2017 13th IEEE Conference on Automation Science and Engineering (CASE)
Xi'an, China, August 20-23, 2017

Adaptive Slicing for the FDM Process Revisited

Florens Wasserfall, Norman Hendrich and Jianwei Zhang
{wasserfall,hendrich,zhang}@informatik.uni-hamburg.de

***Abstract*— Adaptively computing the layer heights for 3D-printed parts has the potential to achieve high quality results while maintaining a reasonably short printing time. The basic concept, several error measures and variations of the algorithm have been around in the literature for two decades now, but never showed significant impact on widely used slicing software. Users of our early test implementations reported two major drawbacks of the existing approaches: the control measures are not intuitively usable and the resulting height distribution in many cases is not optimal for an object, requiring extensive post-editing.**

In this paper, we propose a more intuitive control measure and implementation based on the volumetric surface error and a subsequent manual refinement of layer heights by manipulating an interpolated height-curve. We describe the efficient computation of adaptive layers by analyzing the model surface over the full layer height. All implementations are available as ready-to-use open source software.

I. Introduction

Additive manufacturing processes are inherently subject to discretization effects. For most technologies, including the popular Fused-Deposition Modeling (FDM) process, generating an object from 2-dimensional slices involves a trade-off between printing time and quality, because thin layers result in finer objects but overall take longer to print. The discretization effects, as illustrated in figure 1 and 2 include:

Stairstepping Non-horizontal or -vertical surfaces are approximated by stacked layers, causing deviations between model- and print-surface, especially at shallow angles.

Surface roughness FDM-printed parts are not smooth even at vertical surfaces due to the elliptic surface of extruded traces. The roughness error increases with both layer height and surface angle.

Distortion Due to the stairstepping effect, most objects will loose or gain volume. Depending on the direction of the slope, the stairs are generated either inside or outside of the model surface [1], [2].

To address these issues, the concept of adaptive slicing was introduced, where the layer thickness is variable over the object height. All adaptive approaches aim to minimize the printing time while maintaining a small discretization error. Comprehensive surveys of general and adaptive slicing methods are given in [2], [3].

Several metrics for finding an optimal layer height for a given object geometry have been published. The different metrics can be roughly assigned to the three error types given above.

The authors are with the Department of Computer Science, University of Hamburg, Germany. This work was partly funded by the German Research Foundation in project Crossmodal Learning, TRR-169.

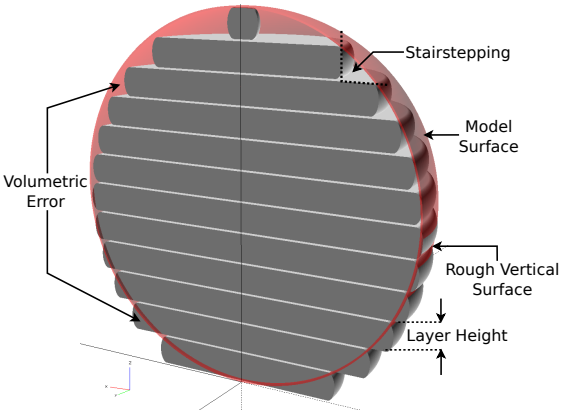

Fig. 1. Printing errors induced by layer-wise discretization. The model shape (red) is deformed by slicing at the top of each layer which causes a gain of volume at negative- and a loss at positive overhangs.

In this paper, we propose a new measure to control the layer height based on the volumetric error between printed- and model-surface which is both technically justified and intuitively usable (section IV). The efficient implementation for adaptive slicing is discussed in section V.

Section VI describes an approach to interactively adjust the layer thickness distribution after the adaptive slicing step by modifying a B-Spline based height curve, which gives the user a more precise and simple method of manipulation than entering a series of cusp values. The spline-based representation additionally applies a smoothing effect to attenuate the sudden thickness variations.

