

Shape Enhancement for Rapid Prototyping

P. Cignoni · E. Gobbetti · R. Pintus · R. Scopigno

Received: date / Accepted: date

Abstract Many applications, for instance in the reverse engineering and cultural heritage field, require to build a physical replica of 3D digital models. Recent 3D printers can easily perform this task in a relatively short time and using color to reproduce object textures. However, the finite resolution of printers and, most of all, some peculiar optical and physical properties of the used materials reduce their perceptual quality. The contribution of this paper is a shape enhancing technique, which allows users to increase readability of the tiniest details in physical replicas, without requiring manual post-reproduction interventions.

Keywords Shape Enhancement · Rapid Prototyping · Unsharp-masking

P. Cignoni
I.S.T.I. - C.N.R. - Via Moruzzi,1 - 56124 Pisa - ITALY
Tel.: +39-050-3152926
Fax: +39-050-3152604
E-mail: Paolo.Cignoni@isti.cnr.it

E. Gobbetti
CRS4 - ViC - Loc. Pixina Manna Edificio 1 - 09010 Pula (CA) - Italy
Tel.: +39-070-9250318
Fax: +39-070-9250216
E-mail: gobbetti@crs4.it

R. PINTUS
CRS4 - ViC - Loc. Pixina Manna Edificio 1 - 09010 Pula (CA) - Italy
Tel.: +39-070-9250347
Fax: +39-070-9250216
E-mail: ruggero@crs4.it

R. Scopigno
I.S.T.I. - C.N.R. - Via Moruzzi,1 - 56124 Pisa - ITALY
Tel.: +39-050-3152929
Fax: +39-050-3152604
E-mail: Roberto.Scopigno@isti.cnr.it

1 Introduction

Nowadays, 3D printing capabilities allow us to produce real world copies of digital 3D models. In the last years, these technologies have become significantly less expensive and improved in terms of printing quality. The number of possible applications of this technology ranges from reverse engineering and rapid prototyping to artistic and cultural heritage application. Not all technologies are equally valuable for all applications fields. In particular, 3D printing technologies exploit different materials with different properties. In the particular case of cultural heritage and artistic reproduction, our main field of application, the most common choice is to use powder-like materials, such as clay, which lead to objects with a diffuse opaque sandy texture grain, optionally colored in order to physically render user defined textures.

Although these sand-like materials are appropriate for the reproduction of many 3D models, their optical and physical properties make it difficult to reproduce fine details, even at scales when they are nominally visible according to the nominal printer resolution. It is therefore important to develop suitable color and shape pre-processing tools, which take into account the limitations of the physical printing process. For instance, a recent work [3] overcomes the perception problems due to an optical property (sub-surface scattering), by exploiting color reproduction capabilities of some 3D printers. However, in many cases, an approach based only on color enhancing is not sufficient. For instance, even though printers claim sub-millimeter resolution, the real, printed geometry is often affected by the very physical properties of these materials, that drastically worsen the surface resolution and decrease detail perception in real-world replicas.

The contribution of this paper is a geometry enhancement technique that counterbalances the effects due to the non-ideal behaviour of the materials used in the printing process, increasing physical replicas quality in terms of visual and tactile perception and detail preservation. The method is based on a volumetric representation of the geometry. The main idea of the approach is to simulate on this volume the physical behavior of the printer, to compare the result to the original geometry, and to modify the input data in order to reduce the difference between original model and printed one.

The main advantage of the proposed method is that much of the geometrical information ruined or completely lost during the printing pipeline is made again visible, improving the readability of the printed object. Further, since we modify the surface geometry, we can appreciate more details when touching the model. In this sense, our technique increases device printing capabilities and makes it possible to print high-quality models at a very small scale or, generally, with very fine details.

Although the proposed technique has some limitations that could occur in some particular pathological structures (e.g., a comb printed at very small scale, where the algorithm will enlarge at the same time all its teeth and the distance between them) the method has proven to work well on all kind of surfaces, with both low or high spatial frequencies.

The rest of the paper is structured as follows. In Sec 2, we review some related works on geometrical modification of 3D models for rapid prototyping. In Sec. 3, we describe in detail the physical printing process. Sec. 4 is devoted to the presentation of our enhancing algorithm. Finally, in section 5 we demonstrate the capabilities of our approach on a number of test cases.

2 Related work

The aim of our approach is to perform a geometric modification of a 3D surface for rapid prototyping purposes. The field of geometric enhancement is a well investigated one, and a lot of processing algorithms have been proposed.

General mesh processing techniques have been proposed to enhance in various controlled ways the overall shape of an object [6,11]. In particular, most approaches for geometric enhancement of meshes are based on the concepts of inverse smoothing and unsharp masking [9,13]. These methods first perform a smoothing operation (typically Laplacian smoothing), and then morph the mesh by increasing the difference between

the original and smoothed version of the surface. These approaches are intended to exaggerate some curves and features, and are well suited for rendering and visualization purposes. Unfortunately their direct application is not adequate for 3D printing for a number of reasons. First of all, these approaches are usually based on the enhancement of all the geometric features of the model, while in our case the set of features that are to be enhanced is strictly dependent on the resolution/properties of the 3D printing device in a very specific manner. Moreover, we have the strong requirement on the geometric correctness of the produced surface, while many of the proposed approaches can easily produce not printable parts or self-intersections that must be avoided when dealing with models for rapid prototyping.

