



Stylized Robotic Clay Sculpting

Zhao Ma^{a,*}, Simon Duenser^a, Christian Schumacher^b, Romana Rust^a, Moritz Bäcker^b, Fabio Gramazio^a, Matthias Kohler^a, Stelian Coros^a

^aETH Zurich, Switzerland

^bDisney Research Zurich, Switzerland

ARTICLE INFO

Article history:

Received July 5, 2021

Keywords: Robotics, Digital Fabrication, Sculpting, Clay, Path Planning

ABSTRACT

This paper presents an interactive design system that allows the user to create and fabricate stylized sculptures in water-based clay, using a standard 6-axis robot arm. This system facilitates the materialization of abstract design intentions into clay, through the algorithmic formulation of sculpting styles, the optimal path planning of the sculpting toolpaths, and a subtractive robotic fabrication process using customized tools. Unlike other precision-driven fabrication technologies, the authors embrace artistic uncertainty by conducting manual and robotic sculpting experiments and incorporating prominent parameters that affect the fabrication quality. The versatility of the described approach is demonstrated by designing a series of sculpting styles over a wide range of 3D models and robotically fabricating them in clay. Additionally, the paper explores various strategies for designing stylized robotic sculpting patterns by generating toolpaths informed by different techniques.

© 2021 Elsevier B.V. All rights reserved.

1. Introduction

Sculpture is one of the oldest forms of three-dimensional visual art, and clay is among the most widespread and frequently used sculptural media. Clay is malleable and can be formed into any imaginable shape, which makes it suitable for both additive and subtractive processes. During a sculpting process, artists utilize a variety of techniques and employ their hands or different tools to form a piece of clay until it satisfies the intent of their artistic expression.

In today's industrialized context, CNC milling is widely used for the manufacturing of three-dimensional sculptures, and often substitutes traditional manufacturing processes such as stone carving or foam cutting. However, conventional CNC milling techniques are limited when applied to soft materials like water-based clay. Highly ductile materials are notoriously difficult to cut mechanically, and the strong adhesive tendency

of clay greatly hinders the removal of small shavings. A special technique called “cryogenic machining” uses low-temperature coolant to freeze soft or elastic materials (for instance, rubber) temporarily during the milling process [1], but has to our knowledge not been used for clay material.

On the other hand, human sculpting still holds a special place due to its close association with arts and crafts. Due to the non-linear nature of the design process, artists usually rely on an interactive process to think and create through their minds and hands simultaneously. Compared to machining, manual clay sculpting satisfies this need with layers of sculpting strokes that are easily modified and superimposed on one another. Additionally, the often imperfect surface finish records the working process of the artist, and thus becomes a feature of artistic expression (Figure 1). Such patterns and textures are difficult to achieve, and are often not considered in CNC machining—and if they are required, they are typically achieved by subsequent special surface treatments.

With the long term goal of endowing robots with human-level skill, we present a user-guided design and motion plan-

*Corresponding author: Tel.: +41 77 980 63 49;
e-mail: ma@arch.ethz.ch (Zhao Ma)



Fig. 1. Various clay sculpting and modelling examples by human artists.

ning framework for robotic clay sculpting. By isolating those parameters related to the aesthetics of the sculpture from the fabrication process, we enable the user to define the style of the result. Our system automates the sculpting process by generating feasible motion trajectories that can be executed by robotic manipulators.

In order to make the computational problem tractable, we focus on a sculpting process that only involves material subtraction, using custom-shaped wire loop tools (Figure 3). The use of such customized tools poses two challenges: since the tools are not rotationally symmetric, applying conventional path planning algorithms for CNC machining, which treat the tool as axisymmetric, is not an option. Thus, new planning algorithms are needed to suit our tool, i.e. manage all 6 Degrees of Freedom (DoFs). The other challenge is how to support a wide range of possible sculpting strokes that unleash the expression of the user’s creativity while simultaneously complying with the aforementioned algorithms.

We open our investigation with a set of experiments aimed at identifying the primary parameters that affect the expression of the user’s design intent. Building on the insights gained through these experiments, we then address the challenges listed above by separating the entire pipeline into two independent units:

- **User-Guided Initialization** exposes a set of parameters for the user to control interactively, and transfers the input style information into a series of initial tool positions, i.e. a toolpath, that matches their design intent. This part aims to provide the user the freedom to design the sculpting strokes;
- **Path Planning** takes the initial toolpath as input to an optimization, and computes the complete robot trajectories. This part aims to find a high fidelity approximation of the input that balances accuracy and design expression. It further resolves any collisions, and respects all other workspace constraints of the fabrication robot.

