Voxel-Based Assessment of Printability of 3D Shapes

Alexandru Telea[1] and Andrei Jalba[2]

[1] Institute Johann Bernoulli,
University of Groningen, the Netherlands
a.c.telea@rug.nl
[2] Department of Mathematics and Computer Science,
Eindhoven University of Technology, the Netherlands
a.c.jalba@tue.nl

Abstract. Printability, the capability of a 3D printer to closely reproduce a 3D model, is a complex decision involving several geometrical attributes like local thickness, shape of the thin regions and their surroundings, and topology with respect to thin regions. We present a method for assessment of 3D shape printability which efficiently and effectively computes such attributes. Our method uses a simple and efficient voxel-based representation and associated computations. Using tools from multi-scale morphology and geodesic analysis, we propose several new metrics for various printability problems. We illustrate our method with results taken from a real-life application.

1 Introduction

Recent advances in 3D printing technology have made it possible to create faithful replicas of 3D models with increasingly higher resolutions, wider ranges of materials, higher printing speeds, and lower costs. High-quality, low-cost 3D printing is now available for personal users, *e.g.* through online services [12].

However, these advances have also created new challenges. Printer resolution limitations introduce several differences between input models and output objects, *e.g.*, in order of gravity: small layers of the surface being peeled, thin shape parts being fully removed, and shape break-up in several parts due to narrow connections. Currently, such defects are detected largely by manual inspection. This is not scalable for online printing services faced with thousands of models uploaded per day. What is needed is an automatic printability assessment method.

In this paper, we present a framework for 3D printability assessment. We analyze known defects which occur during printing and propose several metrics to detect and measure their criticality. We next compute these metrics fully automatically, but also allow users to tune parameters in order to select what is critical when printing a given model on a given printer. As 3D printing technology works in a raster fashion, we implement our proposed metrics using a voxel-based approach relying on multi-scale morphology and geodesic analysis. For maximal performance, we use a GPU implementation based on NVidia's CUDA.

The structure of this paper is as follows. In Section 2, we overview 3D printing basics and related work in shape thickness measurement. Section 3 presents the several stages of our printability assessment framework and related metrics. Section 4 presents results obtained on 3D models from a real-life 3D printing process. Section 5 discusses our method. Section 6 concludes the paper with future work directions.

2 Related work

3D printing technology constructs a model layer by layer, in a process similar to voxelization. The main critical element here is the shape's local *thickness* t: Shape details thinner than the so-called printer resolution τ result in empty voxels, which can lead to the problems named in Sec. 3. Hence, local thickness detection is a necessary (but as we shall see, not sufficient) part of printability assessment.

Several methods address thickness measurement. Hildebrand *et al.* define local thickness t at a point $\mathbf{p}$ in a 3D shape S, given by a set of voxels $\Omega \subseteq \mathbb{R}^3$, as the diameter of the largest inscribed ball B centered at $\mathbf{x}$ that contains $\mathbf{p}$ [3], *i.e.*,

$$t(p) = 2 \max(\{r \mid \mathbf{p} \in B(\mathbf{x}, r) \subset \Omega, \mathbf{x} \in \Omega\}), \tag{1}$$

with r the radius of ball B. Dougherty *et al.* implement the above by evaluating the distance transform $D(S)$ over the skeleton (or medial axis) of S and marking all points $p \in \Omega$ with the maximum of $D|_\Omega$ [3]. However, this method requires a robust and exact medial axis transform, which is prohibitively expensive for large voxel models.

Yezzi *et al.* compute thickness between surface pairs by minimal-length surface-to-surface paths using a PDE-based approach [14]. While this method is relatively fast (12 seconds for a 256^3 voxel dataset), its extension to higher-genus models is not evident. An overview of voxel-based thickness estimation in medical applications is given in [11].

Mesh-based methods for thickness computation have been proposed by Lambourne *et al.* [5]. However, they typically require a clean mesh with no self-intersections, do not offer a volumetric object representation (which is essential for subsequent printability assessments), are more complex to implement, and are only shown to work well on relatively simple shapes. Many engineering tools measure the so-called *wall thickness* [4]. However, all such tools we are aware of work only for models consisting of few relatively large, simple, surfaces separated by clear edges.