The volumetric consistency issue and the fact that 3D printing devices usually rely on a volumetric discretization into voxels of the geometric shapes to be printed, lead to the choice of volume enhancement techniques. A number of authors have produced interesting results in this area. However, most of the techniques mainly deal with feature enhancement for visualization. Typically, they use a specific/particular volume rendering equation in order to improve the visibility of features [8,1,4]. Another approach [10] uses the same concept of unsharp masking applied to the voxel value instead of the color signal. They do not modify the volume data in terms of geometry, so, again, the method is not directly applicable.

Our work has also similarity to image enhancement techniques. Image processing for image enhancement is a very mature field, and a full review is beyond the scope of this paper. The techniques which are more closely related to our case are those that aim at preserving image topology after downsampling. For instance, Decenciere et al. [5] and Ma et al. [7] aim at preserving thin or small lines in downsampled images. The main application of these algorithms is the image visualization using mobile multimedia devices, which have small screens with small resolution capabilities. The result is achieved by performing an analysis of connected components of the neighborhood of each pixel and apply some enhancing filters before downsampling the image. A straightforward implementation of these 2D methods in the 3D domain is not possible. Moreover, 2D methods perform a uniform downsampling over the image, while we need to modify only particular critical parts of our volume in an adaptive way to take into account spatially varying detail erosion. The 2D kernels that are useful for the analysis of pixel neighborhood cannot be blindly applied to our case without taking into account the physical behaviour of the printing process.

For all these reasons, even if we can build on all of the insights of the previously cited methods, it is clear that a custom algorithm for improving printing of physical replicas is needed.

3 Physical Printing Process

There is a wide variety of 3D printing technologies, exploiting different kinds of materials and based on different technologies, such as deposition techniques, sintering strategies and other approaches. In this paper, we focus on the extremely popular glue and powder approach used by the ZCORP printing machines. Part of the described process (the discretization/slicing part) is common to the vast majority of technologies, while the modeling of the effect of diffusion process is very peculiar of this kind of machines. On the other hand, the overall approach (simulate and counterbalance) can be applied to different rapid prototyping technologies.

The printing process, depicted in Fig. 1, consists in three main steps, which respectively involve software, hardware (the printer) and operator. First, the software takes as input the 3D mesh of the printed surface and produces a discrete volumetric slice-by-slice representation of the geometry. Each single slice is a two-dimensional, 1bit mask, where the values zero or one mean that the voxel is respectively outside or inside the volume. The printer deposits a thin powder layer, reads the slice mask and, for each discrete voxel, if the value is one, deposits a glue drop on the powder. During this task, it acts as a normal sheet printer, although it uses glue instead of ink and powder in place of paper. It repeats this procedure for each slice, building the entire physical reproduction. At the end of this step, the model is buried in the powder and it is necessary to clean-out the printed object by removing extra powder around it. In order to make the model more robust, it is usual to coat its external surface with another special glue.

At the end of the process, we expect to have a faithful physical reproduction of the digital model. Unfortunately, each step of the printing pipeline introduces modifications with respect to the original surface. The more obvious problem is the slicing operator, which reduces the original model resolution to the finite sampling grid of the printer along x,y and z . Besides this, another problem arises during the glue deposition. In this step there, glue diffuses within the powder, resulting in two main side-effects, hole filling and detail erosion. Figure 2(a) schematically shows these effects in two extreme cases. If we have a very narrow hole, the glue deposited on the boundary diffuses into adjacent voxels, completely covering the hole. Thus, in the

printed model this hole will disappear. The opposite behaviour happens with tiny parts. Glue diffusion makes these parts less robust since it decreases the glue density around them and they will likely be broken during the cleaning step, which thus tends to erode tiny details.

4 Volume Enhancement

Our goal is to print models which are as similar as possible in shape to their digital representation. The main idea behind our approach is to simulate the physically behavior of the printer. Then, by looking at the results of the printing simulation, we try to counterbalance the errors introduced by the process by modifying the input geometry. An overview of the proposed algorithm is depicted in Fig. 2(b). Its main steps are the following:

- we start from the original shape and, by simulating the printing process, we produce another shape, which we expect to be as close as possible to the real printed one (e.g., it includes the unwanted geometric biases due to glue diffusion, with hole filling and erosion);
- we compute the difference between these simulated printed model and the original one, and we adaptively grow or shrink the original model in order to reduce this difference; after this step, if we re-apply the printing-simulation step, the resulting difference is expected to decrease;

The above process may possibly be iterated a few times, in order to further increase the similarity between the printed model and the original one. Even though, because of the heuristic shrinking and growing step, the method is not mathematically guaranteed to converge to a shape with the absolute minimal difference with respect to the original one, we have found that this simple and practical approach works very well in practice.

4.1 Printing Simulation

Before starting with printing simulation, we must transform the input mesh into a volume. We apply the voxelization method based on the work of Westermann and Ertl [12]. For each sampling direction (x,y and z) we choose the voxelization grid size as the minimum between the 3D printer sample spacing in that direction and the size of the smallest detail we want to preserve. Obviously we lose all details smaller than the volume grid resolution. The obtained data is an array of voxels with a state value equal to 1 or 0, which indicates whether a voxel lies inside or outside the object volume.