II. Related Work

Dolenc and Mäkelä [4] introduced the widely used *cusp height* measure, to control the *stairstepping* effect. The cusp height describes the maximum deviation between printed part

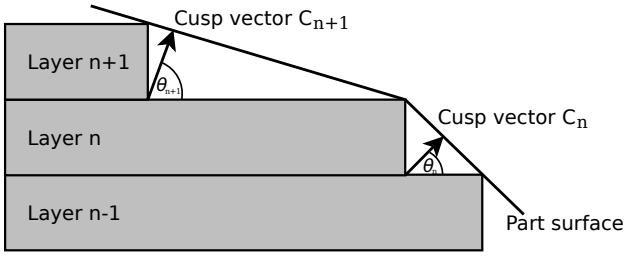

Fig. 2. Definition of the *cusp height* by Dolenc and Mäkelä [4]. The cusp vector C describes the maximum deviation between part surface and printed object, as caused by the stairstepping effect. The adaptive algorithm reduces the layer thickness such that C is maintained below a $C_{\max}$ specified by the user.

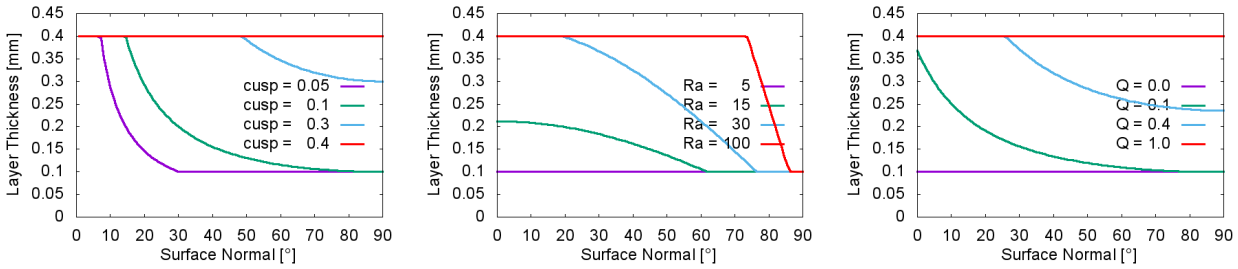

Fig. 3. Resulting layer thickness as a function of the surface normal ($\theta = 0° \Rightarrow$ vertical surface, $\theta = 90° \Rightarrow$ horizontal surface) for given control parameters, assuming minimal and maximal layer thickness of 0.1 mm and 0.4 mm respectively. The *cusp* measure (**left**) never generates flat layers for nearly vertical surfaces, while the R_a measure (**center**) never maximizes the layer thickness for horizontal surfaces. Our proposed measure Q (**right**) achieves both minimal and maximal layer thickness depending on the control value.

and model surface as illustrated in figure 2. The resulting layer height h_c is computed as

$$h_c = \min\{L_{\max}, C_{\max}/\sin\theta\} \tag{1}$$

where $L_{\max}$ is the maximum printable layer thickness, $C_{\max}$ is the user given cusp limit and $n_z = \sin\theta$ is z-component of the surface normal for a particular facet.

The cusp limitation can be partly relaxed to allow for a higher layer thickness for regions which are considered less important [5], [6]. To achieve this, the surface of an object is partitioned by analyzing the angle between each adjacent pair of facets. The user is then required to interactively enter a cusp limit for each partition.

The cusp metric was also utilized for parallel [7] and *local adaptive slicing* [8], where independent branches of an object are sliced individually depending on their surface geometry, and to adaptively refine the surface of an object while maintaining thick interior layers [9].

The surface *roughness* of FDM-printed parts was analyzed by Pérez [10] and Pandey et.al. [11]. They found the roughness R_a to be dependent on the slope of a surface and introduced a limitation of the layer thickness h_r by:

$$h_r = \frac{R_a \cos\theta}{70.82} \tag{2}$$

where R_a (µm) now specifies the maximum roughness as control parameter, and θ is the angle of the surface [12]. The constant 70.82 was empirically determined from the surface edge profile of printed objects. This roughness limitation was also used in an adaptive slicing implementation for the SLS process [13].

A variable layer thickness based on the *distortion* was computed by comparing the area of two adjacent slices, represented by their perimeter polygons [14]. This approach fails to detect slopes and features if positive and negative changes in the geometry cancel each other out, resulting in equal areas. This was improved by projecting each pair of contours to the XY, XZ and YZ plane [15] and computing the area of the deviation triangles at the borders.

Siraskar et al. recently proposed an octree data structure that represents the object by a 3-dimensional occupancy grid. A set of layer heights is then generated based on the local octree resolution [16]. While the approach is interesting and different from other approaches, the layer thickness is again fundamentally based on the slope of the surface.