Considerable work exists in the area of topology-accurate 3D shape digitization, *e.g.* [13]. However, our problem is different: given a digitization process not under our control (the 3D printer), and a specific set of quality criteria (mainly thickness-related), we must assess whether a given polygonal model fullfils these criteria under the given digitization.

3 Printability Assessment

Printability assessment supplements thickness estimation with additional constraints. A 3D shape is printable when the removal of its thin regions (which do not print) do not create *critical* topological or geometrical changes, as follows (see also Fig. 1):

- disappearance of salient detail, or *spikes*, *e.g.* the mouse's tail or whiskers;
- disconnection of large shape fragments by removal of so-called *bridges*, *e.g.*, the mouse's thin arms;
- creation of large *holes*, also called tunnels, *e.g.*, center of sandal's sole.

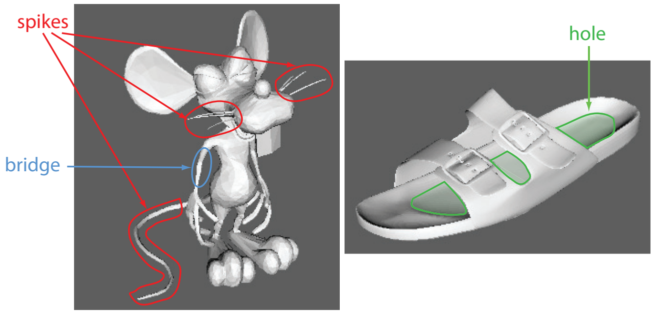


Fig. 1. Printability critical events: spikes, bridges, and holes (see also Fig. 5)

Filling of small object regions such as holes of interstices due to printer discretization is not seen as a problem in 3D printing practice, as such details are few in typical 3D shapes on the printing market. Also, filling does not cause parts of the object to break off or disappear, which is seen as the largest problem.

Following discussions with a market-leading 3D printing service provider [12], we learned that an effective printability assessment method should be able to detect and measure all above-mentioned events related to spikes, bridges, and holes for a shape of arbitrary topological and geometric complexity and given printer resolution. Secondly, the criticality of events should be quantifiable. Intuitively, this should match the visual salience of these events. Finally, the method should be automatic and efficient, *i.e.*, handle thousands of models a day on a PC computer. According to their experience, no such ready-to-use method exists.

We next present a method that efficiently and effectively implements the above requirements. Fig.2 shows our computational pipeline.

3.1 Voxelization

First, we transform the polygonal 3D input shape S into a binary voxel model. We use the method of Nooruddin and Turk [9], based on an optimized implementation of *binvox* [7], which delivers high performance and can handle any

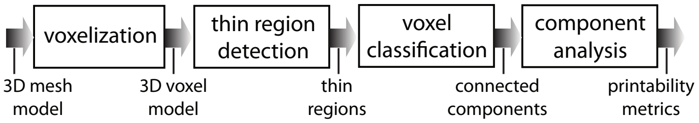


Fig. 2. Thickness assessment pipeline

closed mesh, including self-intersecting ones (see Sec. 4). The output is a binary volume consisting of the shape Ω (foreground) and its complement, the background, $\overline{\Omega} = \mathbb{R}^3 \setminus \Omega$. The voxelization resolution used in practice exceeds that of the 3D printer, so small potential errors caused by the voxelization methods should have no effect on our assessment.

3.2 Thin region detection

Given a user-defined thickness τ (equal to the printer resolution), we next detect the locally thin areas Θ of Ω. This can be formulated as a *top-hat* transform in the context of multi-scale morphology [1, 6], as follows.

Given the set $\Omega \subseteq \mathbb{R}^3$, its distance transform with respect to the $||\cdot||_2$ (Euclidean) norm is defined as

$$D(\Omega)(\mathbf{x}) \equiv \bigwedge_{\mathbf{y}\in\overline{\Omega}} ||\mathbf{x}-\mathbf{y}||_2\,. \qquad (2)$$

Note that the distance transform of a shape can be computed both inside, respectively outside, of a particular shape, by inverting the roles of Ω and $\overline{Omega}$ in Eq. 2, as inferred from the context of use.

