

3D Printing of Nonplanar Layers for Smooth Surface Generation

Daniel Ahlers, Florens Wasserfall, Norman Hendrich and Jianwei Zhang
Computer Science Dept., University of Hamburg, Germany
{ahlers,wasserfall,hendrich,zhang}@informatik.uni-hamburg.de

Abstract—Additive manufacturing processes are inherently subject to discretization effects. For most technologies, stair-stepping artifacts impair the surface quality of 3D printed objects, especially when the surface slope is close to horizontal.

In this paper we propose a novel Fused Deposition Modeling (FDM) slicing approach that combines nonplanar and planar layers, increasing printing quality and resulting in smoother, stronger object surfaces. Our slicing algorithm automatically detects which parts of the object should be printed with nonplanar layers and uses a geometric model of the printhead and extruder to generate collision-free toolpaths.

Our open source implementation is based on the popular Slic3r tool and can be used on all common three-axis 3D printers. We present typical printing results and compare surface quality as well as slicing and printing times with traditional and adaptive planar slicing.

I. INTRODUCTION

The fact that layer-based additive manufacturing processes are inherently subject to discretization effects is widely known and has been addressed by many researchers for more than two decades. Fig. 1 illustrates the effect of stair-stepping on the surface quality of FDM-printed objects in general and shows how it becomes increasingly dominant in regions with only a small surface inclination. Stair-stepping not only affects the aesthetic perception but can also influence the mechanical properties, e.g., the aerodynamic performance of a wing as demonstrated in Fig. 9. Several attempts have been made to mitigate the effects of stair-stepping and improve the surface quality. They roughly categorize into:

Adaptive slicing: Stair-stepping artifacts can be significantly reduced by increasing the z -resolution, but this also drastically increases the required print time. Adaptive slicing is an attempt to automatically balance the trade-off between printing speed and quality by reducing the layer thickness

only at regions where the surface geometry requires a higher resolution. This is typically directly related to the local surface angle of an object.

Post-processing: The surface of a 3D-printed object can be smoothed mechanically or chemically with an additional post-processing step. However, this typically requires substantial CNC-milling machinery and manual work and is not further considered in this document.

Nonplanar printing: With nonplanar layers, the extrusion precisely follows the actual surface contour of an object instead of slicing it into planar layers. Almost perfect surfaces can be achieved particularly for smooth curvatures as demonstrated in Fig. 2. However, tool path generation (G-code) for nonplanar layers is a difficult, partly unsolved problem. Only surfaces with a 1D-curvature are printable with linear extrusions without distortion and 5-axis machinery is required for the general case. Self-collisions and transitions between different nonplanar surfaces are additional major problems.

Since nonplanar layers are particularly important for “flat” surface regions which suffer strongly from stair-stepping, we attempt to identify such regions, check for collisions and automatically generate a mixed tool path with mostly planar layers and nonplanar patches at the surface.

The rest of this paper is structured as follows. First, Sec. II summarizes the relevant previous work on adaptive slicing and curved-layer toolpath approaches for FDM printing. Sec. III explains how to identify those parts of the given object mesh suitable for nonplanar printing, while Sec. IV introduces our nonplanar slicing algorithm. Collision avoidance, described in Sec. V, is the key step to make

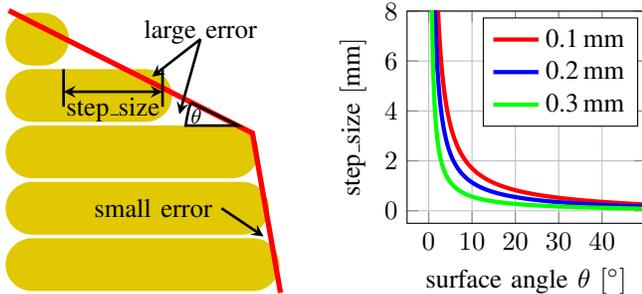


Fig. 1. Illustration of the stair-stepping effect for FDM-printed surfaces (left) and $step_size$ for different layer heights (right). Note that the stair-stepping length increases drastically for surface angles $\theta < 20^\circ$.



Fig. 2. FDM 3D-printed relief. **Left:** traditional slicing showing strong stair-stepping artifacts, **Right:** nonplanar slicing with smooth upper surface.

the algorithm work on common three-axis machines and the popular slicing software Slic3r [1]. Example results are presented in Sec. VI. The paper concludes with a summary and an outlook on future work.