III. Problem Statement

For the FDM process, none of the techniques introduced above is implemented in widely available software. We therefore decided to develop and contribute an adaptive slicing implementation to the existing toolpath generator *Slic3r* [17], mostly for two reasons: Slic3r is open source, and the internal representation of layers already was very suited to generate variable thicknesses. From the user feedback during the early release phase we identified a number of significant problems:

- The quality control parameters, particularly the cusp value, are technically well-founded but require a certain level of previous user knowledge, and the effect is not intuitively comprehensible and predictable.
- A single global control parameter was used for the entire object. Using a non-uniform cusp or roughness value as described above results in a cumbersome and confusing user interface, even more reliant on the non-intuitive parameters.
- In terms of surface quality, sudden transitions from thin to thick layers or vice versa are often perceived as worse than thicker layers over the entire object.

IV. Combined FDM Surface Quality Parameter

While specifying a $C_{\max}$ *cusp* or R_a *roughness* limit can be very useful from an engineering perspective where the user has an actual technical requirement for a resulting surface, the effect is not intuitively predictable for most users due to the nonlinear behavior. Also, an algorithm based on the cusp measure alone (eq. 1) will never generate small layer heights h_c for almost vertical surfaces regardless of the chosen cusp limit $C_{\max}$ (figure 3 left), making it impossible to achieve the highest possible printing quality only by modifying the cusp limit. Using the R_a limit instead suffers from an analog effect for almost horizontal surfaces (figure 3 center).

Therefore, we propose a new error measure Δ based on the actual object surfaces generated by the FDM process, which also takes into account that vertical surfaces can be significantly improved by reducing the layer height. As illustrated in figure 4, the surface quality of an FDM-printed part is mainly determined by two factors: the stairstepping

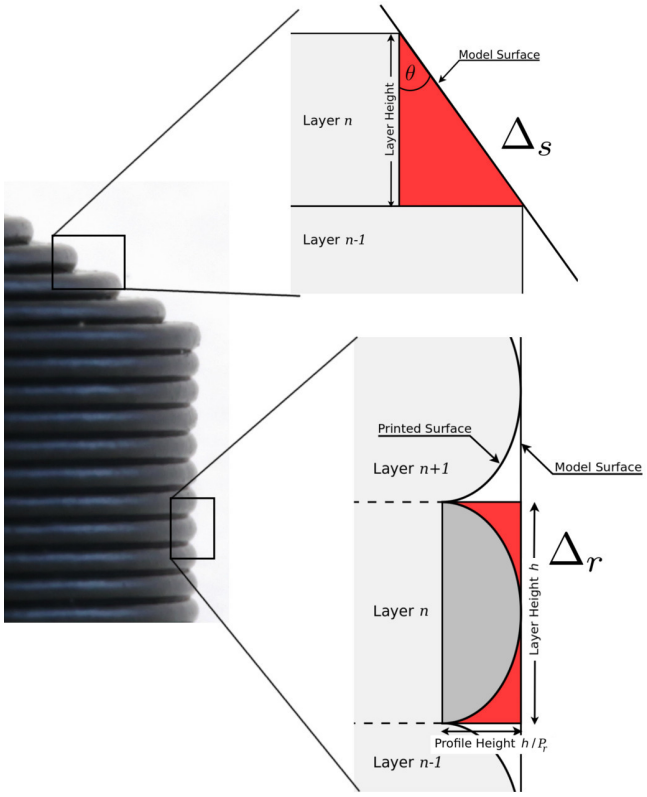

Fig. 4. Illustration of the two main sources of non-smooth surfaces: the stairstepping effect (**top**), quantified as the difference between ideal and printed surfaces (red triangle Δ_s) and the surface roughness induced by the circular nozzle tip (**bottom**), measured as the difference between ideal surface and elliptic extrusion line (red area Δ_r).

effect and the edge profile of the round, extruded plastic threads. While the stairstepping effect depends on the surface angle, the roughness error applies to all surfaces and solely depends on the layer thickness, as further explained in section IV-A. We therefore propose to use the sum of both effects as an error measure:

$$\Delta = \Delta_s + \Delta_r = \frac{sin\,\theta}{2} \cdot h + C_r \cdot h \tag{3}$$

Δ_s quantifies the area of deviation caused by the stairstepping effect and can be described by the triangle between printed- and model-surface as depicted in figure 4 (top). This factor corresponds to the cusp measure. Δ_r describes the area between printed- and model-surface which is not covered by plastic due to the extrusion profile (figure 4 bottom). As explained in section IV-A below, C_r is an empirically determined constant factor which describes the elliptic surface structure. Both error values are scaled by the layer height h to describe the absolute area deviation over the entire surface.