The 3D unit ball B associated to the norm is

$$B \equiv \{\mathbf{x} \in \mathbb{R}^3 \mid ||\mathbf{x}||_2 \leq 1\}.$$

The *multi-scale set dilation* and *erosion* of Ω by B at scale s are given, respectively, by

$$\delta_B(\Omega, s) \equiv \Omega \oplus sB$$
$$\epsilon_B(\Omega, s) \equiv \Omega \ominus sB.$$

We detect locally thin areas Θ of the shape Ω using a multi-scale (set) top-hat transform at scale $s = \tau$, *i.e.*,

$$\Theta(\Omega, \tau) = \Omega \setminus \delta_B(\epsilon_B(\Omega, \tau), \tau). \qquad (3)$$

The erosion in Eqn. (3) removes border elements, narrow spikes and bridges connecting the main object structures. Dilating the resulting set reconstructs the borders of the object, but not the other removed structures. Finally, taking the set difference between Ω and the smoothed (*opened*) object, the thin structures are effectively extracted. By setting τ to the printer resolution, we thus detect the unprintable parts, *e.g.* the blue areas in Fig. 3.

Since thresholding the distance transform $D(\Omega)$ at levels $s > 0$ yields multi-scale dilations of Ω by balls sB of radius s, *i.e.*,

$$\delta_B(\Omega, s) = \{\mathbf{x} \in \mathbb{R}^3 \,|\, D(\Omega)(\mathbf{x}) \leq s\} \tag{4}$$

and similarly,

$$\epsilon_B(\Omega, s) = \{\mathbf{x} \in \mathbb{R}^3 \,|\, D(\overline{\Omega})(\mathbf{x}) \geq s\}, \tag{5}$$

one can express Θ from (3) using (4),(5) as

$$\Theta(\Omega, \tau) = \Omega \setminus T_l(D(T_h(D(\overline{\Omega}), \tau)), \tau) \tag{6}$$

where T_h and T_l are the sets obtained by thresholding at level τ its (function) argument and by keeping only values higher and lower than τ, respectively.

The dilation replaces corners and edges from the eroded shape with spherical caps, respectively edge-swept cylindrical surfaces of radius τ, a well-known result from mathematical morphology. Practical observation shows that this rounding closely follows the actual behavior of 3D printers due to small deviations of the printer head of the order of the machine resolution.

Unlike other approaches (see Sec. 2), our thin-area estimator(Eq. (3)) can be very efficiently computed due to the result in Eq. (6), using a fast and exact method for computing the Euclidean distance transform (see Sec. 5). As an example, Fig. 5 e shows the volume-rendered distance transform of the rabbit model - blue indicates small distances, while red indicates large distances from the model's surface.

3.3 Voxel classification

As already outlined, not all thin regions are equally critical for printability. Equation 3 does not distinguish between topological or geometric shape properties besides local thickness. For example, sharp edges would be classified as thin, while their removal amounts to limited rounding (Sec. 3.2), which is acceptable. Area or volume-based aggregated metrics are not sufficient for discrimination, *i.e.*, a shape with significant small-scale noise would yield thin regions with cumulative large area and/or volume, the removal of which would not significantly alter the shape. What is needed, is the detection and analysis of salient spikes, holes, and bridges.

To detect and analyze such events, we next classify the voxels into four categories (Fig. 3): thin (T), rump (R), interface (I) and boundary (B), as follows[3]:

$$\begin{aligned}
R &= \Omega \setminus \Theta \\
I &= \{\mathbf{x} \in \Theta \,|\, \exists \mathbf{y} \in n_6(\mathbf{x}), \mathbf{y} \in \Omega, \mathbf{y} \notin \Theta\} \\
B &= \{\mathbf{x} \in \Theta \,|\, \exists \mathbf{y} \in n_6(\mathbf{x}), \mathbf{y} \in \overline{\Omega}\} \\
T &= \Theta \setminus (B \cup I),
\end{aligned}$$

[3] We recommend viewing the figures in this paper in full color

where $n_6(x)$ denotes the 6-connected voxel neighbors of $\mathbf{x}$. In a post-processing step, we next mark all I voxels with no B or T neighbors as R. This removes one-voxel-thick 'interface shells', *i.e.* thin surface-like voxel sets, which are tangent to rump, but not to thin regions, and thus have no further use in the printability assessment. Finally, we compute the thin (T) 26-connected components. For each component C_i, we store its I, T, and B voxels, boundary area A_i^B, and volume V_i. Also, for each C_i, we compute and store the 26-connected interface components I_{ij} as well as their areas A_{ij}^I. For robust area estimation on voxel surfaces, we use the fast estimator proposed by Mullikin and Verbeek [8].