II. RELATED WORK

Several approaches have been proposed to model and measure the surface quality of 3D-printed objects. Dolenc and Mäkelä [2] introduced the widely used *cuspl height* measure, which describes the maximum deviation of a simplified rectangular stair-step profile from the ideal object surface. The cusp metric was later utilized for an entire class of adaptive slicing algorithms, including local relaxation [3], [4], parallel [5] and local adaptive slicing [6], where independent branches of an object are sliced individually depending on their surface geometry. Adaptively refining the surface of an object while maintaining thick interior layers was proposed in [7].

A second approach to quantify the surface quality of FDM-printed objects was introduced by Pérez [8] and Pandey et al. [9] where the well-known *surface roughness* measure R_a was related to the surface angle. In our previous work, we defined a combined error measure which considers both the effects of stair-stepping on near-horizontal surfaces and the distortion of mostly vertical surfaces due to the round extrusion profiles [10].

The stair-stepping effects can be reduced with multi-directional toolpaths in a single object [11], [12]. This is done by first decomposing the model into different parts, where every part is then sliced with a different suitable orientation, and parts are finally merged into a toolpath for the whole object. The multi-directional object can be printed with larger overhangs and improved surface quality.

Curved Layer Fused Deposition Modeling supports varying z -heights on a single printing layer. Chakraborty et al. introduced this method and suggested first algorithms [13]. In their work, the surface is defined as a parametric surface, and the toolpath is generated along this surface. They also defined three key factors for the printability of curved layers: proper toolpath generation, extrusion orientation, and bonding between adjacent extrusions. Although the algorithms were presented, they were not used to create real physical objects. First real prints were done by Huang et al. [14] while still using parametric surfaces (and not facet models as are the de-facto standard). Parts with printed curved layers that follow the actual geometry of the designed object were introduced by Singamneni et al. [15] where the surface is rasterized with a script and the actual toolpaths are generated along those points.

Since not every surface is printable with a curved surface, curved layers have to be combined with planar layers [16], [17], [18]. The printable regions are identified in the surface mesh and offset to the inside of the object. The offset part is sliced nonplanar with curved layers and the rest of the model is sliced with planar layers.

Collisions within the toolpath are a big problem for printing nonplanar layers. Micali et al. [19] presented an

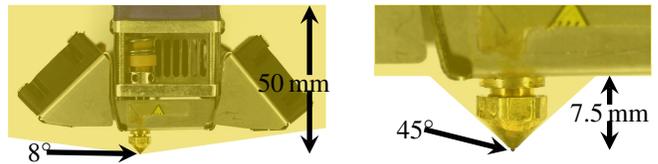


Fig. 3. The collision model of the Ultimaker 2 printer [23]. **Left:** taking the whole printhead into account where $\theta_{np} = 8^\circ$ and *nonplanar.height* is 50 mm. **Right:** taking only the nozzle into account with an $\theta_{np} = 45^\circ$ and *nonplanar.height* of 7.5 mm. With these configurations, either large surfaces with a small θ_{np} or small surfaces with a large θ_{np} can be printed.

algorithm that can generate a 3D toolpath along a free-form surface and check the toolpath for collisions by generating an envelope surface with an approximated pointy nozzle. However, the algorithm only checks for collisions inside the surface that should be printed and was not tested on a real print. When printing with a three-axis printer, the nonplanar extrusion is not laid down orthogonally to the printed part. Jin et al. [20] showed that the nozzle is either a bit too high or too low regarding the printing direction. They presented a method that compensates this dynamically according to the printed slope and direction of the current path.

Khurana et al. [21] tested the mechanical properties of either planar and nonplanar printed objects. Nonplanar printed parts were both stronger and stiffer than their regular printed counterparts. Kubalak et al. [22] showed that the layer bonding can be increased by printing a reinforcement shell onto the outer surface of the object to shift the mechanical stress away from the layer bondings.

III. IDENTIFICATION OF PRINTABLE SURFACES

For a general purpose implementation of a non-planar slicing algorithm, it is important to automatically identify surfaces where non-planar layers are both technically printable and a significant quality improvement is anticipated.

A. Self-Collisions

Self-collisions with the nozzle, extruder body, or cooling fans of the printhead potentially occur when the printhead applies material on inclined surfaces below the current printing layer, where material has already been deposited on the previous layers.

A simplified printhead model is introduced to prevent those collisions (Fig. 3). Only two parameters define the model: the maximum nonplanar angle θ_{np} and the maximum *nonplanar.height* that are printable without collisions. The maximum nonplanar angle is collision free in any direction above the given nozzle tip. The *nonplanar.height* is the maximum printable height of the surface that can be printed with the configured angle without collisions with any other part of the printer. This typically either is the gantry system with a low θ_{np} and high distance, or the extruder body with a very high θ_{np} but only very limited vertical clearance (Fig. 3). Both parameters are different for every printing platform and are therefore implemented as configuration parameters which must be provided by the user.