We demonstrate the versatility of our computational approach using examples of increasing geometric complexity, and finally fabricate these objects with a *Universal Robot UR5* (5 kg-payload version), to assess the degree to which the simulated results translate to the real world. We then further explore and extend the style generation methods, and classify them into three categories with supporting analysis and discussion.

1.1. Overview

As illustrated in Figure 2, our pipeline proceeds as follows:

1. Given a mesh representation of the target model as input, the user sketches free strokes on the model to define preferred independent areas for further processing.
2. For each disconnected stroke group, the system computes a decomposed patch.
3. With a minimal set of user input as directional guides, our system generates the initial sculpting paths for each patch.
4. After the toolpaths have been initialized, the optimization adjusts them to eliminate collisions and smoothen sharp corners, while still maintaining the artistic expression.
5. Finally, once the toolpath has been successfully optimized, the user can preview the simulated result, or directly execute the trajectory information computed from the optimization to obtain a physical artefact of the “stylized” target geometry in clay.

This paper is an extended version of the conference paper that appeared in the proceeding of the ACM Symposium on Computational Fabrication 2020 [2]. The rest of the paper is structured as follows: Section 2 covers work related to our research. Section 3 shows a series of design experiments we conducted to understand how the material deforms with specific fabrication parameters. Section 4 explains how we built our design system based on the important parameters extracted from those design experiments. Section 5 presents the optimization formulation that helps to transform design intent into collision-free, feature-preserving robot trajectories. We demonstrate the capacity of our system on a set of physically fabricated examples in Section 6, and extend the methods in Section 7 for exploring and analysing various styles in addition to the original paper. We finally discuss the limitations and future work in Section 8.

2. Related Work

Clay Fabrication. As a representative material for geometry forming, clay sculpting has been widely used in the arts and crafts as a hand modelling process [3, 4]. Recently, its economical and malleable characteristics have also led to increasing popularity in 3D printing and multi-axis robotic applications. These applications typically employ a customized tool attached to a common 3-axis CNC machine [5] or a 6-axis industrial robot arm [6] and manufacture artefacts through additive, subtractive, or formative processes. Additive processes usually deposit clay either in layers to create sealed surface geometries, or in a woven style [7, 8] to create patterns. Deposition processes start either from a non-planar base geometry [9, 10] or a planar base that is gradually transformed to a non-horizontal fabrication plane [11, 12]. Taking advantage of the material’s malleability, digitally controlled throwing of the clay has also been studied in order to erect large-scale building structures at a distance [13]. Subtractive and formative processes, however, generally start with an initial block of clay which is shaped by either applying pressure to deform the material [14], or cutting material away [15]. Weichel et al. [16] combined additive and subtractive processes (i.e., milling) using two distinct tools. In contrast to these works, our method targets the fabrication of smaller-scale, detailed, high-curvature geometries that exhibit the characteristics of sculpted clay.

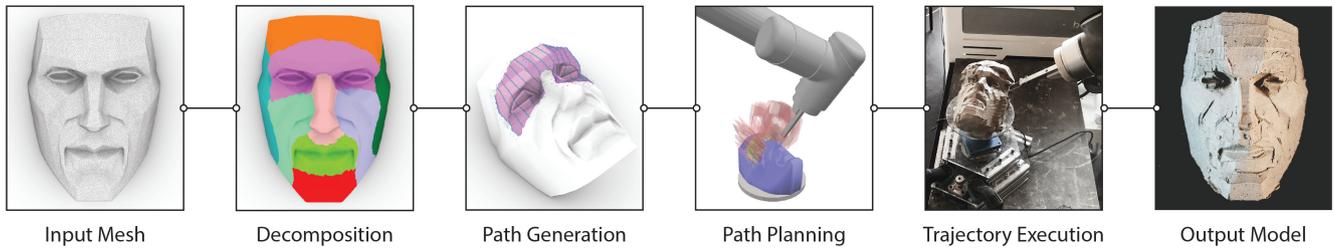


Fig. 2. System overview. It takes four steps to design & sculpt a given model with specific styles: 1) the system takes a general triangle mesh as input and decomposes it based on the drawn strokes. 2) The user specifies sculpting styles based on the patch-level parameters, and generates a set of initial toolpaths using our system. 3) Using the initialized toolpaths as input, the optimization computes robot trajectories while maintaining style information and simultaneously resolving collisions. 4) The trajectories are executed on a UR5 to sculpt a physical clay model that matches the optimized results.