When using Δ as the control parameter, the layer height h of a given surface can be computed from equation 3 as

$$h \leq \frac{\Delta}{(\sin\,\theta)/2 + C_r} \tag{4}$$

As Δ still depends on the layer heights supported by a given printer, we further map Δ to a single intuitive quality parameter $Q = [0..1]$. This is done by finding the minimum and maximum possible Δ values for a given range of layer heights $[h_{\text{min}} .. h_{\text{max}}]$ according to the capabilities of a specific printer. The lower boundary is determined by vertical surfaces where $sin\,\theta = 0$ and therefore only depends on Δ_r:

$$\Delta_{min} = C_r \cdot h_{min} \tag{5}$$

The upper boundary is limited by horizontal surfaces where $sin\,\theta = 1$ and therefore computes as:

$$\Delta_{\text{max}} = \frac{h_{\text{max}}}{2} + C_r \cdot h_{\text{max}} \tag{6}$$

With equation 5 and 5 the scaled parameter Q_s is calculated as

$$Q_s = Q \cdot (\Delta_{\text{max}} - \Delta_{\text{min}}) + \Delta_{\text{min}} \tag{7}$$

Figure 3 illustrates the resulting layer heights according to different quality settings, where $Q = 0$ selects minimum layer heights, $Q = 1$ enforces maximum layer heights, and values in-between select smooth profiles according to the cusp values.

A. FDM Surface Profile

To model the actual surface profile of FDM-parts, we printed a set of specimens with layer thicknesses from 0.1 mm to 0.5 mm in steps of 0.1 mm and nozzle diameters of 0.25, 0.35 and 0.5 mm. The resulting profile was recorded with an optical camera and compared to different geometric primitives (figure 5). Overall, the semi-elliptic primitive showed the highest concordance with less then 10% average deviation between measured- and primitive area. The ratio of *layer height / profile height* was found to be independent of the extruder diameter. We determined $P_r = 3.3 \pm 0.53$, which is consistent with the value of $P_r \approx 3.3$ reported in [12, p. 64].

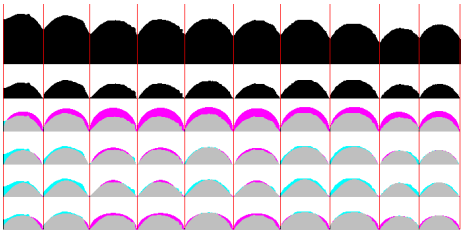

Fig. 5. Image processing steps to analyze the surface profile from figure 4. From top to bottom: **1, 2**: binarization and cropping of original image. Subsequent comparison against **3**: semi-circle, **4**: cropped semi-circle, **5**: sinus, **6**: semi-ellipse. Note that the semi-ellipse matches the data best, resulting in our model (equation 8).

Given the elliptic geometry, the area of deviation Δ_r for a single layer normalized over the layer height h can be described as:

$$\Delta_r = \frac{\overbrace{h \cdot \frac{h}{P_r}}^{A_{rect}} - \overbrace{\frac{1}{2} \cdot \frac{h^2\,\pi}{4\,P_r}}^{A_{ellipse}}}{h} = \frac{8-\pi}{8\,P_r} \cdot h = \overbrace{0.18403}^{C_r} \cdot h \tag{8}$$

V. Accelerated Adaptive Slicing

To compute the height of a layer, it is necessary to find all facets intersecting this layer. The search is accelerated by pre-sorting the facets by their lowest and highest Z-value. Our algorithm stores a reference to the last facet with Z_{bottom} below the current slicing plane. Finding the relevant facets is then done by iterating through the sorted list, starting from this reference and going to the first facet with Z_{bottom} above the slicing plane. The efficiency depends on the object: in the best case, only intersecting facets are analyzed for each iteration. In the worst case (one facet extends over the full object height), all facets are considered for each iteration.