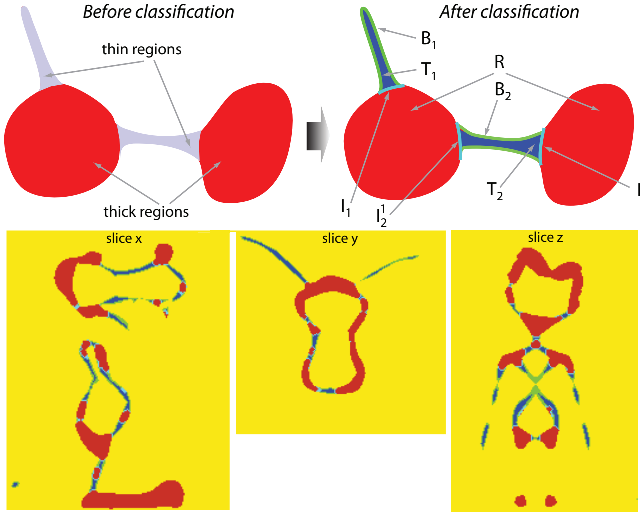


Fig. 3. Voxel classification into thin (T=blue), rump (R=red), interface (I=cyan), and boundary(B=green). Bottom: slices from actual voxel model in Fig. 1 left)

For example, the shape in Fig. 3 (top) has two thin regions, a spike and a bridge; the spike has one interface, while the bridge has two. Thin region classification is done by topological analysis: Bridges have several interfaces, $|\{I_{ij}\}_j| > 1$; spikes have an interface of genus zero; and holes have an interface of genus larger than zero. This information is important in assessing the printability metrics presented next.

3.4 Printability metrics

We now describe two metrics which are computed on the thin components resulting from the voxel classification (Sec. 3.3) to assess a shape's printability.

Area-based metric Spikes, bridges, and holes share one common property: they are thin, elongated, structures, which connect to the rump by small-size interfaces. The *visual salience* of such a structure C_i can be encoded as the ratio of their (total) boundary area $A_i^B = \sum_j A_{ij}^B$ and interface areas $A_i^I = \sum_j A_{ij}^I$, *i.e.*, $\mu_A(C_i) = A_i^B / A_i^I$. This metric has several desirable properties. First, it marks large removed structures with small interfaces (connections to rump), *e.g.* the ears of the mouse in Fig. 1, as more critical. This matches practical 3D printing insight: large structures are visually more salient, and the smaller their interfaces are, the higher the chance is that a 3D printer will not manage to render such connections, or that such connections will be mechanically fragile – an important property for the lifetime of printed shapes. Secondly, it treats spikes, holes, and bridges uniformly, without the need of a full-shape connectivity analysis. Computing μ_A from the classification results (Sec. 3.3) is straightforward.

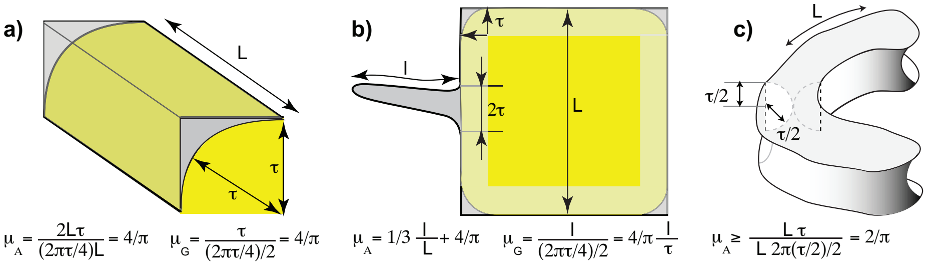

a)
L
τ
τ
b)
l
τ
2τ
L
c)
L
τ/2
τ/2
μA = 2Lτ / (2πτ/4)L = 4/π
μG = τ / (2πτ/4)/2 = 4/π
μA = 1/3 l/L + 4/π
μG = l / (2πτ/4)/2 = 4/π l/τ
μA ≥ L τ / L 2π(τ/2)/2 = 2/π

Fig. 4. Printability metrics: edge (a); 2D view of spike on edge (b); thin hole (c); Thin areas are gray and rump areas are yellow respectively.