Design Input & Interactive Robotic Processes. Besides additive and subtractive robotic processes for clay, there is a larger body of work that combines interactive design with virtual morphing or physical fabrication. These approaches often use a customized user interface to collect the design input and translate it into commands to modify the digital target model, or to physically actuate the robot.

Various researchers have developed interactive rapid-prototyping systems that provide instant feedback or guidance during the fabrication process [17, 18, 19, 20]. These systems emphasize user interaction, and embrace the imprecise mapping between the digital models and the physical results. Clifford et al. [21] employed a different approach, similar to Schwartz and Prasad [15], but using the hot-wire cutting technique on styrofoam to generate customized carving patterns. Similar pattern effects have also been achieved on wood [22, 23], in which a neural networks was applied to collect feature data from human operators using a gouge in order to replicate carving movements robotically. However, the research only explored single-movement features (such as torque and carving angles), and relies on precise predetermined toolpaths to carve out visually similar patterns on low-curvature surfaces. In contrast, our approach allows the user to focus on sculpting styles, while the system generates and optimizes the toolpaths behind the scenes, and actuates the robot to sculpt on a large variety of surfaces.

Another difference of our system lies in the decoupling of the design and fabrication process. Instead of requiring the user to participate in the entire design-to-fabrication sequence, and to incrementally modify the design based on the fabricated results, we fully automate the fabrication process. At the same time, we facilitate the design process through an interactive toolpath initialization, and compute feasible toolpaths that balance the user’s design inputs and model accuracy through an optimization process. In our context, the distribution of toolpaths vitally influences the appearance of the final surface. Kontovourkis and Tryfonos [24] and Rael and Fratello [25] demonstrated potential applications in this direction, but did not address the potential of subtractive robotic clay sculpting. This leads us to develop novel style-oriented toolpath generation techniques.

Path Planning. While toolpath planning plays an essential part in our work, we frame it in a broader context of path generation

problems where a large body of work is available in CNC machining [26, 27, 28, 29, 30, 31]. One main difference of our work lies in the customized tool, which requires specially designed path planning algorithms to explicitly take into account all of its 6 DoFs, where a normal milling bit or a dragging knife only requires the path planning algorithm to manage 5 DoFs. Dragomatz and Mann [32] surveyed general path generation methods used in CNC milling, and Elber and Cohen [33] summarized two main approaches, *isocurves* and *contours*, and their strengths and weaknesses. Our method is similar to the *isocurve* approach, but allows for more flexibility in toolpath generation as we emphasize design expression over machining time or toolpath length. To increase the variety of surface styles, we further employ a “divide-and-conquer” approach to toolpath generation by splitting the target geometry into several sub-sections. Similar approaches have also been applied to multi-axis milling path generation [34, 35].

The general problem of robotic path planning has been studied extensively in the past, and software packages are readily available. For example, the Open Motion Planning Library [36] provides a collection of sampling-based algorithms to plan a feasible path between two points, subject to optimality conditions. The *Descartes* package of the ROS-Industrial project [37] implements a tree search to find a robot trajectory that matches a suite of prescribed tool positions. Unlike these applications, however, toolpath planning for our application calls for long trajectories with dense sampling, in order to accurately follow the fine-scaled details of the target shape. Further, the entire length of the cut path is heavily restricted by collision constraints and computationally expensive optimization criteria. Such conditions are inherently challenging for sampling based approaches. Notably, De Maeyer et al. [38] report that already for 50 trajectory points, memory usage starts to become a matter of concern for the tree search used by *Descartes*. In our application we routinely exceed this number ten-fold. We therefore choose to rely on iterative optimization, namely Newton’s method, to handle the large number of parameters—albeit at the expense of global optimality.

