EUROGRAPHICS Workshop on Graphics and Cultural Heritage (2016)
C. E. Catalano and L. De Luca (Editors)

A Soft Union based Method for Virtual Restoration and 3D Printing of Cultural Heritage Objects

R. Gregor[1] †, P. Mavridis[1] ‡, A. Wiltsche[2] § and T. Schreck[1] ¶

[1]Graz University of Technology, Institute of Computer Graphics and Knowledge Visualization, Austria
[2]Graz University of Technology, Institut für Architektur und Medien, Austria

Abstract

Recent improvements in 3D acquisition and shape processing methods lead to increased digitization of 3D Cultural Heritage (CH) objects. Beyond the mere digital archival of CH artifacts, there is an emerging research area dedicated to digital restoration of 3D Cultural Heritage artifacts. In particular several methods have been published recently that, from a digitized set of fragments, enable their reassembly or even the synthesis of missing or eroded parts. Usually the result of such methods is a set of aligned but disconnected parts. However, it is often desirable to produce a single, watertight mesh that can be easily 3D printed. We propose a method based on a volumetric soft union operation that can be used to combine such sets of aligned fragments to a single manifold mesh while producing smooth and plausible geometry at the seams. We assess its visual quality and efficiency in comparison to an adaption of the well-known Poisson Reconstruction method. Finally, we provide practical insights on printing the results produced by our method on digitized fragments of real CH objects.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems

1. Introduction

In this paper, we address the problem of creating a single manifold mesh from a set of parts with plausible seams at the transitions. The input set is structured as a set of spatially close and possibly also intersecting meshes. While our proposed method is not strictly limited to, it is especially well suited for the virtual restoration of Cultural Heritage objects.

1.1. Virtual Restoration of 3D CH Objects

Recent improvements in 3D acquisition and shape processing methods lead to increased digitization of 3D CH Objects. Beyond the mere digital archival of CH artifacts, there is an emerging research area dedicated to digital restoration of 3D Cultural Heritage artifacts. In particular several methods have been published recently, that, from a digitized set of fragments, enable their reassembly or even the synthesis of completely missing parts. Such methods are very useful to generate restoration hypotheses more efficiently than before while simultaneously avoiding the risk of additional damage to precious 3D CH Objects and also eliminating the need for physical access to the real fragments.

For example Palmas et al. [PPCS13] propose a computer assisted system for virtual reassembly of CH objects, that can be operated interactively by a CH domain expert. Methods such as e.g. [PKT01, HFG*06, MAP15] rely on automated matching of various shape features to automate reassembly tasks even further.

Complementary, there are methods that try to generate missing parts by exploiting self-similarity of CH objects such as in [SGS14, MSAP15] where approximated global symmetry-planes or -axes are used to synthesize new geometry. Another strategy outlined in [GSP*14] is to find and transfer similar geometry from an external shape repository. Alternatively, and not related to the CH domain, in [LF08] user-provided sketches are used to find, align and merge suitable shapes from an external repository as new parts are always aligned to produce intersections.

Usually, the result of such reassembly and completion methods is a set of aligned but disconnected parts with either remaining small gaps at the fracture lines or self intersections from transferred geometry. Although Lee et al. [LF08] stitch together disconnected meshes at their intersections, they neither aim to fill gaps between parts nor to produce non-trivial geometry at the transitions.

When the task is to restore CH objects in order to generate a plausible hypothesis of the intact shape, it is often desirable to not

† gregor@tugraz.at
‡ p.mavridis@cgv.tugraz.at
§ wiltsche@tugraz.at
¶ tobias.schreck@cgv.tugrias.at

© 2016 The Author(s)
Eurographics Proceedings © 2016 The Eurographics Association.

only focus on defects of the mesh structure itself but to restore and avoid the introduction of as many defects within the actual geometry itself.

Obviously, manually modeling more plausible geometry at the seams between different parts of a digitized object can be an extremely time-consuming process that does not scale well with the ever-growing number of digitized CH objects and the increasing level of detail produced by modern 3D scanning hardware.

On the other hand, it seems natural to directly use the results of virtual CH object restoration for digital fabrication techniques such as 3D printing that are gaining momentum in the CH domain.

Here, however, we can identify two simple but important practical issues: First, as the results of reassembly methods are usually still disconnected parts, 3D printing them demands for a subsequent manual reassembly after the printing process. Obviously in this context, the practical utility of virtual reassembly methods is strongly affected. Second, when partially restored fragments, e.g. generated by above-mentioned completion methods, intersect each other geometrically, their 3D printed counterparts can not be reassembled correctly. Again this clearly impairs the potential utility of above-mentioned virtual restoration methods for digital fabrication. Our method can be readily used to circumvent both issues. As a secondary benefit, the printing process itself is simplified. Practically, printing a single solid object instead of multiple fragments reduces consumption of both time and printing material, demands for less manual intervention during printing and also yields a model with increased physical solidity.

1.2. Contribution

In this paper, we propose a robust and efficient method based on a volumetric soft union operation that produces a single watertight mesh from disconnected and also possibly intersecting fragments of a CH object. A single intuitive parameter can be used to steer the appearance of the geometry generated at the seams in between the fragments. In this way, our method can produce both geometrically plausible and clearly distinguishable transitions between fragments. The latter might be desired to preserve information about the location of the actual fracture lines.

Using digitizations of real defective CH objects, we provide experimental measurements for the effectiveness and efficiency of our method, assess the quality of the generated geometry by means of visual inspection and compare our method to a more traditional approach that first performs a parametrized vertex filtering step and subsequently applies the well-known *Poisson reconstruction*. As a proof-of-concept, we provide practical insights on the actual 3D printing of various models generated by our proposed method.

2. Related Work and Alternative Approaches

In this section we provide an overview of related work, in subsection 2.3 we describe our adaption of the well-known Poisson Reconstruction for our problem by combining it with a vertex filtering step. This method will be used for comparison with our Soft Union based method.