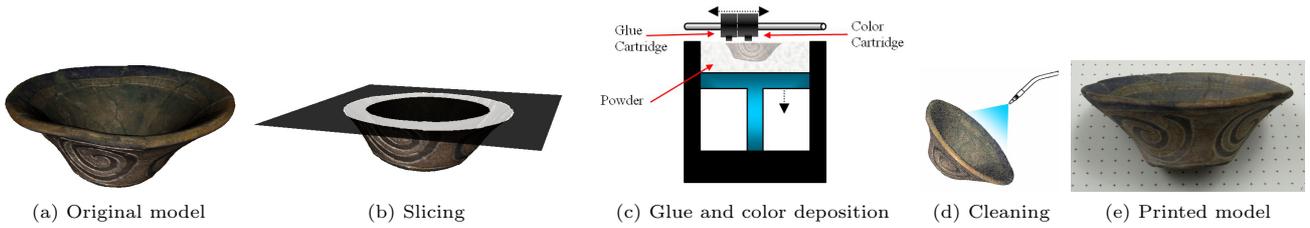


Fig. 1 Physical printing process steps: (a) The original 3D digital model. (b) Slice representation of the model. (c) Layer by layer deposition of powder, glue and color. (d) Extra powder removal using compressed air. (e) Final real-world replica.

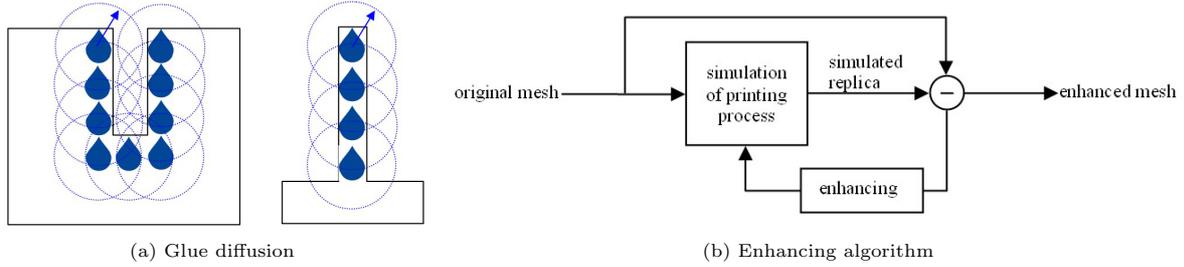


Fig. 2 (a) Because of glue diffusion, narrow holes disappear and tiny parts are easily eroded during the cleaning process. (b) The flowchart of the proposed algorithm, which takes as input the original model, produces another model by simulating the physical printing process and, by comparing these two models, it produces an enhanced model to be printed.

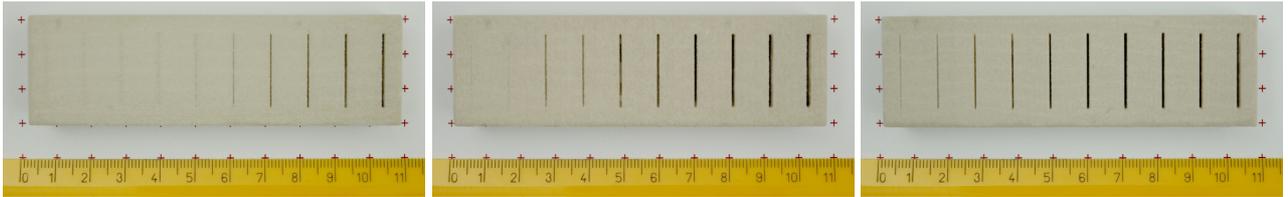


Fig. 3 The same 3D digital model (representing a parallelepiped with holes of thickness ranging from 0.1mm to 1mm) printed three times with the normals of the holes collinear with x,y and z axis directions (from left to right).

This is the representation of the input model that acts as input to the physical printing process simulation.

As stated above, the printer software first voxelizes the mesh according to the printer sampling grid by applying a slicing operator. Note that our original model is already voxelized, but with a sampling grid that is typically smaller than the device’s grid. Thus, we simulate the 3D printer’s lower resolution by downsampling the original volume using a morphological operator with a kernel size equal to the printing grid. If at least half a kernel contains voxels with value one, we set all kernel voxels equal to one. Otherwise, we set all these voxels as zero. We have experimentally tested that this approach is consistent with the typical printer software behavior. With an ideal printing process, the model obtained at the end of this task could become the simulated printed model and the enhancing algorithm would be only a resolution issue.

However, a glue diffusion effect (as depicted in fig.2(a)) acts as a spatial low-pass filter on the real replica, modifying the expected printed geometry because of hole filling and detail erosion. In order to simulate this be-

haviour, we start by applying a special smoothing operator to our volume. The main issue here is the choice of the proper smoothing kernel, which does not depend on the 3D printer resolution but, rather, on the printing process itself and on the physical behaviour of materials used for printing.

