. As discussed in Section 4.2, the signed distance field to the mesh in the mask region R_D works like an over-determined set of boundary conditions for the computation of the bi-harmonic field in the domain B_H. The computation of optimal masks for general mesh topologies in B_H is however beyond the scope of this paper and will be investigated as part of our future work.

In our present implementation, we use distance fields to compute the mask. We start by computing two unsigned distances for every B_H grid vertex: the unsigned distance to the mesh triangles in B_H and the unsigned distance to the border edges of the hole H. Then, the region R_D is defined as the set of B_H grid vertices such that their distance to the mesh triangles in B_H is lower than the distance to the border edges of the hole H. The complement of R_D in B_H is defined as the region R_B, the set of grid vertices such that their distance to the mesh triangles equals the distance to the border edges of the hole. Figure 3 shows the mask and the regions R_D and R_B in a simple 2D example. Both unsigned distance fields are computed by using a priority queue while iteratively propagating distances from cells containing mesh triangles to neighbor cells.

A signed distance field to the mesh around the hole can now be computed for every mesh vertex in R_D by simply using the sub-regions R_D^+ and R_D^- to assign coherent signs to the already computed distance field to the mesh.

4.2. Biharmonic field approximation

Our approach solves the mesh completion problem in each B_H by computing a discrete bi-harmonic approximation to the signed distance field to the mesh.

Let O_H be the set of grid vertices in the two outer layers of the B_H grid. We use a discrete approximation of the bilaplacian for each grid vertex V in $B_H - O_H$, involving a $5 \times 5 \times 5$ kernel. This kernel is a straightforward 3D extension of the well-known 2D discrete bilaplacian on uniform grids expressed as a convolution by matrix:

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & -8 & 2 & 0 \\ 1 & -8 & 20 & -8 & 1 \\ 0 & 2 & -8 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

To ensure a smooth behavior of the field on the boundaries of B_H, we also include a laplacian optimization term for every grid vertex in O_H. In short, we have laplacian terms for grid vertices in the two outer layers of the B_H grid and bilaplacian terms for the rest of grid vertices.

The second part of the optimization function includes the constraints or boundary conditions. For every grid vertex V in R_D, we impose that its final value $F(V)$ should be equal to the value of the already computed signed distance field $D(V)$ at this point.

In other words, instead of trying to find the field $F(V)$ by solving the bilaplacian equation $\Delta^2(F(V)) = 0$ for all grid vertices in B_H, we solve a constrained optimization problem by computing:

$$\underset{F}{argmin} \sum (\Delta^2 F(V))^2$$

for all V in B_H, with the constraint $F(V) = D(V)$ for all V in R_D.

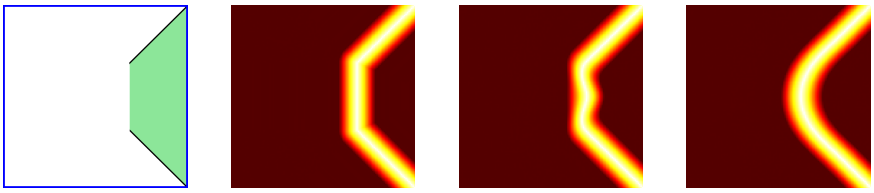


Figure 4: We can close a hole directly through triangulation (*left*) and use it to create an initial distance field (*center left*) that we can use as an initial estimate of an iterative solver. This method risks getting trapped in non-optimal configurations (*center right*). However, using a direct method provides the expected result (*right*).

To avoid potential instabilities produced by mesh noise in the vicinity of the holes which can be visible in the $D(V)$ field, we convert the constrained optimization into an unconstrained optimization problem:

$$\underset{F}{argmin}(\lambda \sum_{V \in B_H - O_H} (\Delta^2 F(V))^2) + \lambda \sum_{V \in O_H} \Delta(F(V))^2 + (1-\lambda) \sum_{V' \in R_D} (F(V') - D(V'))^2$$

This optimization problem is quadratic, with a well-defined solution involving a linear system of equations. We have observed that the solution of this unconstrained problem is more robust and stable.