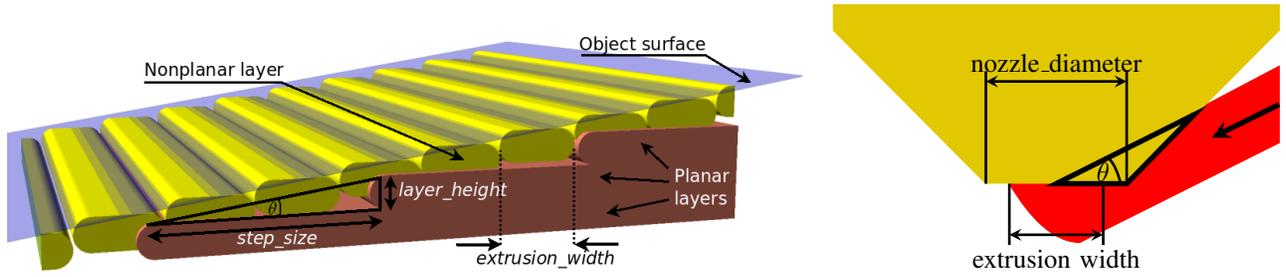


Fig. 4. Effect of the object surface angle θ on the quality of nonplanar printed layers. **Left:** nonplanar extrusions along the inclination (yellow) on top of planar layers (brown) significantly reduce the stair-stepping effect for small angles. **Right:** if the nozzle moves perpendicular to the inclination, the plastic is slightly squished (down) or expanded (up) by the nozzle tip, depending on the movement direction.

B. Surface Inclination and Quality

The quality of a nonplanar inclined surface printed with a three-axis (xyz) gantry system is also influenced by the surface angle, but the effect is inverse to the stair-stepping effect; almost horizontal surfaces are printed with very high quality. The surface roughness is mainly determined by two effects which can be described independently by considering only movements along or perpendicular to the inclination.

In the first case, stair-stepping occurs at the highest planar layers and is “subsampled” by the first nonplanar layer as illustrated in Fig. 4 (left). The resolution of the nonplanar layers is higher than the same geometry printed with planar layers if the *extrusion_width* is smaller than the *step_size*. This implies that for a given *extrusion_width* and *layer_height* the threshold surface inclination computes as:

$$\theta_{th} = \arctan\left(\frac{layer_height}{extrusion_width}\right) \quad (1)$$

Regions with a surface angle $\theta < \theta_{th}$ should therefore be printed nonplanar. This limits the angle for the region classification to

$$\theta_m = \min(\theta_{np}, \theta_{th}) \quad (2)$$

In the second case, the surface deformation is caused by the angle between the nozzle orifice and the surface. During a horizontal movement, the plastic thread is sheared off by the rim of the orifice, resulting in a flat surface. For non-horizontal movements, the orifice becomes increasingly “ellipsoid”, leaving a convex or concave profile, depending on the movement direction (up or down) as sketched in Fig. 4 (right). While it is generally possible to predict the resulting extrusion profile, it depends on the exact nozzle geometry and we, therefore, forgo a detailed analysis at this point.

C. Region Classification

To actually get regions from the tessellated STL mesh that meet the previously defined criteria, all facets that have a smaller angle than the threshold angle θ_m are stored. All these facets are then grouped into connected components by recursively marking all neighbors of each triangle. Each connected component forms a nonplanar surface that is printable with the defined maximum printing angle θ_m . All surfaces are checked whether the difference between the

minimum and the maximum height of all facets is greater than the *nonplanar_height* to ensure that none of the surfaces exceeds this limit. To ensure that none of the surfaces exceed the *nonplanar_height*, each surface is checked whether the difference between the minimum and the maximum height of all facets is greater than the *nonplanar_height*. All surfaces which exceed this limit are removed from the nonplanar surfaces list. Since this approach also finds small areas that sometimes only contain one facet, all surfaces with a surface area smaller than 20 mm^2 are also removed from the list. Although each area itself is not causing collisions while printing, the extruder can collide with previously printed structures close to the nonplanar surface. In a final step, each surface that causes a collision is also removed from the list of potential nonplanar surfaces. The *object collision avoidance* with other regions of the object is explicated in detail in section V.

IV. NONPLANAR TOOLPATH GENERATION

The general idea of the nonplanar toolpath generation is to extend planar slicing so that nonplanar layers replace regular ones at the top of printed objects. All other regions are sliced and printed with the regular planar slicing implementation. The regions that are nonplanar printable are moved to the highest possible layer and get warped down onto their intended position after the toolpath generation.