Unfortunately, there is no cue on the printer datasheet that could be useful to derive these numbers, so we must obtain these values experimentally. For this reason, we printed a 3D model that contains different size holes, ranging from 0.1 mm to 1mm thickness. Since the printer has three different resolutions in x,y and z, it will print a fine detail in a different way depending on its orientation. Thus, each test model is printed three times, with the normals of the holes collinear with the three axes directions. Analyzing Fig. 3 we can note that, even though the printer resolution in the x direction is about 0.08mm, in the physical replica we can only appreciate holes of thickness equal or bigger than 0.6-0.7mm. We find a similar behaviour in y direction, where we are able to see holes equal or bigger than 0.3mm, even though the nominal resolution is about 0.05mm. Sur-

prisingly, even if the resolution in z is 0.1mm, we can appreciate all the holes. The reason is that after depositing the glue for one layer, the time spent to deposit the powder is enough to make the glue become solid, so the new glue drops cannot widely spread over past layers (along z direction). This suggests us to consider just a two-dimensional smoothing kernel for simulating the diffusion process. This experiment suggests also to set the kernel size in x and y equal to the thickness of the first visible hole. In our case (fig. 3) the first visible holes in x and y are respectively 0.7 mm and 0.3 mm thick, so, considering that the printer's resolution in xy plane is 300x450 dpi, the applied 2D kernel will have a size of 16x10 voxels (remember that the voxelization is made with half the printer grid spacing). After applying this smoothing kernel we have a volume with voxel values in the range $[0:1]$. Finally, by thresholding these voxel values, we produce a volume with in or out voxels that simulates both the hole filling effect and the thinning/disappearing of tiny parts.

4.2 Enhancing

Before applying the enhancing algorithm, we compute the difference between the simulated printed model and the original one. The result is a volume with voxel values equal to -1, 0 or 1. In order to enhance the model, we work only on the voxels inside or outside the volume adjacent to model boundary in order to modify the volume geometry with respect to the difference signal. The idea is to look at the neighborhood of each voxel to decide whether its value should be changed or not. We apply the following equation:

$$\frac{1}{27} \sum_{\Omega_{ijk}} V^S - V^O = \begin{cases} 0 & V_{ijk}^{enh} = V_{ijk} \\ > 0 & V_{ijk}^{enh} = 0 \\ < 0 & V_{ijk}^{enh} = 1 \end{cases} \quad (1)$$

where V_{ijk} is the in or out original voxel adjacent to model boundary, Ω_{ijk} is the 3x3x3 kernel centered at $[i, j, k]$, V^S and V^O are respectively the voxels of the simulated and original voxelization models and V^{enh} is the enhanced voxel. If the average of the difference values in the neighborhood is zero, it means that the voxel remains unchanged from the high-resolution original model and the printed one (simulated), so we can keep the voxel value unchanged. If the average is bigger than zero (for instance inside a hole) and if the original voxel value is one, it means that its adjacent voxels receive too much glue; so, we change it as zero to decrease the local amount of diffused glue. Similarly, if the average is less than zero (tiny part), we set new voxel value equal to one. In such a way we dilate or erode the

boundary by one voxel at time. In order to have a nice improvement, we can iterate this procedure N times, where N corresponds to the ratio between the volume and the printer grid spacing.

In order to better understand the enhancing process we present an example applied to a 2D domain. In Fig. 4(a), we show a 2D in/out model (here black/zero means in and white/one means out). We present the steps of the algorithm during the first iteration: Fig. 4(d) shows the model after the voxelization, while Fig. 4(e) is the result of the slicing operator. As you can see, if there is no glue diffusion, the ideal print is very close to the original model. On the other hand, Fig. 4(f) shows the effects of glue diffusion on the real (simulated) printed model; the clover stem completely disappears, and there is a bit of hole filling effect in the upper part of the model. To perform the enhancement, we first compute the difference (Fig. 4(g)) between the voxelized and the real printed model, and, then, we perform the enhancing step (Fig. 4(h)). If this is the last iteration, Fig. 4(h) is the input to the printer, otherwise, we proceed with the next iteration. After the proper number of iterations we obtain the final enhanced model (Fig. 4(b)). If we re-apply the simulation of the real printing process to have an idea of what the real printer should produce (Fig. 4(c)), we can note that we preserve in the result both the holes (in the upper part) and the clover stem. Thus, the printed models is closer to the original one.

4.3 Printing the enhanced model

After we have reached the last iteration and computed the enhanced volume, we could give this model directly to the printer, since it is straightforward to compute slices from the volume data. Unfortunately, the printer's software accepts as input only triangulated meshes, and, thus, we need to convert our model to a triangle mesh. The printer software will then convert (again) the mesh into a set of slices. In order to avoid any further modification of the geometry during these conversions, we use the simple cuberille representation [2]. A clever and more complex isosurface polygonization (e.g., marching cubes or dual contouring) is useless and could be potentially dangerous for our purposes.

4.4 Handling color

If we need to apply the proposed algorithm to a mesh with a vertex color component, it is important to manage this color during the process. The resulting mesh

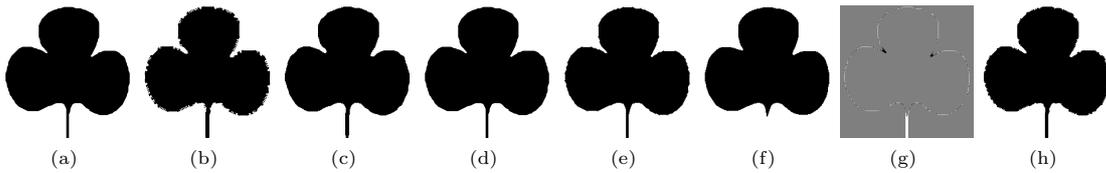


Fig. 4 The enhancement algorithm applied to a 2D domain: (a) The original 2D model; (b) The enhanced model (i.e., the input of the printer); (c) the simulated print of the enhanced model. Main steps of a single enhancing iteration: (d) voxelization of the original model; (e) simulation of the model printed with an ideal printing process; (f) simulation of the real printed object; (g) difference between the models (d) and (f); (h) the enhanced model at the end of one iteration.

will have different number of vertices positioned in different x, y and z coordinates, and thus a straightforward assignment of color per vertex is not possible. Actually, we need to be aware of the color only in two steps of our algorithm: the voxelization and the enhancing step.