An alternative could be to provide an initial estimate of the distance field obtained from triangulating detected holes and then use for instance the technique presented in [5]. This mechanism is not good enough, as illustrated in Figure 4 with the voxelization of a 2D hole, and in Figure 1(f) for a 3D example.

As also discussed in Section 6, this bi-harmonic solution is well behaved even for complex topological configurations like holes containing islands of groups of mesh triangles and unrepaired areas with complex geometries.

5. Performance

The system set forth in the previous section is sufficient to successfully solve the formulated problem, but even for small resolutions (32^3, 64^3) it is too expensive. Since we may find large holes or complex shapes that require higher resolutions, we need to improve its efficiency. In this section we present how to extend the basic implementation using an adaptive multiresolution scheme.

5.1. Multiresolution approach

To address the high cost of solving the complete system at higher resolution, we first obtain a good estimate of field distances at a lower resolution (16^3 or 32^3, depending on the complexity and size of the hole) through a direct solver and then use the resulting solution to construct an initial guess of an iterative solver at higher resolutions.

If the size of the voxelization that we want to apply to the solution of the problem is N^3, we choose a resolution M^3 where $M < N$, and down-sample by discarding values. No interpolation mechanism is used during this step which helps to facilitate the construction of the corresponding mask of size M^3 because, as described in the previous section, the mask can only take one of two values (in/out R_D). It is also important to note that the result of this decreased resolution system will be used as the initial value of the iterative algorithm, not as the final solution.

After solving the system of size M^3 using a direct solver (Cholesky in our implementation), we need to apply a process of up-sampling to obtain a distance field of size N^3. For this step we could apply many interpolation schemes. In our current implementation we have tested both trilinear interpolation and more sophisticated Bspline-based interpolations. Using splines gives a smoother estimate and, in many cases, closer to the desired solution, but the cost is substantial when compared with other steps (for an example, see Section 6, Table 1). We have therefore desisted from using any more precise interpolation scheme, as the increase in precision is vastly outweighed by the time needed to reach the result.

Moreover, the main advantage of applying more powerful methods of interpolation (such as B-splines) is in the fact that the initial guess provided to the iterative solver will be closer to the final solution. However, we can apply the above process into smaller steps so that the up-sampling has to calculate fewer values and deviates less from the global minimum. In our implementation, we start with the system of size M^3, solve it using a direct solver, apply up-sampling to a size of $(2M)^3$, and apply an iterative solver (Conjugate Gradient in our case). Repeating the last two steps we eventually reach the desired resolution N^3. Mainly, this reduces the number of iterations required at each multiresolution level for the chosen iterative solver to do its job. Furthermore, as each step of up-sampling is much smaller, a trilinear interpolation is good enough as shown in Section 6. In all the cases we have tested time gained by using trilinear interpolation exceeded what was obtained by the reduction of the number of iterations, while the accuracy of the resulting solutions was largely unaffected.

5.2. Adaptive refinement

To further improve efficiency we have made the process adaptive. That is, we only compute new values for nodes in each step when they are sufficiently close to the surface. In particular, for each step of the multi-resolution process we extract the discrete band DB of the current voxelization (set of voxels containing the zero isosurface), and we enlarge it a distance δ (measured in voxels) by a dilation operator, obtaining EDB. When assembling the linear system described in Section 4, only nodes inside EDB are considered to be unknowns. The rest are considered constant, eliminating from the system all equations that end with no unknowns. Finally, before performing the upsampling needed to continue with the next multiresolution step, we replace the values of the unknowns into the

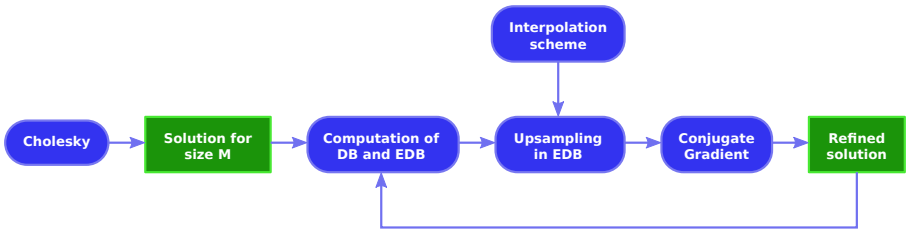


Figure 5: Outline of the adaptive multiresolution strategy used to generate the results presented in Section 6.