Since the position of intersections is not previously known, adaptive slicing requires iterative computation of layer heights. For a given layer n, this is naively done by finding all facets which would intersect with the upper boundary of the previous layer n-1 and computing the height from the facet with the highest tilting angle. It is possible that a facet closely above the current slicing plane violates the quality requirement, so every facet intersecting a given layer at any height has to be analyzed, without knowing the actual height in the beginning of the process. This could be done by using the printer's maximal layer height as an upper limit and reducing the layer height during the process.

However, in some cases the result would be non optimal. Consider a case of a nearly horizontal facet directly on top of a nearly vertical facet as depicted in figure 6.

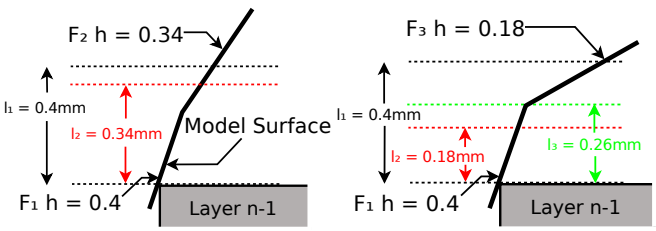

Fig. 6. Reduction of the layer height due to near facets with higher slopes. The height of layer n is first determined with 0.4 (l_1) at the top of layer n-1. In a second step, all facets touching the new layer are verified to be conformal with the error limit. Facet F_2 requires a maximum thickness of 0.34 mm, the layer is therefore reduced to this value (**left**). Facet F_3 would require a thickness of 0.18 mm which shrinks the layer to a height where it would not even touch the facet in question (**right**). To avoid this, the new layer height is set to the facets minimum at 0.26 mm.

In the left case, the lower value from Facet F_2 (0.34mm) is optimal. In the right case, the upper facet would reduce the layer height to 0.18mm (red line), resulting in a layer that would not even touch the facet in question.

To avoid this, the layer height should be reduced to the lowest point of F_3 (Z_{min}) only. To achieve this, our implementation computes the height limitation for every facet intersecting the upper boundary of layer n-1 in a first iteration and cycles through every remaining facet touching the layer from the first iteration in a second run, reducing the layer height to $\max(c, Z_{\text{min}})$ if necessary. Figures 7 and 10 show example objects printed with static layer heights and our adaptive slicing algorithm.

Fig. 7. Example for adaptive slicing. Bayonet coupling (**left**) and close-ups of the bayonet printed with static (**top right**) and adaptive (**bottom right**) resolution. In this application, the grade-smoothness is crucial for the function, while the optical surface quality is less important. The steps generated by non-minimal layer height will prevent the inner and outer parts from gliding into the lock-position. Note that automatic adaptive slicing generates thin layers only where required, while maximal layer height h_{max} is used for the vertical structures below and above the actual bayonet part to improve printing speed.

VI. Interactive Layer Refinement

In some cases, the result of adaptive slicing is not optimal: e.g. for regions with small, sloped artifacts, sudden transitions between different layer heights are optically unpleasant. Also, the layer thickness may have negative side effects on the resulting print. A typical case is shown in figure 8, where an overhanging structure is on the same layer as a tilted surface. Therefore, our algorithm selects a thin layer height, but the resulting thin filament strand is prone to drop a bit. In this case, locally increasing the layer thickness actually improves the result.

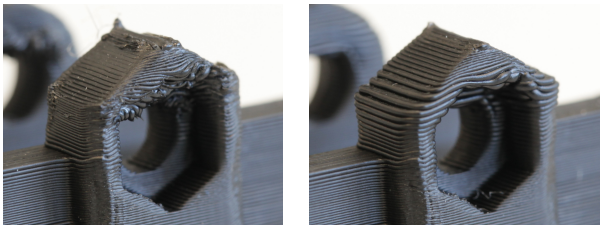

Fig. 8. Example of a geometry with unsupported overhangs where high quality slicing has a negative effect due to underextrusion at "hovering" lines (**left**), where the high surface angle of the overhang causes the adaptive algorithm to minimize the layer thickness. Locally relaxing the quality requirement causes higher stairstepping but significantly improves the geometric accuracy (**right**).