The adimensional area-based metric μ_A is intuitive to interpret. For *spikes*, since interfaces are locally spherical (Sec. 3.2), μ_A is lower bounded by 1, when round shape parts are considered asymptotically thin for $\tau \to 0$. For right-angle edges, $\mu_A = 4/\pi$, *i.e.* the ratio of the area of two rectangles sharing the edge and having one side of length τ and the area of a quarter-cylinder (Fig. 4 a). Sharper spikes, which we want to detect, yield larger μ_A values. Near-constant cross-section spikes of length l, *e.g.*, the mouse's tail or whiskers in Fig. 1 have μ_A of the order of l/τ, since A^I is of the order of τ^2. This allows one to easily threshold μ_A to get all thin, elongated features longer than a given l. For *bridges*, consider a bridge component C_i having n branches $j = 1 \ldots n$ of boundary and interface areas A_{ij}^B and A_{ij}^I, respectively, so $\mu_A = \sum_j w_j (A_{ij}^B / A_{ij}^I)$, where $w_j = A_{ij}^I / A_i^I$. Hence, a bridge's metric is the sum of its branches' area metrics weighted by their interface sizes. For *holes*, μ_A is lower bounded by $2/\pi$, *i.e.* the ratio of a surface component of constant thickness 2τ and perimeter L and a cylinder surface of radius τ and length L (Fig. 4 c).

Geodesic length metric However effective, the area-based metric will fail detecting certain complex unprintable structures. Imagine a thin cylindrical filament of radius τ and length l touching the edge of a cube of size L (Fig. 4 b).

The thin region T (gray in Fig. 4 b) contains the filament *and* all edges of the cube, hence,

$$\mu_A = \frac{2\pi\tau l + 12(2\tau L)}{12(2\pi\tau/4)L} = \frac{1}{3}\frac{l}{L} + \frac{4}{\pi}. \tag{7}$$

If L is much larger than l, $\mu_A \to 4/\pi$, the right-angle edge metric, so we miss the unprintable filament. Note that this only occurs if the filament and cube edges form a *single* component, otherwise the filament retains its separate, large, μ_A value.

To solve this problem, we introduce a different metric, μ_G, based on geodesic lengths. For a given component,

$$\mu_G = \frac{\max_{\mathbf{x}\in B,\mathbf{y}\in I} ||(\mathbf{x} \rightsquigarrow \mathbf{y})||_2}{\max_{\mathbf{x}\in \partial I,\mathbf{y}\in I} ||(\mathbf{x} \rightsquigarrow \mathbf{y})||_2}, \tag{8}$$

where $(\mathbf{x} \rightsquigarrow \mathbf{y}) \subset \partial\Omega$ is a geodesic *path* on the shape surface $\partial\Omega = B \cup I$ between points $\mathbf{x}$ and $\mathbf{y}$.

The adimensional metric μ_G measures the eccentricity of a component. For right-angle shapes, $\mu_G = \mu_A = 4/\pi$. Long *spikes* have high μ_A values. For the filament-and-cube shape, $\mu_G = (4/\pi)(l/\tau)$, *i.e.*, the value of μ_A for a right-angle-shape times the filament's eccentricity l/τ. For *bridges*, μ_G reflects half of the length of the longest interface-to-interface geodesic path, which is useful, as such branches are the likeliest to break after printing. For *holes*, μ_G equals the hole's surface diameter divided by the hole's wall thickness, which intuitively marks large-area, thin, regions as most critical.

We efficiently compute μ_G using the distance transforms $D(\partial I)|_B$ and $D(\partial I)|_I$ restricted to the voxels of B and I, respectively. These are nothing that the application of distance transforms in 3D voxel space on the voxel sets of B and I from the sites-set ∂I respectively. Note that, implementation-wise, ∂I is also included in B.

The maximum values of $D|_B$ and $D|_I$ give the maximal geodesic lengths from ∂I to all points in B and I, respectively. Since these distance transforms are computed on 3D surfaces rather than volumes, we cannot use volumetric distance transforms (see Sec. 5). Instead, we use the exact Euclidean TFT method [10], as it is simple to implement and can handle curved voxel surfaces. Any other fast, exact distance transform on curved voxel surfaces can be equally used. The Euclidean TFT is $O(N \log N)$ for a surface of N voxels. For typical B and I surfaces in a 512^3 voxel volume, a CPU-based Euclidean TFT takes a few seconds. As an example, Fig. 5 c shows the geodesic distances for the thin fragments of the mouse model. The tail's tip (red) sticks out as being the furthest point from the body.