distance field, preserving the values outside EDB. This process is outlined in Figure 5

The distance δ we use to dilate DB determines how much we gain in terms of efficiency with the adaptive method, but too small a number will cause the surface to be unable to move to voxels outside of EDB. In all our tests we have found a distance of $\delta = 1$ suffices to produce satisfactory results. Using a factor of 2 between the sizes of each multiresolution level ensures that the final surface will not be found more than one voxel away from the solution of the previous step.

6. Results

We have tested our algorithm with different interpolation schemes to compute the up-sampling in EDB (see Figure 5), in order to determine the best compromise between efficiency and quality of the results, as mentioned in the previous section. All times reported in this section have been measured on a PC with a Core i7 CPU at 3.20GHz, with 12GB of memory, and at 64 bits. The GPU is a GeForce GTX 570, but it was only used for rendering and does not impact the times reported.

As a typical example, Table 1 shows the time spent and the approximation

Interpolation Method	RMS Bilapl.	RMS E3	time (sec)
Trilinear	0.00099643	0.00583506	23.114
B-spline	0.00020538	0.00537014	710.151

Table 1: Comparison of grid refinement with interpolating uniform B-splines and with trilinear interpolation.

error achieved when interpolating the higher resolution grid with simple trilinear interpolation, and when using an interpolating uniform b-spline. The columns list the root mean square error in the solution of the bi-laplacian equations, the root mean square distance from the resulting mesh to the ring-three around the hole, and the time in seconds of the whole computation. As expected, the

better interpolation yields smaller errors, but in all cases we have tested, this improvement seems too little for the huge increase in computation time. We think the results achieved through trilinear interpolation are of high quality already, and therefore it is not sensible to spend the time needed to use a more precise interpolation scheme when raising the resolution of the grid.

Figure 6 shows four examples of holes from the scanning of the "Portalada", and the results of our hole-filling algorithm. Notice how holes with intricate shapes and inner islands (Figure 6(j)) are handled properly.

A comment is in order regarding the measure of geometric precision in the data above. In dealing with meshes produced by 3D scanners, we've found that —because holes originate in regions the scanner cannot reach— points on the boundary of the holes are usually more noisy than those slightly further away. Consequently, we measure the fitting of the new surface not right on the edge of the hole, but on vertices that lie at an edge-distance of 3 from the hole. The difference is witnessed by the data in Table 2, that lists —for the holes in Figure 6— the error on the edge of each hole (RMS E0) and on the vertices of its 3-ring (RMS E3), both before smoothing the distance field and after completing the algorithm. The table shows that both errors are similar in magnitude, so

Test Case	RMS bi-lapl.	RMS E0	RMS E3
hole a (beg)	6.86325	0.00488603	0.00434737
hole a (end)	0.00248534	0.0161777	0.0128677
hole d (beg)	18.4518	0.00573393	0.00505522
hole d (end)	0.00537352	0.0161812	0.0134089
hole g (beg)	9.66015	0.00411624	0.00373306
hole g (end)	0.00099643	0.0268693	0.00581358
hole j (beg)	8.40509	0.00433601	0.00342995
hole j (end)	0.00254796	0.032138	0.00934204

Table 2: Errors for the test cases in Figure 6.

using RMS E3 does not influence the quality of the result, while it does provide a more stable measure.