Therefore, a more detailed user control beyond the selection of the global quality parameter Q is needed. Figure 9 shows our interactive layer height control tool. Here, the left (main) part of the application window shows the 3D preview of the object, while the right window plots the corresponding layer thickness (x-axis) distribution over the height of the object (y-axis). Each line corresponds to an actual layer. The layer height distribution can be initialized either by the adaptive slicing algorithm or by a static layer height. The thickness can be locally modified by dragging the curve to the left or right with a quadratic (left click) or linear (right click) modifier. The range of modified layers is selected (widened or narrowed) by vertical mouse movements.

A. Spline Based Height Interpolation

A fundamental problem of layer height manipulations is to maintain the position of features above the modification.

Changing the thickness of a layer will shift the positions of all higher layers and this effect cumulates over multiple modifications. To solve this, the height distribution is described by a B-spline, initialized with the set of layers computed by the adaptive slicing algorithm. The B-spline provides a set of spatially stable interpolation points.

User interactions will change the value, but not the position of the affected interpolation points. The resulting set of layer heights is then computed by iteratively querying the B-spline for its value at the upper boundary of the previous layer. Using a B-spline representation additionally has a slight smoothing effect and therefore reduces sudden layer height changes.

B. Layer Height Gradation

A low resolution or imprecise motor microstepping at the Z-axis can cause artifacts if the height of a layer cannot be exactly matched by the printer. A small deviation between layer height and printer resolution results in aliasing effects when the error accumulates over several layers until the rounding effect inserts or removes an additional step. The resulting layer is slightly taller or shorter, but the amount of extruded plastic is constant leading to an over- or under-filled layer. This effect can be easily avoided with a static layer height by setting the height to a multiple of the Z-axis step-size, which is not possible for adaptive layer generation. This is solved by an optional layer height gradation parameter, which provides the printer's Z-resolution. The layer height is rounded to a multiple of the resolution after reading the height value from the B-spline. This feature also works for static slicing.

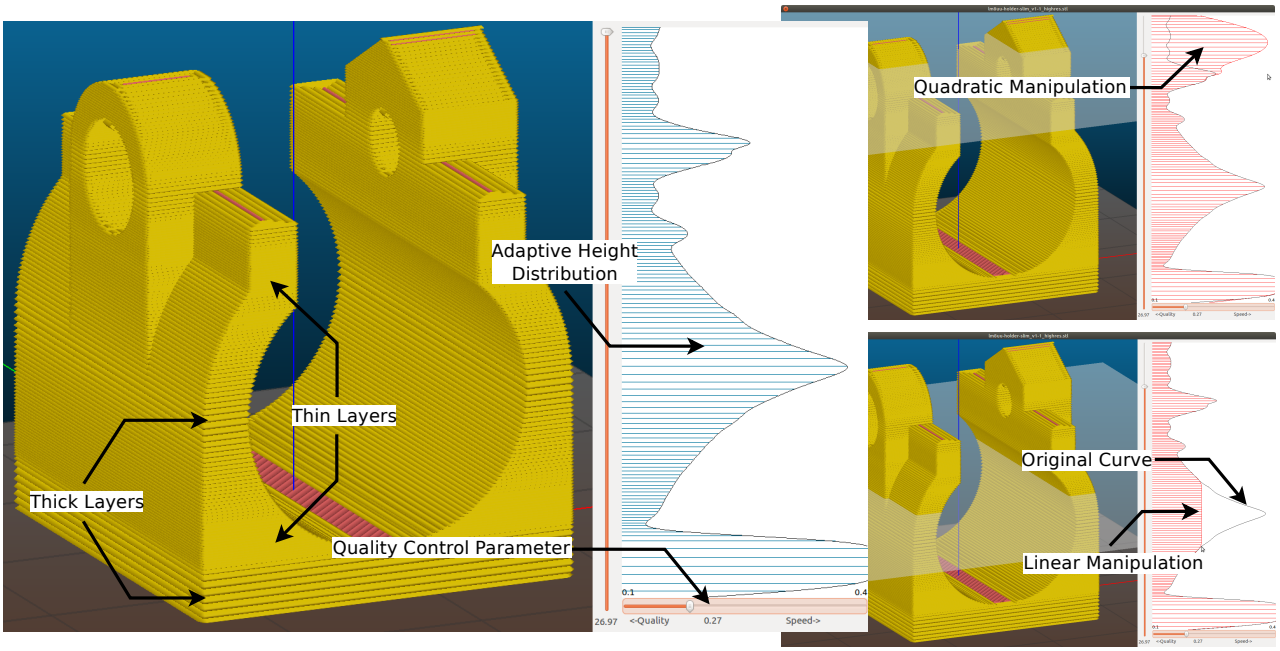

Fig. 9. Intuitive user Interface for adaptive slicing and layer refinement [18]. The Quality value is set by a slider between maximum quality and maximum printing speed. The layer height distribution is plotted vertically next to the preview. Local height changes are possible by dragging a quadratic (**top**) or linear (**bottom**) manipulator with a left or right mouse-click directly in the plot.