4 Results

Figure 5 shows several results computed on a MacBook Pro laptop with 4 GB RAM and a GeForce 8600M GT card with 512 MB RAM. The models range from a few thousand to over hundred thousand polygons, and clearly contain non-trivial geometry and topology. All models are voxelized at 256^3 resolution.

The classification (using the same colormap as in Fig. 3) and and metric values (using a blue-to-red colormap) are rendered composited with the original polygonal surface (in gray). We see that the classification accurately detects thin regions, even in complex cases. For example, the mouse's torso is largely green, since the model is here actually not full, but consists of a thin shell – see Fig. 3 bottom. The area metric marks the ears as the most critical thin components. Indeed, these have a very large area but very small interfaces. However, compared to the ears, the tail is much less important, as it has a relatively small area compared to its interface. The geodesic metric also finds the ears as critical, but also finds the tail, which is thin and very elongated. A similar phenomenon occurs for the 'bars' model (Fig. 5 f-h). Here, the long vertical spike (marked with a stippled line) is clearly detected by the geodesic metric. For the rotor model (Fig. 5 i-k), the area metric identifies fragments of the rotor's outer edge (red in Fig. 5 j) as most critical. Note that these are bridges, and their removal would change the rotor's blade topology, *i.e.* disconnect its blades. The geodesic metric finds the four inner attachment parts (red Fig. 5 k) as most critical, as they are the most elongated thin regions to get removed. For the 'logo' model (Fig. 5 l), the two metrics identify critical fragments similarly. Finally, the sandal model shows how genus 0 and genus 1 thin regions (the sandal tip and heel, respectively) are captured, and that both the area and geodesic metric have values which do not depend on the thin region's genus, which desirable for user setting of the critical metric value.

The area metric captures critical fragments of large surface *area*, while the geodesic metric captures fragments of large *elongation*. The two are different, but equally critical, printability problems. In practice, we found that marking those fragments as printing problems, which are found critical by either metric, to be a robust and good solution to our overall assessment goal.

We validated the relevance of these results, *i.e.* the fact that the regions detected as critically thin for 3D printing by our algorithm would indeed create problems in actual 3D printing production in practice. Although the validation only involved domain experts looking at our results and not physical printing, the outcome was positive: our method identifies regions which a domain specialist would also mark as critical.

5 Discussion

Printability assessment Our method is able to capture all top-level printability requirements (Sec. 3) by means of two simple, adimensional metrics. The metrics work uniformly for all types of events (spike and bridge removal and hole creation) on all geometric and topological configurations we could test on around 100 models used in 3D printing production. The entire method has only two parameters: the printer resolution τ and the desired, critical-event detection threshold for the used metrics. For now, the method can be used in a semiautomatic manner: the algorithm processes a set of shapes, identifies those with printability problems, and displays them (with highlighted problems as in Fig. 5) for further human inspection. Although not fully automatic, this is a major step forward as compared to the current procedure where users can only see the original 3D shapes.

Scalability The most expensive step of our method, the thin region detection (Sec. 3.2), is implemented by adapting the recent CUDA-based distance transform (DT) of Cao *et al.* [2]. On our platform (Sec. 4), this step takes under 2 seconds for 256^3 voxels. Since Eqn. 6 uses only distance transforms and thresholding operations, we can safely extrapolate the timings from [2] for a Tesla C1060 card to our case, *i.e.* 0.7 seconds (512^3 voxels) and 5.8 seconds (1024^3 voxels). Since we work in voxel space, polygon count for the input model does not affect these values. Apart from thin region detection, all other steps are of negligible costs (seconds) and thus implemented on the CPU in C++.

Our current CUDA implementation requires 8 bytes GPU memory per voxel, *i.e.* 128 MB for 256^3 voxels, 1 GB for 512^3 voxels, and 8 GB for 1024^3 voxels. Current 1 GB cards thus allow measuring printability of objects of 10 cm size at the resolution of $100/512 \simeq 0.2$ mm, which is in line with the practical requirements for 3D printability assessment. It is, however, possible to improve on this by using the slice-based DT computation refinement sketched in [2], which allows volumes up to 8192^2 voxels per slice to be handled with only 512 MB with a small performance loss.