A. Layer Generation

The object model is first sliced into horizontal slices to generate the layers. These layers are usually evenly distributed along the z -axis of the printed object. On each layer, multiple polygons are generated which represent the outline of this layer. The polygon points are created where the horizontal layer cut intersects with any edge of facets. The horizontal slices and one exemplary layer polygon are illustrated in Fig. 5 (left). The planar layers now also contain the area where nonplanar layers should be printed. To replace these, all areas that are potential top surfaces and lay below the found nonplanar surfaces are marked (Fig. 5(center)). The potential nonplanar layer areas *PNL* are found with the following polygon operation where L_n is the layer area, and *NS* is the 2D projection of the nonplanar surface mesh.

$$PNL = (L_n \setminus L_{n+1}) \cap NS \quad (3)$$

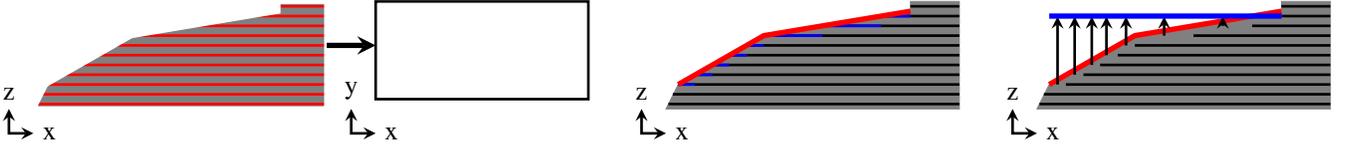


Fig. 5. The three steps of the layer generation **Left**: generation of layers and the individual layer polygons **Center**: identification of potential nonplanar areas **Right**: moving potential nonplanar layer to the highest layer.

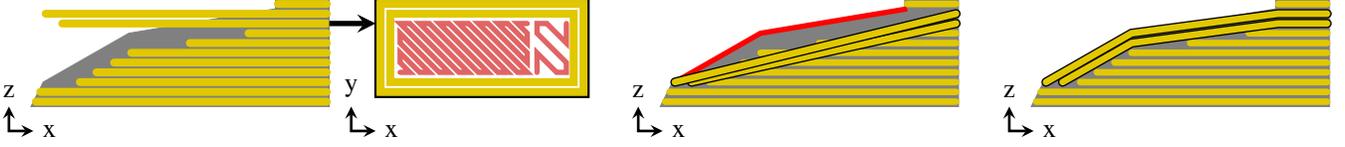


Fig. 6. The three steps of the nonplanar toolpath generation **Left**: planar toolpath generation on the layers **Center**: projection of the existing points of the path **Right**: calculating new points on facet intersections.

All these marked areas are then moved to the highest layer where potential top surfaces are found for the currently processed nonplanar surface (Fig. 5 (right)). Since usually top surfaces contain more than one shell layer, additional nonplanar shell layers are generated similarly by repeating the process and moving the found areas to the layer below the previous one. The topmost nonplanar surface area is marked as `stTopNonplanar` and the internal shell areas are marked as `stInternalSolidNonplanar` as they might require individual printing parameters.

B. Toolpath Generation

The moved layers are now planar layers that float above their intended position in midair. As they are still planar, the standard planar perimeter and surface filling algorithms are used to fill those regions with planar toolpaths. The perimeter toolpaths are generated by offsetting the outline polygon one half extrusion width to the inside. This step is repeated until the desired number of perimeters is generated. Next, the surfaces of the layer are classified into their different types. The surface type sets the printing parameters like speed or filling ratio. Since the nonplanar surfaces are already classified, their classification stays untouched. Also, all regions below them are classified as internal surfaces rather than top surfaces. All top layer layers are then filled with a continuous filling pattern as usual (Fig. 6 (left)).

The three-dimensional tool path is generated out of the previously created two-dimensional toolpath by projecting down every extrusion path that lays on a layer with an attached nonplanar surface. Extrusion paths are generated out of multiple two-dimensional points. By adding a z -component to each point of the path, it transforms to a three-dimensional extrusion path. Each point of the path that lies above a nonplanar surface is vertically projected downwards to the height of the nonplanar surface mesh to get the extrusion path to follow the actual surface geometry of the printed object. All facets of the original surface mesh are checked against every point of the extrusion path to find the corresponding facet for every point. Each point is projected downward onto its corresponding facet. The

intersection point replaces the planar point in the extrusion path. Since the x - and y -component stay the same, only the z -component is added to the point. The process is repeated for each shell layer with an additional z -offset of one layer height. The resulting tool path is shown in Fig. 6 (center).