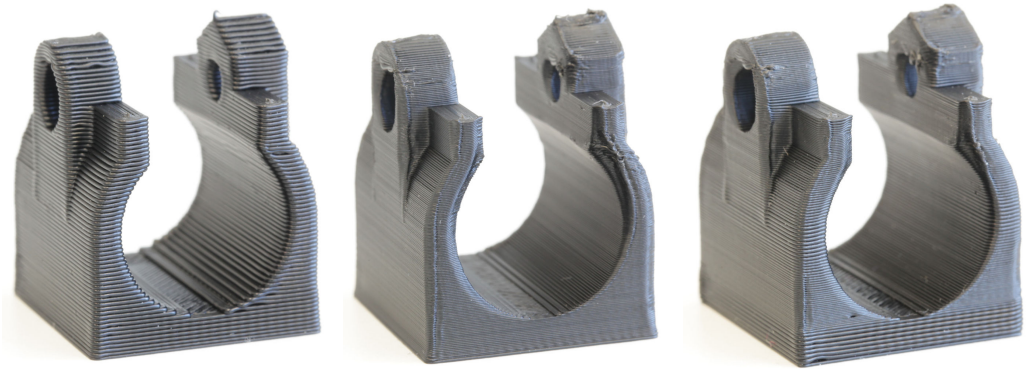

Fig. 10. Bearing holder object, printed with maximum (**left**), minimum (**center**) and adaptive layer height (**right**)

VII. Evaluation

We processed and printed a number of test objects to evaluate the performance of the slicing algorithm and the print quality.

A. Performance of the Implementation

The performance of layer generation highly depends on the number of facets of a model. Slic3r parallelizes many tasks by computing on a per-layer basis. Unfortunately, this is not easily possible for the geometry analysis (section V), since the positions of higher layers depend on the height of their predecessors.

TABLE I
Runtime of the slicing algorithm for different objects

Object	Configuration	Runtime [s]	Adaptive [%]
Bearing clamp	Static [164 Layers]	0.18	-
740 Facets	Adaptive [163 Lay.]	0.19	5.3
Handle	Static [192 Layers]	1.03	-
13,020 Facets	Adaptive [193 Lay.]	1.36	24.26
3DBenchy	Static [295 Layers]	4.21	-
225,154 Facets	Adaptive [295 Lay.]	4.60	8.48

Table I shows the processing time for three typical objects with increasing surface complexity and resolution. For better comparability, the static layer height was set such that the resulting number of layers is equal to the adaptive generation. The last column measures the fraction of CPU time spent for adaptive layer generation as a percentage of the runtime of the full slicing process; as can be seen, the overhead for adaptive slicing is often insignificant. Slic3r was originally written in Perl and partly ported to C++ to increase the performance. The surface analyzer is fully ported to C++ which massively improved the performance compared to our earlier implementations.

B. Printing Time and Quality

Figure 10 shows the same object printed with minimum (45 min), maximum (99 min) and adaptive resolution (74 min). The overall printing time is highly dependent on several factors. Two important parameters are the extruder diameter and the infill generation.

Slic3r implements a feature called “microlayering”, where only the perimeter is printed with high quality and the interior (infill) is dynamically combined over multiple layers, similar to the approach described in [9]. Therefore, the effect of adaptive slicing highly depends on the ratio of volume to surface and the use of microlayering. The print time of objects with a large surface but low volume is determined by the generation of perimeters, thus the benefit from adaptive slicing is higher.

VIII. Conclusion and future work

This paper describes the implementation of a novel adaptive slicing algorithm for a broad use with common FDM-printers. A focus lies on the usability of a rather complex concept for users without extensive previous knowledge. To achieve this, we introduced a new error measure based on the volumetric surface deviation between model and printed object, which allows for an intuitive, yet optimal control between quality and printing time.