2.1. Gap Removal in Voxel Grids

In image processing, a classic and very widespread approach is to fill in gaps between nearby segments by applying a series of morphological closing operations on a 2D pixel grid. For 3D voxel grids, morphological operations have rarely seen adoption. Very recently, Sobiecki et al. [SJT16] compare various morphological closing operations to their proposed method for gap removal on binary volumes and easily outperform them in terms of visual quality. Common to all approaches that directly operate on voxel grids, such as binary volumes is, that the inevitable aliasing when converting from an input mesh could significantly impact small-scale texture of highly detailed CH objects. To limit this effect, the resolution of such a voxel grid could be increased. However, memory consumption (and required computation time) will grow cubically with the resolution which does not allow for a promising perspective on the scaling behavior of such methods. In contrast, our method, albeit being volumetric, circumvents the issue by relying on a distance field representation, that does not suffer from this limitation and is known to preserve more detail when using comparable resolution. Also, our method does not rely on the extraction of a curve skeleton as in [SJT16], which might be problematic for some data sets. For example, it is easy to observe that fragments of a solid, coherent CH object such as used in our experiments would often yield a skeleton that does not match well with the skeletons of adjacent fragments.

2.2. Classical Mesh Repair Methods

Local mesh repair methods [ACK13] are designed to detect and repair defects (such as gaps and holes) within the structure of a *single* mesh. While their scope of application might appear to intersect with our problem statement, they are intended to operate on a more low level class of defects. While they provide the benefit of not changing areas where no defects were detected and thus perfectly preserve the input geometry for large portions of the surface, they are simply unable to detect the areas were gaps between adjacent fragments should be closed. They are not designed to discard geometry that is related to defective areas of the input parts (i.e. breaking edges in between adjacent fragments). Also such methods target specific classes of mesh defects individually and are prone to introduce new mesh defects. hence overall, it does not seem promising to adapt local mesh repair methods to target our problem as, more than likely, a chain of subsequent local repair operations would need to be included. However, this chain would be highly unlikely to be robust enough in many cases as defects introduced by individual local repair algorithms could possibly violate the input constraints of subsequent local repair algorithms.

More promising, global mesh repair methods [ACK13] aim to generate a new manifold mesh, that is close to the defective input mesh and free of defects. While they are prone to altering the geometry in intact areas of the input, many of them are guaranteed to produce manifold results and address multiple classes of mesh defects in a single run. Most of the state of the art global mesh repair methods convert the input mesh into a volumetric representation, as e.g., but not limited to voxel grids. Many of them are able to deal with self-intersections or disconnected components when the gaps between them are reasonably close. In the process, they use heuristics to obtain necessary but missing information for populating the volumetric representation of a single connected shape. When converting the volumetric representation to a mesh, some of them exploit the information from the input vertices of the mesh to reduce aliasing effects. However, in order to produce geometrically plausible results, they would still need to be adapted for our problem.

2.3. Filtering and Poisson Reconstruction

Closely related to the class of global mesh repair methods mentioned in the previous section, an area of application that vastly gained in importance over the recent decade is the reconstruction of meshes from point clouds obtained by 3D scans. When reconstructing a mesh from point clouds, usually multiple and potentially noisy point clouds (e.g., obtained by subsequent shots using different sensor positions) are registered and merged before attempting a mesh reconstruction on the unified set of points. Furthermore, they are designed for processing very detailed geometric information in an efficient manner to cope with the very high resolution of recent 3D scanning hardware. It is easy to observe that both of these aspects are highly related to our problem statement.

Improving over earlier methods such as Ball Pivoting [BMR*99], variations of the Poisson Reconstruction first proposed by Kazhdan et al. [KBH06] are considered to preserve more detail of the input geometry [Hop08] and have seen broad adoption for mesh reconstruction.

In first experiments, we found that, with suitable parameters, the implementation by Kazhdan bundled in Meshlab [CCC*08] was successful in producing a single watertight mesh from disconnected input parts but did not remove the fracture lines. Larger gaps could be closed by decreasing the octree depth parameter, leading to loss of texture. When trying to preserve detail, gaps will not concealed or closed reliably (see Figure 1).

As the Poisson reconstruction does not operate on the tessellation, we can address this by a simple heuristic that filters undesired vertices from the input prior to the reconstruction (see Algorithm 1). A vertex is removed vertex from a fragment if two conditions are met. First, there must be at least one vertex from another fragment closer than the average vertex spacing multiplied by a constant factor r. Second, the angle between the vertex-normal and the aggregated normals of its neighbors from a certain other fragment must be at least 90 degrees. The remaining vertices are passed on to a subsequent Poisson Reconstruction.

When using various versions of Kazhdan's implementation, we had issues with sporadically appearing, small disconnected components distant to any input geometry. Hence, we eventually chose the implementation [ASG16] provided as part of CGAL [CGA]. Based on the vertex location and their normals, both formulate and solve a Poisson Equation to approximate an indicator function in volumetric space. Here Kazhdan uses an adaptive octree to substantially improve efficiency over a standard voxel grid. In contrast, CGALs implementation computes a 3D Delauny Triangulation over the input vertices, refines it to remove non-isotropic tetrahedra and then solves the Poisson Equation on the refined tetrahedral triangulation to obtain the indicator function. Subsequently, CGAL extracts an isosurface (i.e. the reconstructed surface) by using a method described by Boissonnat et al. [BO05] whereas Kazhdan applied an adaptive Marching Cubes algorithm for isosurface extraction.

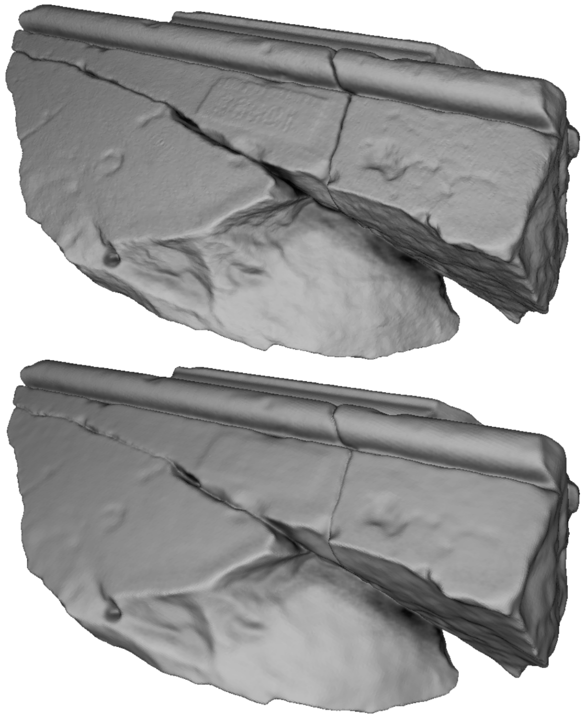

Figure 1: *Aligned input fragments and the result of an unfiltered Poisson reconstruction with octree depth of 10. While the parts can be merged into a single component, the fracture lines are not restored and texture of the surface is lost.*

3. Volumetric Soft Union Approach

In this section we describe a new volumetric approach for the creation of watertight meshes from a set of disconnected and also possibly intersecting parts, such as those resulting from virtual restoration methods (see Section 1.1). The method is able to close small gaps between parts due to missing material that might e.g., be caused by degradations such as erosion or missing parts. An overview of our approach is shown in Figure 2. The input fragments are first converted to a volumetric distance field representation, then a series of soft union operations are performed in order to accumulate the distance field of the final restored object and finally converted back to a mesh. In the remainder of this section, we describe each one of these steps in more detail.