In the voxelization step we build a volume that has a color value for each voxel in its boundary; this color must be consistent with the triangle mesh color. Thus, after computing the volume, we load a kd-tree using the vertices (with the associated color) of the original mesh and, for each boundary voxel in the volume, we find the nearest N vertices in the kd-tree. We use their color to compute voxel color by weighting them proportionally to the inverse of the distance from the voxel.

During the enhancement step we dilate or erode volume boundary. In this procedure, we will have some in (or out) voxels that become boundary voxels. But the in (or out) voxels don't originally have any color information. For this reason we compute the color based on the color of their adjacent voxels. Since we dilate (or erode) the boundary one voxel at a time, we are guaranteed to always find at least one neighbor with associated color information.

4.5 Streaming implementation

Since in our enhancing approach we are dealing with a high resolution printer (the nominal grid spacing is about 0.1mm) and with a volume representation, the amount of data involved in the process easily becomes large even if the original model is relatively small. However, each task performed in the enhancement algorithm works locally on a single voxel or on a small set of neighbor voxels and it operates independently of each other.

For these reasons, an out-of-core streaming implementation of the proposed algorithm proves to be a good choice in terms of efficiency. In such a way, we are able to reduce the in-core memory storage and to speed up the computational time by exploiting coherent access patterns. The idea is to choose a streaming direction (e.g., the z axis), and to store in memory only

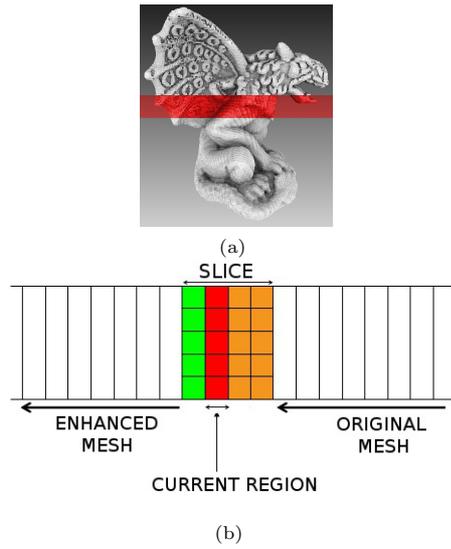


Fig. 5 Streaming regions (a): the current slice (red), the enhanced output (up) and the voxelized input mesh (down); (b) the structure and the dimension of the slice if the voxelization is performed with half the printer's grid spacing.

a small slice at a time. We span with this slice all the volume along this direction. At any given moment, all the data in the volume before the current z position has already been enhanced. In Fig. 5(a), we depict in red the current slice (spanning direction is top-down); the output (up) is the enhanced model, while the input, not processed yet, (down) is the voxelized volume.

Since in our case z is the streaming direction, it is easy to demonstrate that the minimum dimension of the slice (along z) is two voxel layers plus a number of voxel layers equal to the ratio between the z sample spacing in printer's grid and in our voxelization. This is mainly due to the slicing simulation step. In Fig. 5(b), the slice dimension is four because the voxelization grid is half the printer's grid spacing. The slice is structured as follows: the first voxel layer is already enhanced (green); the second voxel layer is the current layer (red), in which we compute the output; the remaining voxel layers (brown) are not enhanced yet but they contain neighboring information necessary to enhance the current one.

5 Results and Discussion.

We tested our method on a variety of models printed at various scales. In this paper, we discuss the results obtained with a set of them. The running times for generating all the shown models from the original meshes were in the order of a few minutes, thus always a tiny percentage of the whole printing time (which takes minutes to hours). Moreover, these preprocessing times could be significantly reduced if we could directly feed the printer with the computed slices instead of recreating a new complete 3D mesh representation.

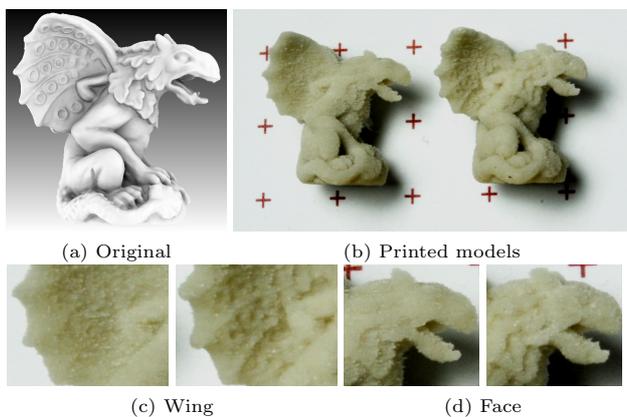


Fig. 9 The Gargoyle model printed on a very small scale: (a) the original model lit with ambient occlusion; (b) the printed replicas of the original and enhanced models; (c) and (d) are the comparisons between small parts of the original and enhanced models. In the latter more details are better perceivable and preserved.