The endpoints of each extrusion line now match the object surface, but the lines are not following the curvature yet. Additional points have to be added at each facet edge of the tessellated model. To achieve this, the intersections of an extrusion line with all facets calculated in planar two-dimensional space and new points are inserted and projected to the correct z -height at each intersection. Since the facets are not sorted, all newly generated points must be sorted along the direction of the currently processed extrusion line before they are inserted. The process is repeated for the additional shell layers. Fig. 6 (right) illustrates the three-dimensional toolpath that follows the surface geometry of the nonplanar surface mesh.

To print the nonplanar toolpath on an actual 3D printer, the G-code is generated. The nonplanar G-code generation is similar to the G-code generation of planar toolpaths. The individual extrusion paths are chained together to a layer extrusion path. All layers are then chained together bottom up to an extrusion path of the whole object. The extrusion amount is calculated individually for each line by the Euclidean distance between the current and the next point multiplied by the filament flow factor. The shell surfaces are simply shifted vertically along the z -axis and not along the individual facet-normals like Huang and Singamneni [16] did. So with a rising surface angle, the distance between the two layers becomes smaller than the intended layer thickness. If not compensated, this would lead to overfilling of the nonplanar surface, since the flow factor is set for the configured layer thickness. So the extrusion amount of each line is multiplied by a correction factor m :

$$m = \cos \left(\arctan \left(\frac{(P_2.z - P_1.z)}{\text{length}(P_1, P_2)} \right) \right) \quad (4)$$

where P_1 and P_2 are the two points of the extrusion line, $\text{length}()$ is the Euclidean distance of both points. The gen-

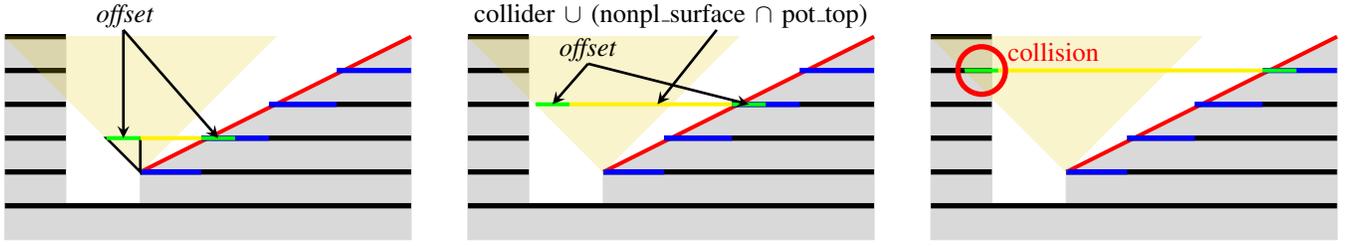


Fig. 7. The layer based collision checking. The light yellow nozzle with the θ_{np} angle is just for visualization and is not part of the algorithm. Layers are black, the nonplanar surface mesh is red, the new collider is yellow and the offset is green. The algorithm iterates through the layers and checks each layer individually with the collider from the previous iteration. It stops when the last topmost layer is collision free or a collision is found.

erated G-code can be printed on a common 3D printer; there are no further modifications necessary beside the correct configuration of the θ_{np} and the *nonplanar_height*.

V. COLLISION AVOIDANCE

Algorithm 1 Layer based collision check

```

1: function CHECK_NONPLANAR_COLLISIONS(surface)
2:   Polygon collider
3:   Polygon nonpl_surface = PROJECTION(surface)
4:   offset =  $\frac{\text{layer\_height}}{\tan \theta}$ 
5:   for all layers below nonpl_surface do
6:     layer_collider = collider - nonpl_surface
7:     if layer.surface  $\cap$  layer_collider  $\neq$  empty then
8:       return collision found
9:     else
10:      pot_top = layer.surface - upper_layer.surface
11:      all_surfaces = nonpl_surface  $\cap$  pot_top
12:      new_collider = collider  $\cup$  all_surfaces
13:      collider = OFFSET(new_collider, offset)
14:    end if
15:  end for
16:  return no collision found
17: end function

```

Collisions cannot occur on planar slicing due to the strict order of the printed layers. The printhead always travels upwards and will never revisit a lower layer. When printing nonplanar layers, however, collisions can occur because the printhead drives down into already printed layers. Since the printer cannot actively detect those collisions, they have to be avoided during the toolpath generation.

Collisions within the nonplanar region itself are impossible because each facet in the nonplanar surface has an angle that is smaller or equal θ_{np} as described in Sec. III, only planar areas on higher layers can cause collisions. This can be prevented by checking the whole nonplanar tool path for collisions. When a collision would occur, the surface is removed and printed planarly instead.