3.1. Truncated SDF Data Structure

In our pipeline, every input fragment $\mathcal{Y}$ is represented by a distance function

$$\phi(\mathbf{x},\mathcal{Y}) = \underset{\mathbf{y}\in\mathcal{Y}}{min}\ \|\mathbf{x}-\mathbf{y}\|_2\,, \qquad (1)$$

© 2016 The Author(s)
Eurographics Proceedings © 2016 The Eurographics Association.

Algorithm 1: The Vertex Filtering step, r controls the radius for vertex filtering. The computed average vertex spacing s is used to steer parameters of the subsequent Poisson Reconstruction. Time complexity is $O(n\,log(n))$ where n resembles the number of vertices aggregated over all fragments.

```
Input: list of fragments $F_i$ each consisting of a list of vertices
       $v_{i,j}$ with their normals $n_{i,j}$ attached, filter radius
       parameter $r$
for each fragment $F_i$ do
    create Kd-Tree $T_i$ over all $v_{i,j} \in F_i$
    compute average vertex spacing $s_i$ over all $v_{i,j} \in F_i$
end
compute global average spacing $s := \frac{\sum_i (s_i * |F_i|)}{\sum_i |F_i|}$
for each fragment $F_i$ do
    for each vertex $v_{i,j} \in F_i$ do
        for each Kd-Tree $T_k$ where $k \neq i$ do
            find list of indices $L$ of $v_{k,l}$ in $T_k$ in radius $r * s$
              around $v_{i,j}$
            if $|L| > 0$ and $0 \geq n_{i,j} \cdot \sum_{l \in L} n_{k,l}$ then
                mark $v_{i,j}$ for removal
            end
        end
    end
end
for each Fragment $F_i$ do
    remove all marked vertices $v_{i,j}$from $F_i$
end
Output: Concatenation of all $F_i$, average vertex spacing $s$
```

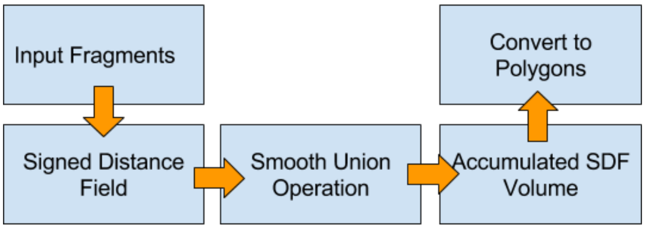

Figure 2: *An overview of our soft union based method for fracture line restoration: After the first fragment has been converted to a truncated signed distance field (SDF), each subsequent fragmented is accumulated into a joint distance field by computing the Soft Union operation. Finally, the Soft Union of all fragments is converted to a mesh.*

which encodes the distance of every point in space $\mathbf{x}$ to the surface $\mathcal{Y}$. This function, also called the *distance field* of surface $\mathcal{Y}$, is discretely sampled on a 3D grid that extends in spatial vicinity of the surface $\mathcal{Y}$ and is compactly stored using a sparse hierarchical volumetric data structure [Mus13], thus creating the *Truncated Signed Distance Field* (TSDF) representation of the surface. The hierarchical nature of this data structure allows encoding highly detailed surfaces without enormous memory requirements and also offers very fast and cache-coherent access to the encoded data.

3.2. The Soft Union Operation

After this conversion, we can process the input surfaces with simple mathematical functions that operate on distance spaces. For example we can compute a new distance function $\phi(\mathbf{x},\mathcal{U})$ that represents the union of two objects $\mathcal{Y}$ and $\mathcal{Z}$ as simply as:

$$\phi(\mathbf{x},\mathcal{U}) = min(\phi(\mathbf{x},\mathcal{Y}),\phi(\mathbf{x},\mathcal{Z})). \tag{2}$$

However, this hard union operation is not suitable for the creation of a single watertight object when there are gaps between the fragments, which is typically the case after the reassembly of damaged or eroded CH fragments.

To this end, we propose the use of a soft union operation, which is provided by the following mathematical formula:

$$\phi(\mathbf{x},\mathcal{U}) = min(\phi(\mathbf{x},\mathcal{Y}),\phi(\mathbf{x},\mathcal{Z})) - 0.25 \cdot \frac{g(x)^2}{r} \tag{3}$$

where

$$g(x) = max(r - |\phi(\mathbf{x},\mathcal{Y}) - \phi(\mathbf{x},\mathcal{Z})|),0) \tag{4}$$

and $r > 0$ is a user defined parameter that controls the *smoothness* of the union operation, as shown in Figure 3. When r is set to values close to 0, the second term of Equation 3 converges to 0 and the operation corresponds to the traditional *hard* union operation. Therefore, small values of the parameter r tend to leave visible gaps between the surfaces, while larger values properly fill the gaps and produce a smooth and continuous surface. A proper choice of value for this parameter depends on the global scaling of the input object as well as the extent to which gap closing is desired. Thus, this parameter can be adjusted by the user according to his preferences.

Equation 3 describes the soft union operation for two fragments. In the general case, for N fragments, we create the distance function $\phi(\mathbf{x},\mathcal{U})_N$ that describes their soft union with the following recursive equation:

$$\phi(\mathbf{x},\mathcal{U})_i = s\_union(\phi(\mathbf{x},\mathcal{U})_{i-1},\phi(\mathbf{x},\mathcal{Y}_i)), 0 < i \leq N \tag{5}$$

where $\phi(\mathbf{x},\mathcal{U})_0$ is the empty volume, $\mathcal{Y}_i$ the i-th fragment volume and *s_union* is computed using Equation 3.