Precision All distances we compute (CUDA DT for the thin regions and Euclidean TFT for the geodesic length metric) are exact. This is essential as small errors may assess a printable object as unprintable or conversely, which is unacceptable by the users.

Limitations Voxelization resolution is currently the main limitation. Critical events smaller than one voxel will be missed. For example, the mouse's whiskers are not captured by the 256^3 voxel grid used in Fig. 5. However, as outlined above, this problem is directly solved by using graphics cards which can hold larger voxel volumes.

6 Conclusions

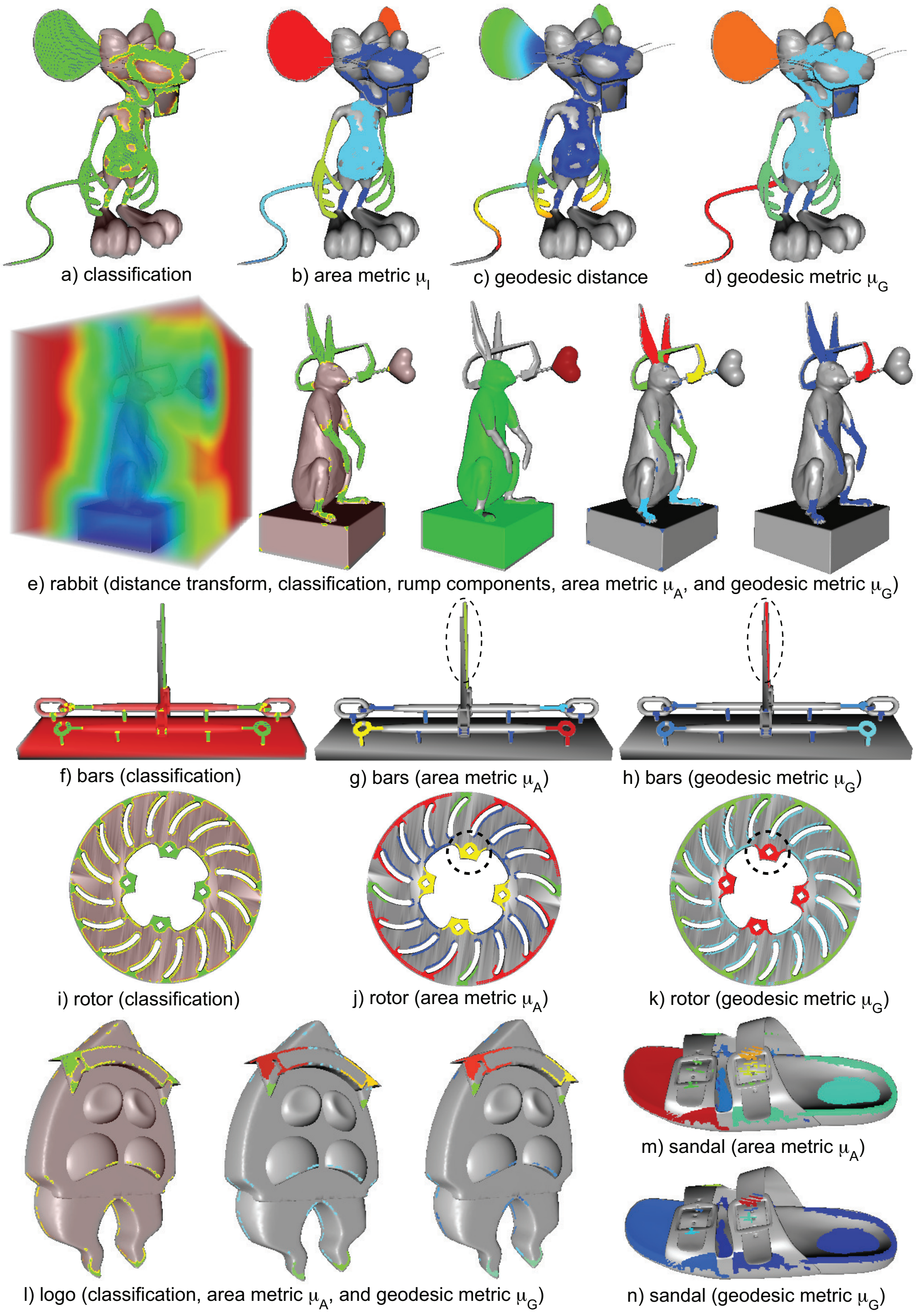
We have presented a solution for the task of automated assessment of 3D printability. For this, we combine a fast and robust method for thin region detection based on distance fields with two new application-specific metrics that uniformly treat all critical printability defects (spike and bridge removal and hole creation) and effectively capture critical events such as large surfaces having thin connections and elongated fragments. The entire pipeline requires only two user parameters: the printer resolution and maximal criticality of admitted defects, the latter which is an adimensional threshold value with intuitive border values. Our current implementation was tested in actual production runs at a 3D printing company and is currently forming the basis of an actual production tool for large-scale 3D printability assessment. In the future, we plan to extend our method in directly assessing 3D printability on mesh models, for optimal performance and accuracy, and also to design new metrics to capture more refined printability problems, such as specific topological and geometrical configurations in combination with specific material properties. Next, we plan to use our method in other application areas where shape thickness measurements are needed, such as 3D metrology for CAD/CAM and engineering.