The layer-based collision checking algorithm (Alg. 1) is used to test for these collisions. The collision checking starts from the lowest layer which is touched by a particular nonplanar layer. Collisions within the nonplanar surface region are always ignored because these are false positives.

On every layer, the intersection of a collider polygon and the current layer polygon is calculated. If this intersection is not empty, there is a collision, otherwise; the intersection of the current layer and the nonplanar surface is added to the collider. Then this collider is offset by the width that the collision model would gain within one *layer_height*. The *offset* is calculated by the following formula where θ_{np} is the maximum nonplanar angle.

$$\text{offset} = \frac{\text{layer_height}}{\tan \theta_{np}} \quad (5)$$

This new extended collider is then used for the collision check on the next layer above. Fig. 7 shows the growth of the collider over several layers, the collider cone is just for visualization and not part of the algorithm. When all layers are collision free, the whole extrusion path of this nonplanar layer is collision-free within the current object. The algorithm does not check for collisions with other structures on the printed bed like models, support or a high skirt because these structures are not generated when the collision checking is done.

Collisions can also occur when traveling from or to a point which lies below the highest printed layer as illustrated in Fig. 8 (left). A simple way to avoid these collisions is to always lift up the printhead to the current maximum printing height, travel to the desired position and lower the printhead again (Fig. 8 (right)). Because travels also occur when switching from one perimeter to another, and because it is very unlikely that collisions occur on very short paths, all travel moves that are shorter than $2 \times \text{extrusion_width}$ are traveled directly without lifting the printhead. For all other travel movements, the printhead is lifted even if no collision would occur.

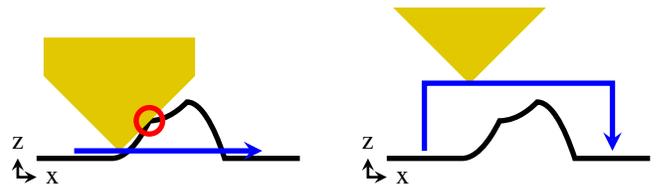


Fig. 8. **Left:** Traveling directly to the target point can cause collisions with previously printed structures. **Right:** Moving up to the current layer height before traveling avoids collisions with previously printed structures.

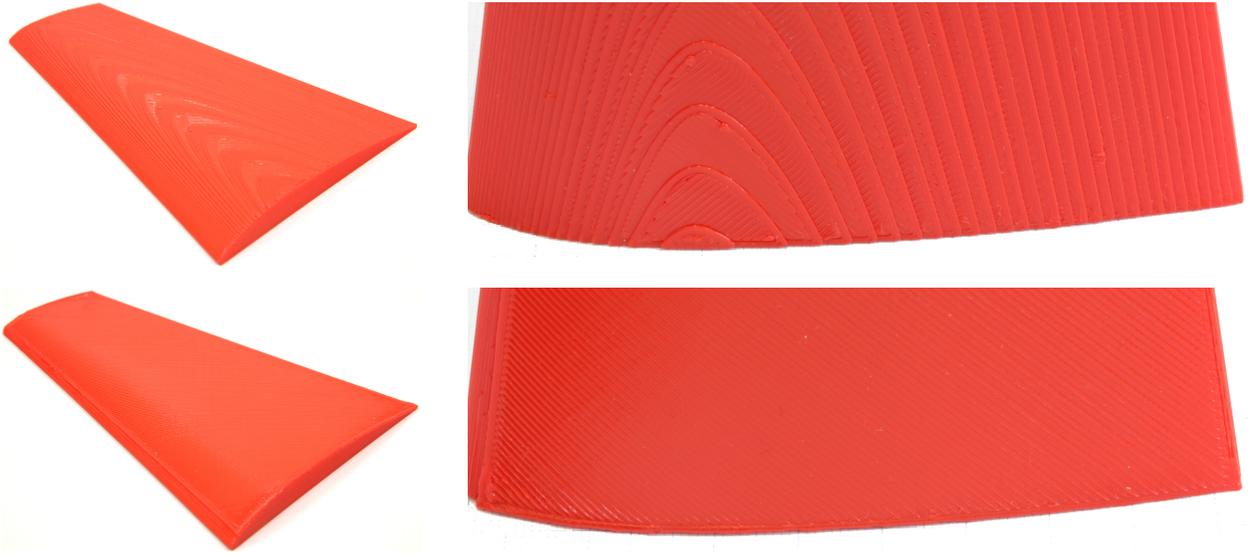


Fig. 9. Comparison of a wing from the NACA 4310 airfoil profile entirely printed with planar layers **top** and a nonplanar surface **bottom**. The surface quality is significantly improved while the printing time is even slightly lower for the second model.