For our experiments, we use a ZCORP-Z450 3D printer [14], that uses powder, binder and color to produce real-world replicas of 3D digital models. It builds the replica (max. 203x254x204 mm) by the deposition of the powder and the colored binder one layer at a time. Its resolution along the x and y axes is 300 x 450 dpi, while along the Z axis it may be set to 0.102 or 0.089 mm. The time required to print a model depends on its height along the z axis and on the selected layer thickness. Moreover, printing time slightly varies with size variations along x and y. It is straightforward that the thinner the layer, the longer it takes to print it. For example, given that the printing speed is about 2-4 layers per minute, the time required to print a model 10 mm in height with a layer thickness of 0.102 mm is about 30 minutes. Once the printing process is completed, it is necessary to wait about an hour and a half before extracting the object from the chamber in order to let the binder dry off perfectly. The next step is to remove all the extra powder from the object surface with compressed air. Due to the physical properties of the

powder, the resulting surface is very friable; therefore, the model needs to be strengthened by covering it in a special glue provided by the manufacturer.

Figure 6 shows each step of the first algorithm iteration applied to a real sample. After we voxelize the original mesh, we simulate the ideal print and the physical print. As you can see in the details in Fig. 7 and 8, the ideal print reduces details only in terms of resolution and, in these cases, the structures remain visible (e.g. the circles in the wing, the ear and the details in the mouth). On the other hand, the glue diffusion and erosion simulation produces hole filling effects (very clear in Fig.7(c)) and the disappearance of some tiny parts, such as the edge of the gargoyle’s ear or the gargoyle’s tooth (Fig. 8(c)). The last two figures (Fig. 6(e) and 6(f)) show the enhanced mesh at the end of the first and the second step. Figures 6(g), 7(d) and 8(d) are the simulation of real print applied on the enhanced models, in order to show how the algorithm preserves such details.

Figures 9, 10 and 11 show some examples of the proposed technique in action. For each model, we present the original 3D digital mesh (lit with a global ambient occlusion lighting model), a photo of the physical reproductions of the original and the enhanced model (the red cross grid is 1cm spacing, thus the scale is very small) and one or more images of particular parts of the 3D printed models. We can appreciate that the geometry of many tiny details is completely cancelled in the replica of the original model, such as very small circles in gargoyle wings and some details in its head, the geometric structure of the golfball surface, and the creases in the Buddha’s dress. All these fine details, printed on a very small scale, are now visible again in the printed replica of the enhanced model.

In addition to visually assessing the quality of replicas, we also made a touch test, which confirmed us that, after the enhancement, the tactile perception of the printed models considerably increases. Many details which are not perceivable in the original model prints, are sensed in the enhanced replicas.

Our method also allows to print colored models. The use of color [3] or geometrical enhancement depends on a number of elements. First of all, the use of color is necessary when the lost geometrical details cannot be adequately recovered due to resolution limits. As mentioned in Sec. 1, printing structures like an extremely fine comb cannot intrinsically be produce good geometries. In these cases, only color could increase the perception of details after the complete loss of tiny structures when printing on a very small scale. On the other hand, a geometrical enhancement is necessary when very low spatial frequency leads to a small variation

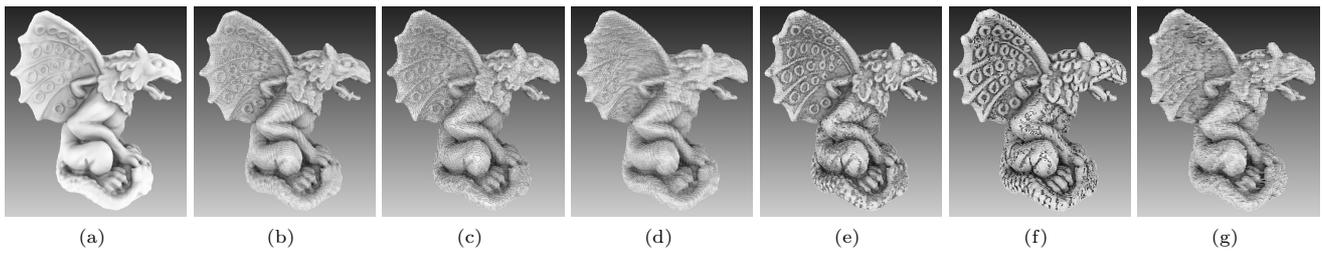


Fig. 6 The steps of a single algorithm iteration applied to gargoyle model: the original mesh (a); the voxelized model (b); simulated printed object in the ideal case (c); simulated printed object with glue diffusion (d); (e) and (f) are the enhanced mesh respectively after one and two iterations. (g) is the simulated real print of the enhanced model (f).

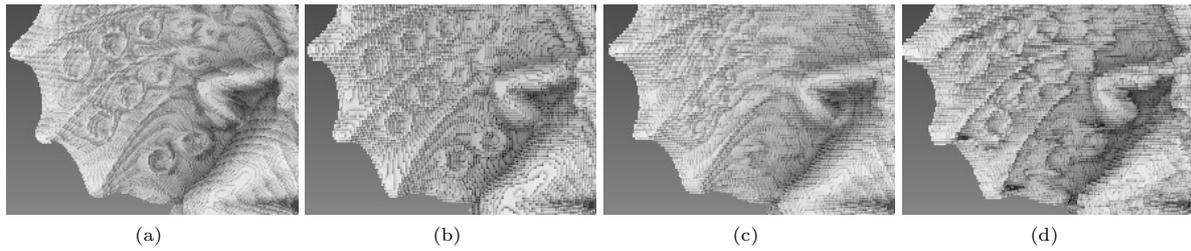


Fig. 7 A detail of the gargoyle model; the ideal print (b) differs from the voxelized model (a) only in terms of resolution, while in the simulated real print (c) a lot of details disappear due to glue diffusion and erosion. In this case the hole filling effect is evident in the circles on the gargoyle's wing. The readability of details disappeared in (c) is improved in the simulated real print of the enhanced model (d).