However, as explained in section VI, the automatic adaptive slicing is not optimal in all situations. To overcome this, a B-spline based graphical control element was developed which permits an interactive refinement of the layer height distribution to adapt the precomputed solution to individual use-cases.

All implementations are integrated into a widely used slicing software. The program and source code are available under an open source license at:

https://tams.informatik.uni-hamburg.de/research/3d-printing/slicing/

Our future work will focus on a better user interface and integration into slic3r. This includes a color based visualization of layers in the 3D-preview and additional manipulators for interactive layer refinement e.g. Bézier or Gaussian curves.

References

[1] K. Tata, G. Fadel, A. Bagchi, and N. Aziz, “Efficient slicing for layered manufacturing,” *Rapid Prototyping Journal*, vol. 4, 1998.

[2] P. M. Pandey, N. V. Reddy, and S. G. Dhande, “Slicing procedures in layered manufacturing: a review,” *Rapid Prototyping Journal*, vol. 9, 2003.

[3] H. H. Nadiyapara and S. Pande, “A Review of Variable Slicing in Fused Deposition Modeling,” *Journal of The Institution of Engineers (India): Series C*, vol. 98, no. 3, pp. 387–393, 2016.

[4] A. Dolenc and I. Mäkelä, “Slicing procedures for layered manufacturing techniques,” *Computer-Aided Design*, vol. 26, no. 2, 1994.

[5] D. Cormier, K. Unnanon, and E. Sanii, “Specifying non-uniform cusp heights as a potential aid for adaptive slicing,” *Rapid Prototyping Journal*, vol. 6, no. 3, pp. 204–212, 2000.

[6] M. Y. Zhou, J. T. Xi, and J. Q. Yan, “Adaptive direct slicing with non-uniform cusp heights for rapid prototyping,” *International Journal of Advanced Manufacturing Technology*, vol. 23, pp. 20–27, 2004.

[7] E. Sabourin, S. A. Houser, and J. H. Bøhn, “Adaptive slicing using stepwise uniform refinement,” *Rapid Prototyping Journal*, vol. 2, no. 4, pp. 20–26, 1996.

[8] J. T. Tyberg, “Local adaptive slicing for layered manufacturing,” 1998.

[9] E. Sabourin, “Adaptive high-precision exterior, high-speed interior, layered manufacturing,” 1996.

[10] C. J. L. Pérez, “Analysis of the surface roughness and dimensional accuracy capability of fused deposition modelling processes,” *International Journal of Production Research*, vol. 40, 2002.

[11] P. M. Pandey, N. V. Reddy, and S. G. Dhande, “Improvement of surface finish by staircase machining in fused deposition modeling,” *Journal of Materials Processing Technology*, vol. 132, 2003.

[12] P. M. Pandey, N. V. Reddy, and S. G. Dhande, “Real time adaptive slicing for fused deposition modelling,” *International Journal of Machine Tools & Manufacture*, vol. 43, 2003.

[13] S. K. Singhal, P. K. Jain, and P. M. Pandey, “Adaptive slicing for SLS prototyping,” *Computer-Aided Design and Applications*, vol. 5, no. 1-4, pp. 412–423, 2008.

[14] Z. Zhao and L. Laperrière, “Adaptive Direct Slicing of the Solid Model for Rapid Prototyping,” *International Journal of Production Research*, vol. 38, no. 1, pp. 69–83, 2000.

[15] M. T. Hayasi and B. Asiabanpour, “A new adaptive slicing approach for the fully dense freeform fabrication (FDFF) process,” *Journal of Intelligent Manufacturing*, vol. 24, pp. 683–694, 2013.

[16] N. Siraskar, R. Paul, and S. Anand, “Adaptive Slicing in Additive Manufacturing Process Using a Modified Boundary Octree Data Structure,” *Journal of Manufacturing Science and Engineering*, vol. 137, no. 1, 2015.

[17] A. Ranellucci. Slic3r Project Website. [Accessed 09.02.2017]. [Online]. Available: http://slic3r.org

[18] F. Wasserfall. Adaptive Slic3r Project Website. [Online]. Available: https://tams.informatik.uni-hamburg.de/research/3d-printing/slicing/