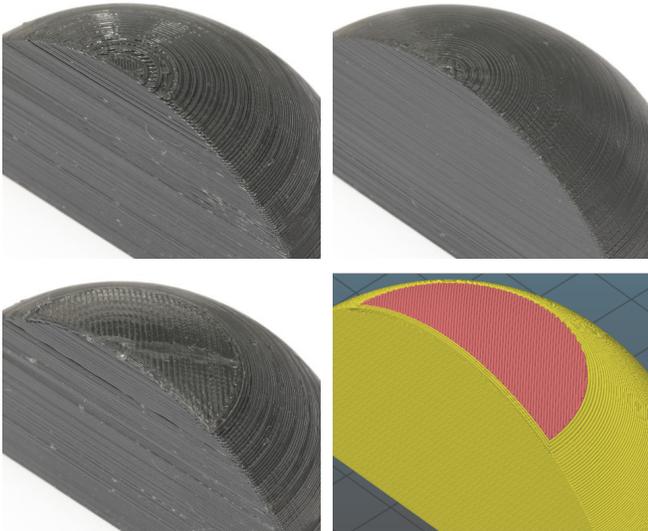


Fig. 10. A quarter-sphere to visualize the surface smoothing on its top and the ability to combine planar and nonplanar layers in a single shell surface. From top-left to bottom-right: planar slicing with fixed layer height, planar slicing with adaptive layer height, nonplanar slicing and the nonplanar toolpath.

VI. EVALUATION

To evaluate the practical use of our implementation, we printed a set of different objects, covering a variety of aspects. Fig. 9 shows a wing profile, with a high requirement for a smooth surface due to aerodynamic reasons. The surface smoothness is improved significantly although both objects are printed with the same layer height. Like in planar printing, the inner geometry of the underlying layers disappear within a few top nonplanar layers that are printed above. Furthermore, table II indicates that the printing time was actually reduced for the nonplanar object. The ability

to generate a complex surface structure with deformations in two dimensions was tested with the relief model shown in Fig. 2. It is also possible to use planar and nonplanar layers in a single object with automatic detection of regions suitable for curved layer printing. Fig. 10 shows three printed spheres with planar, adaptive and nonplanar layers and the nonplanar toolpath which is visualized in Slic3r.

TABLE I
RUNTIME OF THE SLICING ALGORITHM FOR DIFFERENT TEST OBJECTS

| | Relief | Wing | Quarter-sphere |
|-----------|---------|--------|----------------|
| planar | 7 sec | 2 sec | 2 sec |
| nonplanar | 213 sec | 18 sec | 16 sec |
| adaptive | 12 sec | 4 sec | 4 sec |
| facets | 32554 | 1152 | 3778 |

To compare the duration that the modified slicer needs to prepare the tool paths for a given model, all models were generated with planar, nonplanar and adaptive layers. The planar and nonplanar tool path is generated with a layer height of 0.3 mm while the adaptive toolpath is generated with layers between 0.1 and 0.3 mm. Table I shows the runtime and the number of facets for each printed model. The slicing duration for nonplanar toolpaths increases significantly for models with a high number of facets due to the costly checking of extrusion lines against all facets. However, the current implementation is not optimized and we expect the runtime to drop significantly with adequate data structures for fast spatial queries and parallelization. Nonplanar slicing will always be slower than purely planar algorithms since it adds additional steps to the toolpath generation, but the slicing time is typically negligible compared to the print time.

TABLE II
PRINTING TIME OF DIFFERENT EVALUATION OBJECTS

| | Relief | Wing | Quarter-sphere |
|-----------|---------|---------|----------------|
| planar | 123 min | 188 min | 91 min |
| nonplanar | 119 min | 182 min | 93 min |
| adaptive | | 218 min | 125 min |

Table II shows the print time of our evaluation objects on an Ultimaker 2 with a 0.4 mm Olssen Block nozzle. The printing speed of planar and nonplanar printed objects is similar, adaptive slicing results in a higher print time with a surface quality similar to nonplanar surfaces.

VII. CONCLUSION AND FUTURE WORK

We presented a novel approach for nonplanar slicing and 3D-toolpath generation for the FDM process. Our algorithm first automatically identifies those parts of the object that would benefit from nonplanar slicing; it then groups and filters the surfaces to prevent collisions of the printhead with previously printed structures. The resulting G-code can be executed on common three-axis FDM printers. Our algorithm is also integrated into the Slic3r GUI, providing a 3D-preview of the standard planar and the novel nonplanar toolpaths.

Objects printed with our slicing are closer to the original geometry and often also look better due to the smooth curved surface without stair-stepping artifacts. The mechanical properties of the objects should also be improved, as interlayer bonding should increase [21]. Our tests demonstrate that the printing time increases only slightly, in some cases even decreases, when using nonplanar layers, despite slightly longer (collision-avoiding) travel moves.