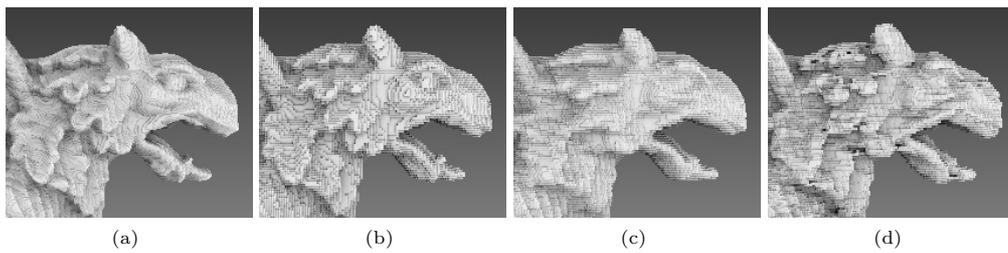


Fig. 8 A detail of the gargoyle model; the ideal print (b) differs from the voxelized model (a) only in terms of resolution, while in the simulated real print (c) a lot of details disappear due to glue diffusion and erosion. In this case the breaking of tiny parts is evident in the gargoyle's ear and in its tooth. The readability of details disappeared in (c) is improved in the simulated real print of the enhanced model (d).

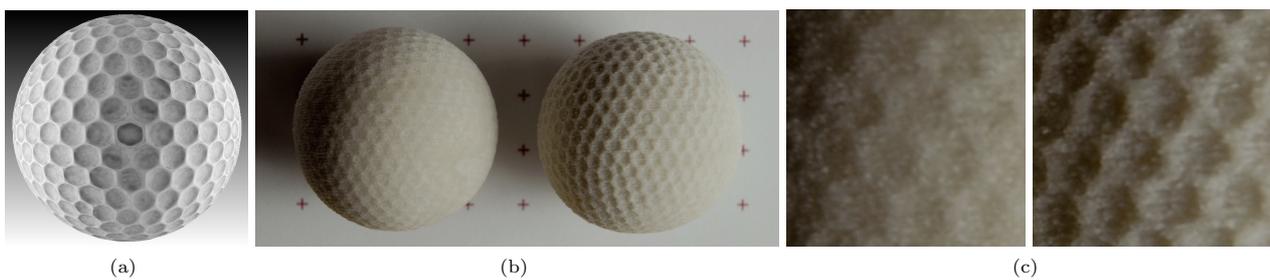


Fig. 10 The 3D model of a golfball: (a) the original model lit with ambient occlusion; (b) the printed replicas of the original and enhanced models; (c) is the comparison between a small part of the original and enhanced model. In the latter more details are better perceivable and preserved.

in color signal in the enhanced printed model. Further, since the tactile perception of an object surface is also of extreme importance, with volume enhancement we increase the model readability in terms of the sense of touch, too. Generally, the combination of both is advisable, in order to overcome the problems and to exploit capabilities of both these techniques.

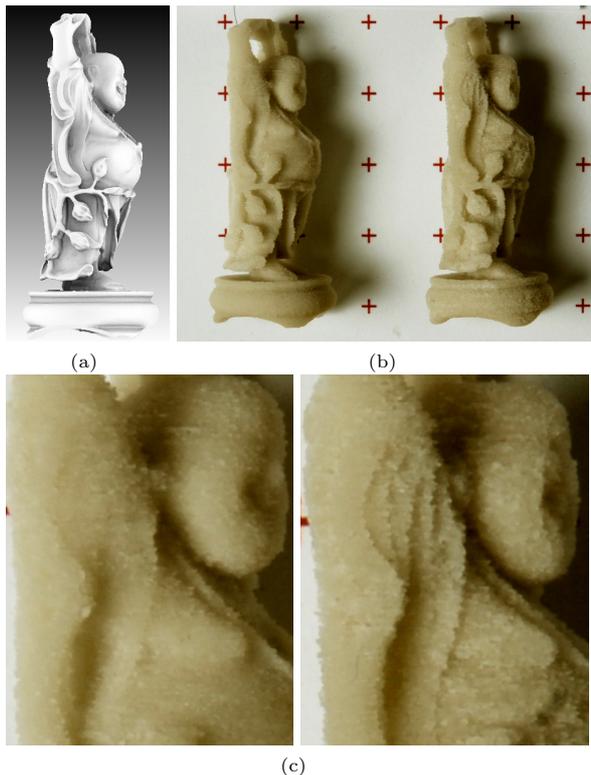


Fig. 11 A 3D model of buddha statuette: (a) the original model lit with ambient occlusion; (b) the printed replicas of the original and enhanced models; (d) is the comparison between a small part of the original and enhanced model. In the latter more details are better perceivable and preserved.

The results of using color and/or volume enhancement are depicted in Fig. 12. The models are lit with diffuse (Fig. 12(a)) and direct (Fig. 12(b)) lighting. With diffuse lighting, the perception is decreased a lot due to sub-surface scattering, so the models enhanced with color look better than the others. However, Fig. 12(b) shows that using only color enhancement is not enough. In this latter case the details of models enhanced with our geometrical approach are more perceivable. These two figures illustrate the importance of combining these two enhancing methods in order to increase object readability with a general variable lighting.