In the previous equations we have assumed continuous distance functions in space. Since in our actual implementation the distance functions are discretized over a grid, Equation 3 is evaluated for the center of each voxel. Only voxels in close proximity to *both* surfaces will be modified. Our approach involves the evaluation of a simple mathematical function in volumetric space, without the need for any filtering or skeletonization operations, as required by the methods described in section 2.3 or [SJT16]. The truncation of the SDF vastly increases efficiency over a dense distance sampling

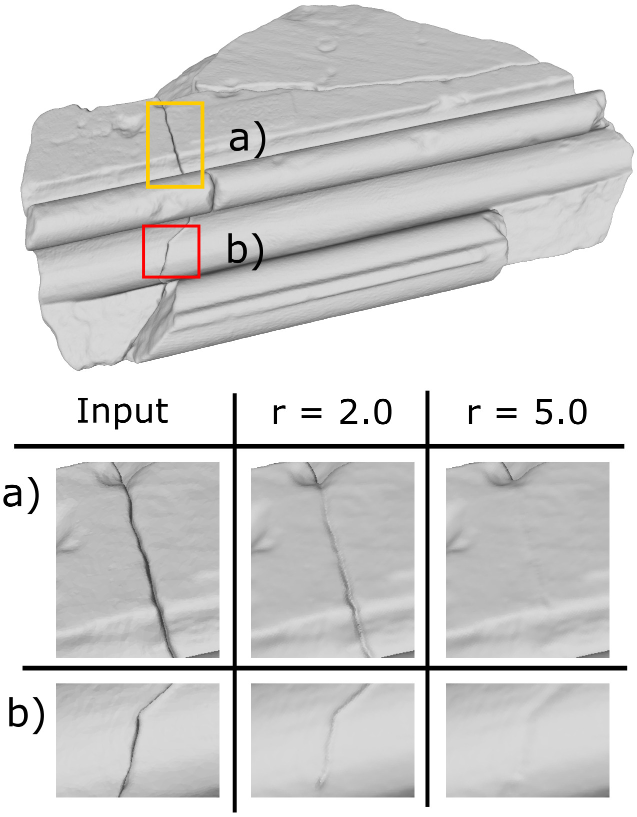

Figure 3: *The soft union operation for different values of the parameter r on the Dora block data set. Small values of the parameter leave visible gaps, while larger values progressively fill the gaps between the fragments.*

in volumetric space. Furthermore, all evaluations are entirely data parallel.

It might be interesting to note that the way the two fragments merge in our method when approaching each other resembles the merging of the Metaballs primitives [Bli82], well-known in computer graphics. However, the underlying data structures and mathematical formulation are completely different. A similar formulation with ours has been used recently in rendering applications [Eva15] but to our knowledge has never been used before in mesh processing.

3.3. Mesh Generation

After the computation of the final volume using Equation 5, our method generates the output mesh using the dual contouring [JLSW02] polygonization algorithm. The output mesh is guaranteed to be manifold. If the parameter r is set large enough, the result will always be a single connected component and it is directly suitable for 3D printing visualization, analysis or further processing purposes.

4. Evaluation

For our experiments, we used a collection of 8 digitized and partially reassembled and completed CH objects, based on the publicly available data from [PRE]. We chose fragment sets that vary in level of detail and number of fragments. Over the chosen fragment sets, the total number of vertices varies between 120,004 and 12.5 million, whereas the number of fragments per set varies between 2 and 8. All of the obtained results contained a single, successfully merged, manifold mesh. However the Filtering/Poisson approach was not able to produce a result within a runtime of 8 hours for the most detailed fragment set.

In the remainder of this section, we first show and compare the quality of our proposed method with our alternative combination of vertex filtering and the well-known Poisson Reconstruction. Second, we provide detailed results concerning the efficiency of both approaches before we, third, provide practical insights on 3D printing the results of our proposed Soft Union based method.

4.1. Visual Quality

On first glance, when examining the visual quality of the resulting meshes from both, our proposed as well as our alternative Filter/Poisson approach, it can be observed that both are able to merge the input fragment sets successfully (Figures 4 and 5). Where fracture lines have been removed, no implausible salient artifacts such as e.g., sharp, chamfered or aliased ridges or grooves have been introduced. However, where larger intersections of fragments are surrounded by two equally oriented, and near parallel surface patches, the Filtering/Poisson approach produces implausible geometry.

For comparing the resulting geometry with the input sets in more detail, we provide a visualization of the geometrical deviations in Figure 4. Two million vertices were uniformly sampled on the surface of each result before their spatial distance to the closest vertex of the input set was obtained. Then, for each vertex of a result, we computed an interpolated value of from the neighboring samples. We consider this a more accurate measure than computing Hausdorff distances for each vertex of the output to the vertices of the input directly, as it would be heavily biased towards the Soft Union results. For each result, it can be observed that the soft-union based method preserves small-scale details of intact areas much better than the Filtering/Poisson alternative, which is especially desirable when operating on high-resolution digitizations of CH objects.

4.2. Efficiency

To assess and compare the efficiency of both methods, we measured their execution time. Our measurements were performed using 64bit binaries on a laptop running a 64bit Arch Linux system featuring a Core i7-6700HQ 2.6GHz quad-core processor and 32GB of DDR4 RAM. Both implementations (also including CGAL 4.8 and OpenVDB 3.1.0) were compiled by the GNU C++ compiler version 6.1.1 using the flags: `-O3 -march=native`. For the provided measurements, we excluded any steps involving I/O activity which generally were at least a magnitude faster than the remaining steps. The timing measurements for the Filtering/Poisson variant are separated into the Filtering Step (see Algorithm 1) and

© 2016 The Author(s)
Eurographics Proceedings © 2016 The Eurographics Association.