In future work, we plan to improve the performance of the toolpath generation using parallelization and better path planning for travel moves. A very interesting challenge concerns the handling of support structures, often needed to FDM-print objects with overhanging parts. These support structures are not part of the original object mesh, but are typically created on-demand after the slicing of the object has been done; resulting in a chicken and egg problem for the collision checking.

Our software has been released as open-source on Github:

<https://github.com/Zip-o-mat/Slic3r/tree/nonplanar>

REFERENCES

- [1] "Slic3r – open source 3D printing toolbox," 2018. [Online]. Available: <https://slic3r.org/>
- [2] A. Dolenc and I. Mäkelä, "Slicing procedures for layered manufacturing techniques," *Computer-Aided Design*, vol. 26, no. 2, pp. 119–126, 1994.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] J. T. Tyberg, "Local adaptive slicing for layered manufacturing," Ph.D. dissertation, Virginia Tech, 1998.
- [7] E. Sabourin, "Adaptive high-precision exterior, high-speed interior, layered manufacturing," Master's thesis, Virginia Polytechnic Institute and State University, 1996.
- [8] 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, pp. 2865–2881, 2002.
- [9] 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, pp. 323–331, 2003.
- [10] F. Wasserfall, N. Hendrich, and J. Zhang, "Adaptive slicing for the FDM process revisited," in *IEEE Conference on Automation Science and Engineering (CASE)*, 2017, pp. 49–54.
- [11] D. Ding, Z. Pan, D. Cuiuri, H. Li, N. Larkin, and S. Duin, "Multi-direction slicing of STL models for robotic wire-feed additive manufacturing," *Solid Freeform Fabrication Symposium*, pp. 1059–1069, 2015.
- [12] H.-m. Zhao, Y. He, J.-z. Fu, and J.-j. Qiu, "Inclined layer printing for fused deposition modeling without assisted supporting structure," *Robotics and Computer-Integrated Manufacturing*, vol. 51, pp. 1–13, 2018.
- [13] D. Chakraborty, B. Aneesh Reddy, and A. Roy Choudhury, "Extruder path generation for curved layer fused deposition modeling," *Computer Aided Design*, vol. 40, no. 2, pp. 235–243, Feb. 2008.
- [14] B. Huang, S. Singamneni, and O. Diegel, "Construction of a curved layer rapid prototyping system: integrating mechanical, electronic and software engineering," in *2008 15th Intl. Conference on Mechatronics and Machine Vision in Practice*. IEEE, 2008, pp. 599–603.
- [15] S. Singamneni, A. Roychoudhury, O. Diegel, and B. Huang, "Modeling and evaluation of curved layer fused deposition," *Journal of Materials Processing Technology*, vol. 212, no. 1, pp. 27–35, 2012.
- [16] B. Huang and S. Singamneni, "A mixed-layer approach combining both flat and curved layer slicing for fused deposition modelling," *Proc. of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 229, no. 12, pp. 2238–2249, 2014.
- [17] R. J. Allen and R. S. Trask, "An experimental demonstration of effective curved layer fused filament fabrication utilising a parallel deposition robot," *Additive Manufacturing*, vol. 8, pp. 78–87, 2015.
- [18] T. Llewellyn-Jones, R. Allen, and R. Trask, "Curved layer fused filament fabrication using automated toolpath generation," *3D Printing and Additive Manufacturing*, vol. 3, no. 4, pp. 236–243, 2016.
- [19] M. Micali and D. Dornfeld, "Fully three-dimensional toolpath generation for point-based additive manufacturing systems," *Solid Freeform Fabrication Symposium*, pp. 36–52, 2016.
- [20] Y. Jin, J. Du, Y. He, and G. Fu, "Modeling and process planning for curved layer fused deposition," *The International Journal of Advanced Manufacturing Technology*, vol. 91, no. 1-4, pp. 273–285, 2017.
- [21] J. B. Khurana, S. Dinda, and T. W. Simpson, "Active-z printing: A new approach to increasing 3D printed part strength," *Solid Freeform Fabrication Symposium*, pp. 1627–1644, 2017.
- [22] J. R. Kubalak, A. L. Wicks, and C. B. Williams, "Using multi-axis material extrusion to improve mechanical properties through surface reinforcement," *Virtual and Physical Prototyping*, vol. 13, no. 1, pp. 32–38, 2018.
- [23] Ultimaker B.V., "Ultimaker 2 user manual." 2017. [Online]. Available: ultimaker.com/download/18090/UserManual-UM2-v2.1.pdf