Acknowledgements

We thank Robert Schouwenburg, CTO of Shapeways, for triggering this research, providing us with 3D printing details, and assessing our work, and Jacek Kustra (Philips Applied Technologies) for insightful technical discussions.

References

1. Roger Brockett and Petros Maragos. Evolution equations for continuous-scale morphological filtering. *IEEE Trans. Sig. Proc.*, 42(12):3377–3386, 1994.
2. T. Cao, K. Tang, A. Mohamed, and T. Tan. Parallel banding algorithm to compute exact distance transform with the GPU. In *Proc. ACM SIGGRAPH Symp. on Interactive 3D Graphics and Games*, pages 134–141, 2010.
3. R. Dougherty and K. Kunzelmann. Computing local thickness of 3D structures with ImageJ, 2007. Proc. Microscopy & Microanalysis Meeting, Ft. Lauderdale, Florida, Aug. 5-9, 2009, www.optinav.com/LocalThicknessEd.pdf.
4. Geometric Global. GeomCaliper tool, 2010. geomcaliper.geometricglobal.com.
5. J. Lambourne, D. Brujic, Z. Djuric, and M. Ristic. Calculation and visualisation of the thickness of 3D CAD models. In *Proc. SMI*, pages 107–112, 2005.
6. Petros Maragos. Differential morphology and image processing. *IEEE Trans. Image Processing*, 5(6):922–937, 1996.
7. P. Min. binvox voxelizer, 2010. www.cs.princeton.edu/∼min/binvox.
8. J. Mullikin and P. Verbeek. Surface area estimation of digitized planes. *Bioimaging*, (1):6–16, 1993.
9. F. Nooruddin and G. Turk. Simplification and repair of polygonal models using volumetric techniques. *IEEE TVCG*, 9(2):191–205, 2003.
10. D. Reniers and A. Telea. Tolerance-based feature transforms. In *Advances in Comp. Graphics and Comp. Vision (eds. J. Jorge* et al.*)*, pages 187–200. Springer, 2007.
11. Y. Sato, K. Nakanishi, H. Tanaka, T. Nishii, N. Sugano, H. Nakamura, T. Ochi, and S. Tamura. Limits to the accuracy of 3D thickness measurement in magnetic resonance images. In *Proc. MICCAI*, volume 2208, pages 803–810. Springer, 2010.
12. Shapeways. Online 3D printing, 2010. www.shapeways.com.
13. P. Stelldinger and K. Terzic. Digitization of non-regular shapes in arbitrary dimensions. *Image and Vision Computing*, 26(10):1338–1346, 2008.
14. A. Yezzi and J. Prince. An Eulerian PDE approach for computing tissue thickness. *IEEE Trans. Med. Imag.*, 22(10):1332–1339, 2003.



a) classification b) area metric μ_I c) geodesic distance d) geodesic metric μ_G

e) rabbit (distance transform, classification, rump components, area metric μ_A, and geodesic metric μ_G)

f) bars (classification) g) bars (area metric μ_A) h) bars (geodesic metric μ_G)

i) rotor (classification) j) rotor (area metric μ_A) k) rotor (geodesic metric μ_G)

m) sandal (area metric μ_A)

n) sandal (geodesic metric μ_G)

l) logo (classification, area metric μ_A, and geodesic metric μ_G)

Fig. 5. Printability assessment results on complex models in the 3D printing industry