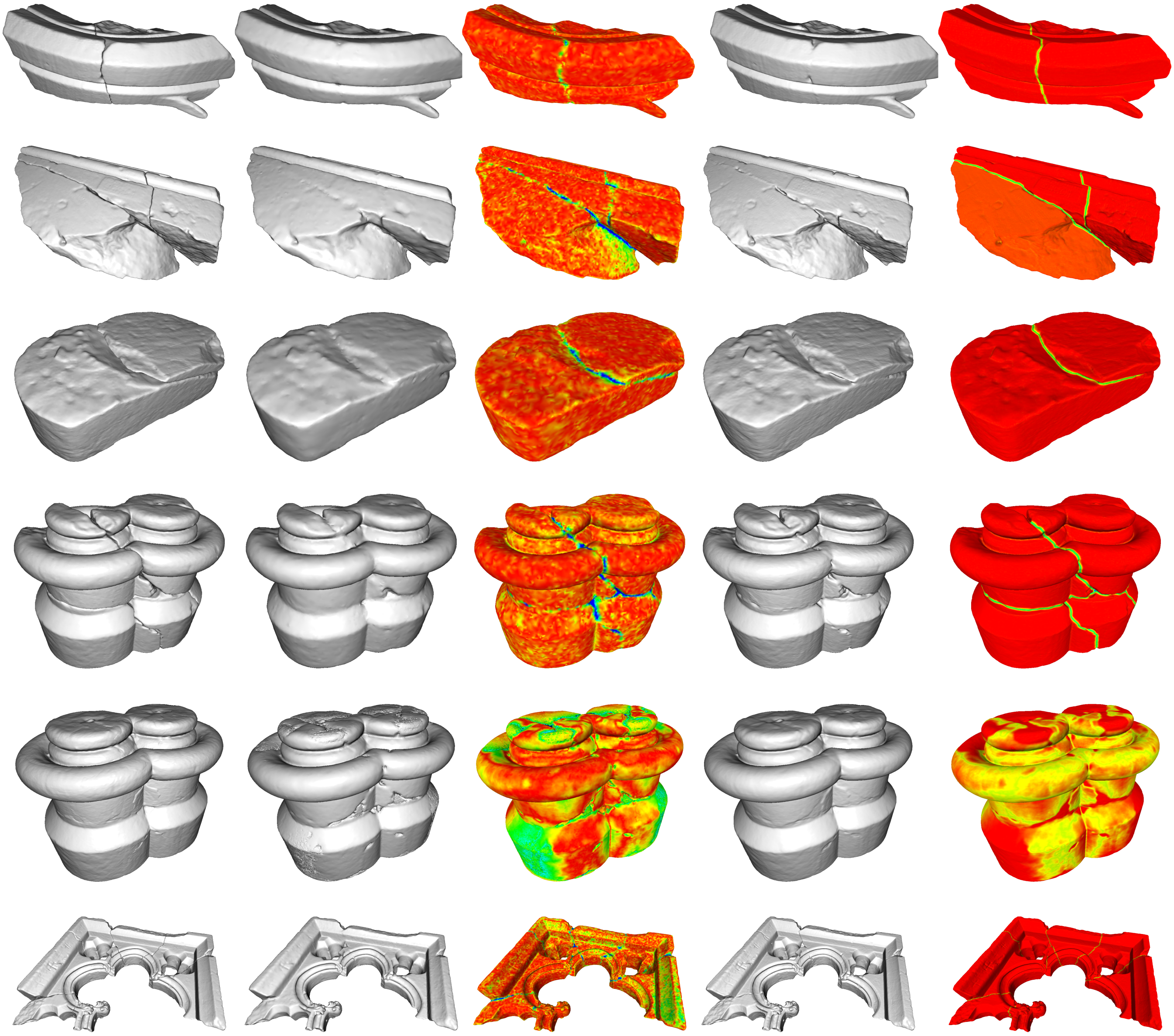

Figure 4: *Some of our results: input (left column), Filter/Poisson results (column 2 and 3) and our proposed method (column 4 and 5). For columns 3 and 5, increasing deviations of a result's surface to the input is represented by a color transition from red (near 0% deviation with respect to bounding box diagonal) over yellow (∼ 0.5%), green (∼ 1%) to blue (∼ 2% or more). Note that row 4 and 5 stem from the same object. However, row 4 is the reassembly of the entire set of digitized fragments whereas row 5 is the result of a reassembly of only a subset of fragments and a subsequent completion based on an approximation of planar symmetry. Here, the large self-intersections are present in the input data used for testing the approaches.*

the time of the Poisson Reconstruction, which includes the time for the 3D Delaunay triangulation, its refinement, solving the Poisson equation, generating the implicit function and extracting an isosurface. Timing measurement for our Soft Union Based Method is split into time for converting all the input meshes into Signed Distance Fields, the computation of the actual Soft Union operation and the time for the Polygonization of its result.

When examining our measurements in Table 1 and Figure 6, it is evident, that our Soft Union based method is several times faster for all sets. We aborted the Filtering/Poisson reconstruction of the EmbrasureFull model after more than 8 hours runtime (i.e. > 30000s). At the time, the Poisson Equation had not been solved. Meanwhile memory consumption (i.e. heap size) stagnated near 9 GB after a few minutes. In comparison, our Soft Union Method finished after less than 11 minutes and never exceeded 2 GB. Surprisingly, our proposed method is faster for the ColumnBaseFull set than for the

© 2016 The Author(s)
Eurographics Proceedings © 2016 The Eurographics Association.

Input Set	#Vertices	#Parts	Filter	Recon.	∑ Poisson	#Vertices	SDF	S.-Union	Polygon.	∑ Soft Union	#Vertices
Foundation	120k	2	0.25	15.53	**15.77**	11k	2.58	0.10	0.16	**2.84**	497k
Arch	130k	2	0.22	20.74	**20.96**	18k	0.88	0.04	0.05	**0.97**	344k
ColumnReassembled	255k	6	0.84	64.72	**65.56**	74k	8.2	1.12	0.42	**9.74**	1,877k
ColumnBase	260k	5	0.86	29.47	**30.33**	29k	6.07	0.59	0.27	**6.93**	924k
Block	280k	3	0.65	43.09	**43.74**	36k	6.00	0.38	0.32	**6.70**	1,229k
Embrasure	496k	8	1.21	101.26	**102.47**	89k	13.07	2.41	0.69	**16.17**	3,138k
ColumnmBaseFull	1,849k	2	7.93	496.55	**504.48**	702k	11.45	0.25	0.29	**11.99**	919k
EmbrasureFull	12,533k	4	60.62	n/a	**n/a**	n/a	70.18	3.66	1.23	**75.07**	5,385k

Table 1: *Runtimes of the Filtering/Poisson-combination (brown) and our proposed Soft-Union (green). Notice that the overhead of the Filtering step is negligible when compared to the time of the Poisson Reconstruction itself. The runtime of our Soft-Union approach is governed by the SDF creation time. Once a SDF representation of an input set has been created, multiple variations with varying radius r can be computed faster by roughly an order of magnitude. Changing the Filter radius of the Filter/Poisson approach requires a complete re-computation.*

Embrasure set, even though the former contains roughly 4 times more vertices.