6 Conclusions

We have proposed a simple and effective technique to modify the geometry of a 3D model to preserve its finest details once it is printed using recent rapid prototyping techniques. The proposed method is based on a discrete volumetric representation of the input data, on a simulation of the shape resulting from the printing process and on a geometric shrinking and growing process that counter-balances the lost in details caused by the physical printing process. We have tested the method on a variety of models. Our tests show that the proposed method improves the appearance of the smallest details of a printed object. We foresee these kinds of methods, eventually combined with color enhancement, integrated within future printer drivers.

References

1. Cai, W., Chen, T., Shi, J.: Detail enhancement in volume rendering. In: J. Zhou (ed.) Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, vol. 2644, pp. 267–272 (1996)
2. Chen, L.S., Herman, G.T., Reynolds, R.A., Udupa, J.K.: Surface shading in the cuberille environment. *IEEE Computer Graphics & Applications* **5**(12), 33–43 (1985)
3. Cignoni, P., Gobbetti, E., Pintus, R., Scopigno, R.: Color enhancement techniques for rapid prototyping. In: The 9th International Symposium on Virtual Reality, Archaeology and Cultural Heritage (2008). URL <http://www.crs4.it/vic/cgi-bin/bib-page.cgi?id='Cignoni:2008:CER'>. To appear
4. Ebert, D., Rheingans, P.: Volume illustration: non-photorealistic rendering of volume models. In: VIS '00: Proceedings of the conference on Visualization '00, pp. 195–202. IEEE Computer Society Press, Los Alamitos, CA, USA (2000)
5. Etienne, D., Michel, B.: Adaptive crossing numbers and their application to binary downsampling. *Image Analysis and StereologyGraph*. **26**(2), 73–81 (2007)
6. Guskov, I., Sweldens, W., Schroder, P.: Multiresolution signal processing for meshes. In: Proceedings of SIGGRAPH, vol. 99, pp. 325–334. Los Angeles: ACM (1999)
7. Ma, R., Singh, G.: Large-scale infographic image downsizing. In: ICIP, pp. 1661–1664 (2004)
8. Marchesin, S., Dischler, J.M., Mongenet, C.: Feature enhancement using locally adaptive volume rendering. In: IEEE/EG International Symposium on Volume Graphics, vol. to appear. IEEE/EG (2007). URL <http://lsiit.ustrasbg.fr/Publications/2007/MDM07>
9. Steiner, A., Kimmel, R., Bruckstein, A.M.: Planar shape enhancement and exaggeration. *Graph. Models Image Process.* **60**(2), 112–124 (1998). DOI <http://dx.doi.org/10.1006/gmip.1998.0461>
10. Tao, Y., Lin, H., Bao, H., Dong, F., Clapworthy, G.: Feature enhancement by volumetric unsharp masking. *Vis. Comput.* **25**(5-7), 581–588 (2009). DOI <http://dx.doi.org/10.1007/s00371-009-0328-2>
11. Taubin, G.: Geometric signal processing on polygonal meshes. *Eurographics State of the Art Reports* (2000)

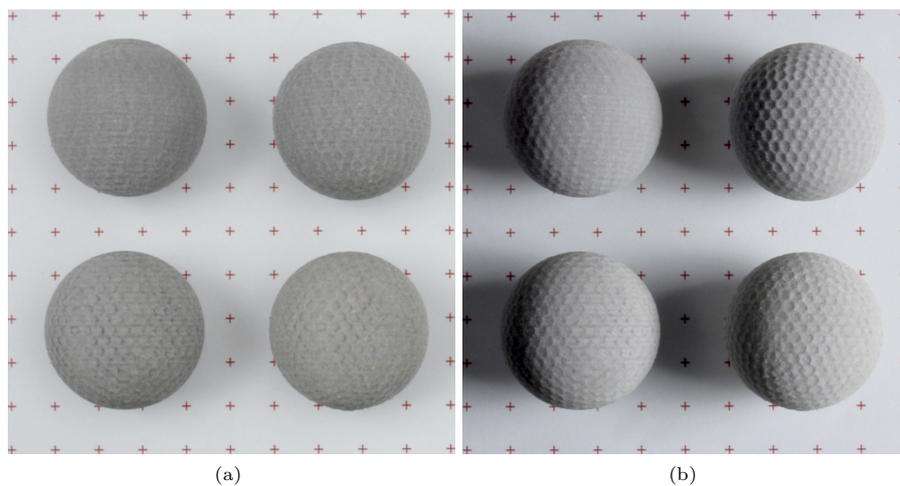


Fig. 12 Combination of color and volume enhancement lit with diffuse (a) and direct (b) lighting. UL: original model. LL: only color enhancement. UR: only volume enhancement. LR: color and volume enhancement.

12. Westermann, R., Ertl, T.: Efficiently using graphics hardware in volume rendering applications. In: SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques, pp. 169–177. ACM, New York, NY, USA (1998). DOI <http://doi.acm.org/10.1145/280814.280860>
13. Yagou, H., Wei, D.: High-boost mesh filtering for 3-d shape enhancement. *Journal of Three Dimensional Images* **17**(1), 170–175 (2003)
14. Z-Corporation: 3d printers. More info on: <http://www.zcorp.com/>