Table 3 shows the time it takes to complete our algorithm for the test holes in Figure 6. We list the time it took to smooth the distance field, along with the

Test case	t. smooth	t. total	unknowns 32^3	unknowns 64^3	unknowns 128^3
hole a	11.971	12.392	20.49%	10.07%	5.05%
hole d	12.19	13.01	21.16%	10.43%	5.21%
hole g	21.33	23.42	35.43%	17.80%	8.91%
hole j	17.72	19.27	28.78%	14.73%	7.43%

Table 3: Run-times for filling the sample holes shown in Figure 6

total time of up-sampling and smoothing, using the multigrid approach. The

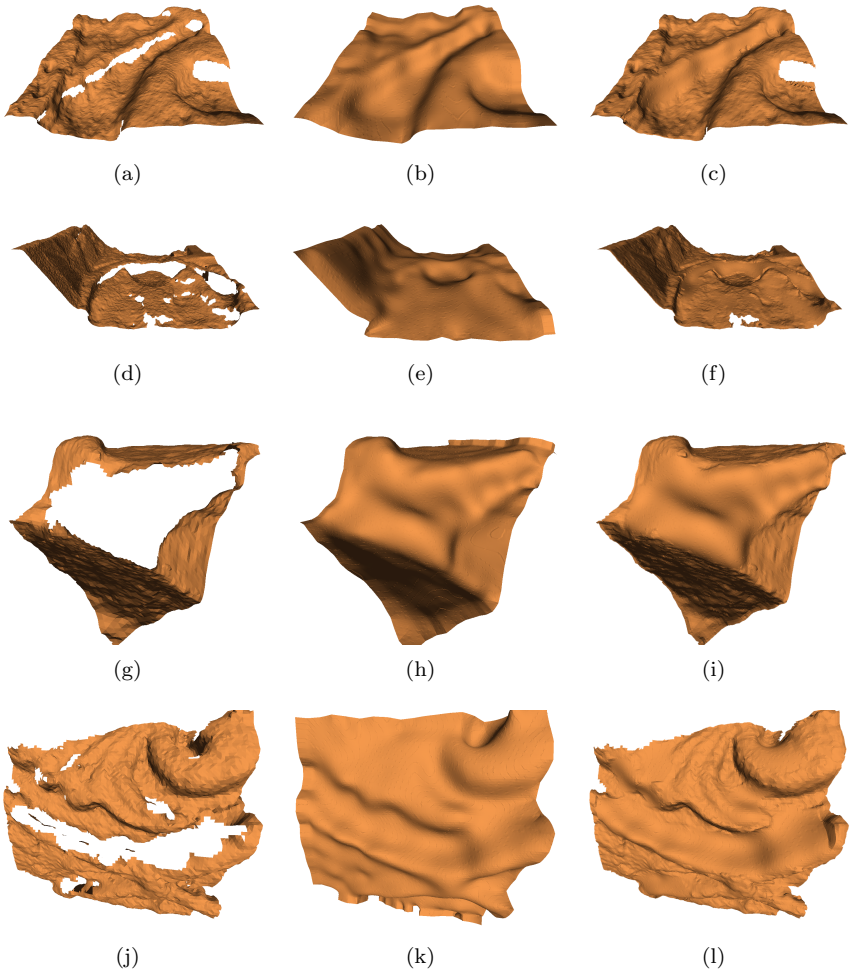


Figure 6: Four cases of holes in the mesh from [1]. The left column shows the scanned meshes, while the middle column shows the iso-surface computed by our algorithm. The rightmost column shows the final mended mesh; notice that all holes not abutting the external boundary are fixed.

last three columns of this table indicate the proportion of relevant unknowns using the multigrid approach with respect to the total number of unknowns in the full system.

In order to facilitate the verification of our results, and hoping that it will promote adoption of the technique, we have made our code available at `https://github.com/achicac/BiharmonicRepair`.

7. Conclusions

We have presented an algorithm to automatically repair complex hole configurations without any user intervention. Our approach is based on several ingredients. We solve a volumetric problem in the vicinity of each mesh hole to compute a suitable approximation of the signed distance field. The algorithm is based on computing a 3D thin-plate solution of the bi-harmonic equation that guarantees C^1 continuity with the previous mesh around the repaired area, and obtains the repairing surface by computing the zero-isosurface of the scalar bi-harmonic field.