4.3. Results of Experimental 3D Printing

For 3D printing the objects shown in Figures 9 and **??** we used a Dimension Elite printer by Stratasys. The Dimension Elite is a fused deposition modeling (FDM) printer with a workspace of 200mm x 200mm x 300mm and a two nozzle printing head. FDM is an additive manufacturing technology widely used by 3D printers. During the printing process, horizontal layers with a thickness of 0.178mm are gradually added starting from the bottom to the top of the model.

Similar to CNC- and other CAD related fabrication machines, the printer software readily accepts models in STL format. Before printing a model, the printer software computes a series of intersections of the model and thin horizontal cuboids, representing the boundary volume for each layer to obtain slices of the model. From these slices, machine code for driving the printer is generated.

The printing head extrudes strings of plastic filament made of ABS, a thermoplastic polymer, for plotting each layer, starting from its outer boundaries and subsequently filling the inner. Where a new layer exceeds the extent of a previous layer, the printer software generates additional support structure beneath it, which is applied by the second nozzle of the printing head. Depending on the fragility of an objects structures, the printer software determines whether the inner of an objects must be filled. For our rather coherent and merged results, material could be saved by leaving cavities only supported by a thin grid, as shown in Figure 7. After printing, the model is cleaned by washing out the support material.

With the meshes produced by our Soft Union approach, there were no additional steps needed to prepare them for printing. Computing the intersections, support structures and the generation of machine code took between 3 and 4 minutes. The printing of each of the models shown in Figures 9 and **??**, where the longest dimension is ranging from 12 to 14 centimeters, took between 16 to 28 hours and the subsequent cleaning process for removing the support structures about 4 to 5 hours.

4.4. Discussion

When considering the run-time measurements of our method, it is evidently well-suited to be used within an interactive system where the user can steer the merging radius r according to the task at hand and individual preferences. Such a practical adoption of our method within an interactive system, would best operate on a series of individual pairwise merging steps, where the user may adjust the soft union's radius r separately for each added fragment. Going further, it might also be used as a local operation where a user interactively positions a brush to merge gaps locally.

Computation time is governed by the creation of the individual SDFs, which only need to be computed once for testing with different parameter values. Also in terms of preservation of small-scale texture, our method shows very promising results. For all but the symmetry completed cases, the deviations introduced by our method on intact areas of an input set are beyond the resolution of the 3D printer. The preservation of textural patterns when using our method for 3D printing, was more limited by the printer model than our method. When taking into account our efficiency assessment and the further growth of resolution of digitization and digital fabrication hardware in the future, there is a strong indication that our method will be able to cope with the requirements.

When going into more detail, further research should be spend on improving the merging of symmetry completed models. In the worst case, our deviation measurements across intact patches of the input set correspond to a physical distance of approximately 2mm (corresponding to the thickness of roughly 10-12 printing layers), if such a model is printed with the maximum supported size for our printer. However when inspecting the 3D printed results of the symmetry-completed embrasure, we cannot observe any implausible textures and salient artifacts that are the result of our method. For an application of our method within an interactive system, the user could reduce the radius r when merging a pair of fragments that intersect to large extent and have near identical boundaries.

In all cases, our proposed method simultaneously outperforms the alternative Filter/Poisson variant and to our knowledge there are no other methods that address the problem with competitive plausi-

© 2016 The Author(s)
Eurographics Proceedings © 2016 The Eurographics Association.

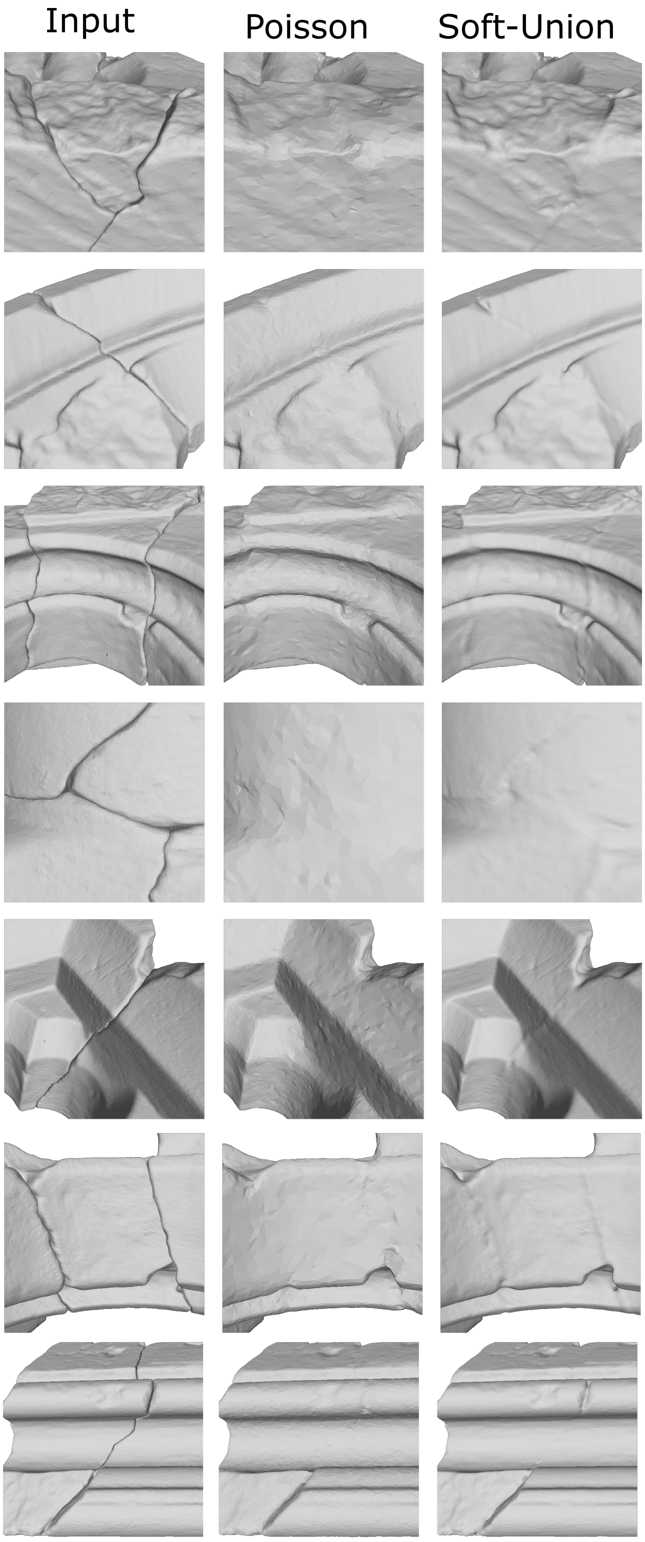

Figure 5: *Closeup comparison of input (left) and restioration results using Filter/Poisson (middle) and our Soft Union (right). Both methods are able to produce plausible geometry near merged fracture lines. However, notice the better preservation of texture such as small scratches and dents for the Soft Union method.*

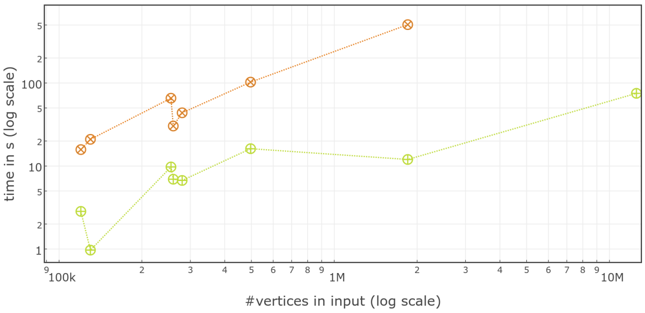

Figure 6: *Runtime of the Soft-Union (green) and Filter/Poisson approach (brown) by number of vertices in the input set. Our Soft Union evidently shows much better efficiency.*

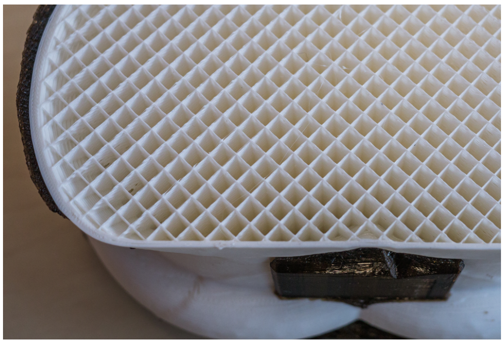

Figure 7: *Thin grid supporting the inside of a partially printed column base object. On the outside, part of the external support structure is also visible.*

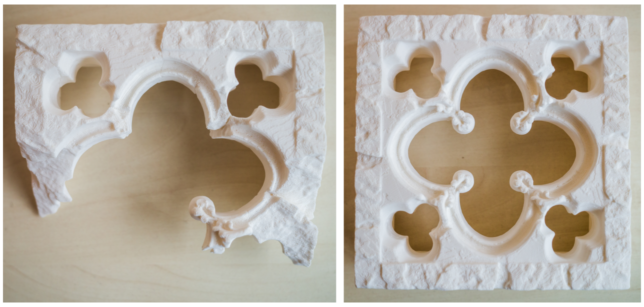

Figure 8: *3D printing results of the embrasure data set. Left: All existing fragments were merged with unconcealed (partially visible) fracture lines. Right: Complete restoration, where the missing parts were computed based on symmetry and all fracture lines were concealed.*

bility, precision and efficiency while also providing guarantees on the manifoldness of results.

5. Conclusions

We have presented an efficient, robust and easy to use method that is especially well suited within the context of virtual restoration of 3D CH Objects. While the mathematical principle used in our method is novel in context of the mesh repair domain, our experiments show, that our method is able to produce watertight meshes from digitally reassembled or completed fragment sets that can be readily 3D printed. Simultaneously it preserves small detail much better than an alternate approach adapting the well-known Poisson Reconstruction. At the same time, our method requires much less resources in term of runtime and memory when very high resolution models are to be restored.

For our test data, the method can be steered by a single parameter in order to produce results that feature plausible geometry at the former breaking lines, thus effectively repairing the defects at the area. Alternatively, the parameter can be adjusted to produce smooth transitions between or merge the fragments of a virtual reassembly to a single object for more efficient 3D printing without concealing the actually digitized information near the breaking lines, which also might be of interest in the CH domain. In both cases, the method speeds up the 3D printing and further automates the dissemination of virtual restorations of 3D CH objects.

Our efficiency measurements indicate, that within a system for interactive Virtual Restoration of 3D CH objects, a CH domain expert could use our method, while testing for different solutions of the merging process in an almost real-time manner.

Acknowledgements

Part of this work was funded by the European Commission in context of the FP7 STREP Project PRESIOUS, grant no. 600533 (http://www.presious.eu/)

References

[ACK13] Attene M., Campen M., Kobbelt L.: Polygon mesh repairing. *ACM Computing Surveys 45*, 2 (2013), 1–33. 2

[ASG16] Alliez P., Saboret L., Guennebaud G.: Poisson surface reconstruction. In *CGAL User and Reference Manual*, 4.8 ed. CGAL Editorial Board, 2016. URL: http://doc.cgal.org/4.8/Manual/packages.htmlPkgPoissonSurfaceReconstructionSummary. 3

[Bli82] Blinn J. F.: A generalization of algebraic surface drawing. *ACM Trans. Graph. 1*, 3 (July 1982), 235–256. 5

[BMR*99] Bernardini F., Mittleman J., Rushmeier H., Silva C., Taubin G.: The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics 5*, 4 (Oct. 1999), 349–359. 3

[BO05] Boissonnat J.-D., Oudot S.: Provably good sampling and meshing of surfaces. *Graph. Models 67*, 5 (Sept. 2005), 405–451. 3

[CCC*08] Cignoni P., Callieri M., Corsini M., Dellepiane M., Ganovelli F., Ranzuglia G.: Meshlab: an open-source mesh processing tool. *Eurographics Italian Chapter Conference 2008* (2008), 129–136. 3