A key ingredient of the algorithm is a volume mask such that the signed distance field to the mesh around the hole can be reliably computed in the mask domain. The rest of the algorithm uses this distance field to over-constrain the boundary conditions of the bi-harmonic equation over all of B_H. A discrete approximation of the bi-harmonic field is obtained by solving a quadratic optimization problem, with a multigrid adaptive solver.

Unlike some previous approaches, our algorithm does not require an initial triangulation of the holes. The computation of the bi-harmonic field is driven by the mask and the signed distance field in its domain.

Our approach is able to successfully address complex topologies with disconnected mesh regions in a robust way. As far as we know, this is the first automatic solution for hole repair in very large triangle meshes that provides a smooth blending at the boundaries.

In the near future we plan to include an option with the possibility to extrapolate the roughness of the mesh around the hole to its smooth bi-harmonic completion by statistically synthesizing similar details. This would be helpful in practical cases where we want to conceal the repaired portion and make it pass as original. We also plan to explore ways to find more general, optimal masks.

Acknowledgements

Argudo, Brunet, Chica and Vinacua's research on this project has been supported by the Spanish Ministerio de Economia y Competitividad (MINECO) under Grant TIN2013-47137-C2-1-P

Argudo's Research has also been supported by a Graduate Research Fellowship from the Spanish Government.

References

[1] M. Callieri, A. Chica, M. Dellepiane, I. Besora, M. Corsini, J. Moyés, G. Ranzuglia, R. Scopigno, P. Brunet, Multiscale acquisition and presentation of very large artifacts: The case of portalada, ACM Journal on Computing and Cultural Heritage 3 (4) (2011) 14:1 – 14:20.

[2] J. Davis, S. R. Marschner, M. Garr, M. Levoy, Filling holes in complex surfaces using volumetric diffusion, in: 3DPVT, IEEE Computer Society, 2002, pp. 428–438.

[3] T. Ju, Fixing geometric errors on polygonal models: A survey, J. Comput. Sci. Technol. 24 (1) (2009) 19–29.

[4] M. Attene, M. Campen, L. Kobbelt, Polygon mesh repairing: An application perspective, ACM Comput. Surv. 45 (2) (2013) 15:1–15:33.

[5] P. Liepa, Filling holes in meshes, in: Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, SGP '03, Eurographics Association, 2003, pp. 200–205.

[6] A. Bac, N.-V. Tran, M. Daniel, A multistep approach to restoration of locally undersampled meshes, in: Proceedings of the 5th International Conference on Advances in Geometric Modeling and Processing, GMP'08, Springer-Verlag, 2008, pp. 272–289.

[7] T. Ju, Robust repair of polygonal models, in: ACM SIGGRAPH 2004 Papers, SIGGRAPH '04, ACM, 2004, pp. 888–895.

[8] S. Bischoff, L. Kobbelt, Structure preserving cad model repair., Comput. Graph. Forum 24 (3) (2005) 527–536.

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

[10] T. Masuda, Filling the signed distance field by fitting local quadrics, in: Proceedings of the 3D Data Processing, Visualization, and Transmission, 2Nd International Symposium, IEEE Computer Society, 2004, pp. 1003–1010.

[11] P. Brunet, A. Chica, I. Navazo, A. Vinacua, Massive mesh hole repair minimizing user intervention, Computing 86 (2-3) (2009) 101–115.

[12] A. Hornung, L. Kobbelt, Robust reconstruction of watertight 3d models from non-uniformly sampled point clouds without normal information, in: Proceedings of the Fourth Eurographics Symposium on Geometry Processing, SGP '06, Eurographics Association, 2006, pp. 41–50.

[13] A. Sharf, M. Alexa, D. Cohen-Or, Context-based surface completion, in: ACM SIGGRAPH 2004 Papers, ACM, 2004, pp. 878–887.

[14] V. Kraevoy, A. Sheffer, Template-based mesh completion, in: Proceedings of the Third Eurographics Symposium on Geometry Processing, Eurographics Association, 2005, pp. 13–22.

Supplemental Material (VIDEO)

Click here to download Supplemental Material (VIDEO): video.mp4