[CGA] CGAL, computational geometry algorithms library. URL: http://cgal.org/. 3

[Eva15] Learning from failure: A survey of promising, unconventional and mostly abandoned renderers for dreams PS4, a geometrically dense, painterly ugc game. In *SIGGRAPH 2015 talk, Advances in Real-Time Rendering, Part II* (2015). 5

[GSP*14] Gregor R., Sipiran I., Papaioannou G., Schreck T., Andreadis A., Mavridis P.: Towards automated 3d reconstruction of defective cultural heritage objects. In *Proceedings of the Eurographics Workshop on Graphics and Cultural Heritage* (Aire-la-Ville, Switzerland, Switzerland, 2014), GCH '14, Eurographics Association, pp. 135–144. 1

[HFG*06] Huang Q.-X., Flöry S., Gelfand N., Hofer M., Pottmann H.: Reassembling fractured objects by geometric matching. In *ACM SIGGRAPH 2006 Papers* (New York, NY, USA, 2006), SIGGRAPH '06, ACM, pp. 569–578. 1

[Hop08] Hoppe H.: Poisson surface reconstruction and its applications. *Proceedings of the 2008 ACM symposium on Solid and physical modeling - SPM '08* (2008), 10. 3

[JLSW02] Ju T., Losasso F., Schaefer S., Warren J.: Dual contouring of hermite data. *ACM Trans. Graph. 21*, 3 (July 2002), 339–346. 5

[KBH06] Kazhdan M., Bolitho M., Hoppe H.: Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2006), SGP '06, Eurographics Association, pp. 61–70. 3

[LF08] Lee J., Funkhouser T.: Sketch-based search and composition of 3D models. In *EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling* (June 2008). 1

[MAP15] Mavridis P., Andreadis A., Papaioannou G.: Fractured Object Reassembly via Robust Surface Registration. In *EG 2015 - Short Papers* (2015), Bickel B., Ritschel T., (Eds.), The Eurographics Association. 1

[MSAP15] Mavridis P., Sipiran I., Andreadis A., Papaioannou G.: Object completion using k-sparse optimization. *Computer Graphics Forum 34*, 7 (2015), 13–21. 1

[Mus13] Museth K.: Vdb: High-resolution sparse volumes with dynamic topology. *ACM Trans. Graph. 32*, 3 (July 2013). 4

[PKT01] Papaioannou G., Karabassi E.-A., Theoharis T.: Virtual archaeologist: Assembling the past. *IEEE Computer Graphics and Applications 21* (2001), 53–59. 1

[PPCS13] Palmas G., Pietroni N., Cignoni P., Scopigno R.: A computer-assisted constraint-based system for assembling fragmented objects. In *Proc. of Digital Heritage 2013 International Congress* (2013), vol. 1, IEEE, pp. 529–536. 1

[PRE] PRESIOUS, 3d data sets. URL: http://presious.eu/resources/3d-data-sets. 5

[SGS14] Sipiran I., Gregor R., Schreck T.: Approximate symmetry detection in partial 3d meshes. *Computer Graphics Forum (proc. Pacific Graphics) 33* (2014), 131–140. 1

[SJT16] Sobiecki A., Jalba A. C., Telea A.: Robust Gap Removal from Binary Volumes. In *EG 2016 - Short Papers* (2016), The Eurographics Association. 2, 4

© 2016 The Author(s)
Eurographics Proceedings © 2016 The Eurographics Association.

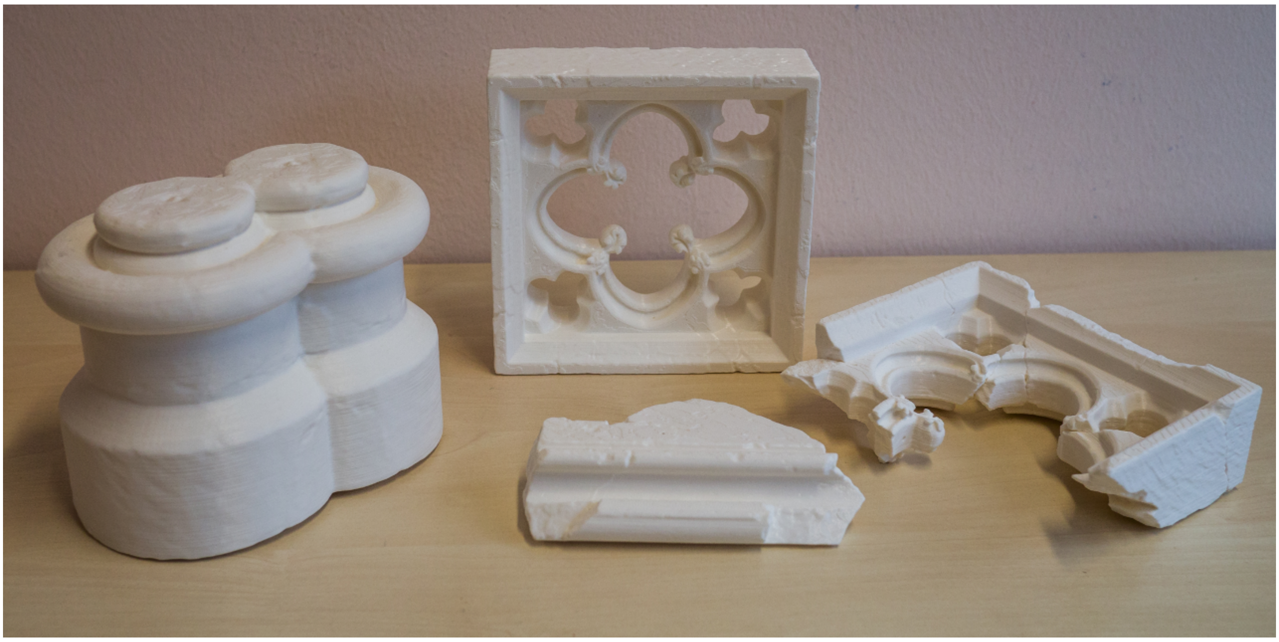

Figure 9: *A collection of 3D printed results produced by our method from left to right: reassembled column based with restorated fracture lines, reassembled and completed embrasure fragments with restorated fracture lines (top), reassembled arch with restorated fracture lines (bottom), and reassembled and merged embrasure fragments with unconcealed fracture lines.*

© 2016 The Author(s)
Eurographics Proceedings © 2016 The Eurographics Association.