Robotic manipulation of a wire-like tool has recently been studied in Duenser et al. [39], where an elastically deformable, heated rod cuts through blocks of polystyrene foam. That work focused on trajectory optimization for a small number of individual cuts using a comparably large tool, rather than on a global cutting strategy. In contrast, the tools we employ are

much smaller in size, such that a global strategy for path generation is necessary. Nevertheless, we draw inspiration from their work for our path planning step, and optimize for feasible robot trajectories in a similar fashion.

Style Transfer. There has been considerable research interests in style transfer for images using machine learning tools [40, 41, 42, 43] in recent years. Other stylization applications in graphics such as stylized rendering [44, 45], simulation of brush strokes [46, 47, 48], stylization of photographs and videos [49, 50] have also been investigated in different circumstances. While these works mainly focus on stylization in the 2D context, few have touched the realm of the three dimensional world, especially of fabrication.

These works mainly focus on stylization in the two-dimensional context, and few have looked at three-dimensional stylization, especially in combination with fabrication. While we draw inspiration from two-dimensional painting techniques, our unique application of stylized clay sculpting with a 6-axis robot arm requires novel techniques for stylization.

3. Design Factor Extraction

Instead of developing a fully automated system similar to existing software for Computer-Aided Manufacturing (CAM), we intend to provide the user with control over those aspects of the fabrication process that are relevant for the design and appearance of an object. Thus, we need to first understand what factors affect the fabricated result of a sculpting process so as to abstract them into parameters that can be built into our system.

In order to reveal the most important design parameters, we conducted a series of experiments involving the interaction between the tool and clay medium. These experiments were designed to help us in three aspects:

- Understand the relationship between the material deformation and sculpting velocity.
- Guide the selection of suitable shapes and sizes for the customized tools.
- Decide on a minimal set of parameters exposed to the user to exert control on the toolpath generation.

Before discussing the details of these experiments, we briefly introduce our tool designs.

3.1. Customized Loop Tool

A conventional *loop tool* (Figure 3) for cutting clay consists of a handle and a planar “loop”, a piece of steel wire or a thin, narrow metal strip bent into rectangular, triangular, or circular profiles to fulfill different cutting needs (size, angle, texture effects, level of detail, etc.). While the sculptor uses their hands for modelling in additive and formative processes, such loop tools are usually used for the subtractive process—cutting a strip of clay off by moving the tool along a desired path.

We use similar customized tools with replaceable “loops” (Figure 6) and a handle that can be attached to the robot. Compared to a conventional milling bit, one important benefit is the



Fig. 3. Left: different manual loop tools used by professional sculptors; Right: our customized loop tool that can be attached to a UR5 as the robot end effector.

non-axisymmetry of the tool, which allows it to cut off clay strips of different widths and sizes by simply rotating around its axis. While this additional flexibility is trivial for human users to control, it adds significant complexity to the planning algorithm—the additional degree of freedom needs to be managed and exploited.

3.2. Parameter Extraction Experiments

We categorize the experiments into two classes: patch-level parameters and path-level parameters. The patch-level parameters affect the selected areas (“patches”) of the mesh on which the preferred sculpting styles are applied; the path-level parameters affect the toolpaths generated on each patch. The selected patch can be either a portion of the whole mesh or the mesh itself.

Patch Geometry (patch-level). This parameter is directly related to how an input model is decomposed. It defines the area and shape to which a particular style can be applied, and can be created with various methods. We implemented a sketch-based method for the interactive design process in our system (Section 4). Note that for simple cases, manual decomposition with any mesh operation software may suffice.

Patch Overlap (patch-level). We observed undesired material aggregation near the seams between individual patches, leading to a clear visual separation. This is caused by the high ductility of the material—when the tool enters or exits the clay at the material interface, it carries forward some material by pushing or pulling, rather than causing a clean separation. This effect is most visible when entry and exit locations accumulate in the same spot. We found that we could reduce this effect sufficiently by introducing an overlap between adjacent patches, as illustrated in Figure 4, and thus eliminating the accumulation.

Toolpath Length (path-level). The above-mentioned material property has a similar impact on the toolpaths generated for a specific patch. Regardless of the generation method, a toolpath will start/end in three circumstances: 1) at the start/end point of another toolpath (toolpaths connected), 2) along the length of another toolpath (toolpaths overlapped), 3) at an empty area (toolpath disconnected).

Our experiments showed that for a specific patch, 1) and 2) will always create leftover material at the intersection, and 3)

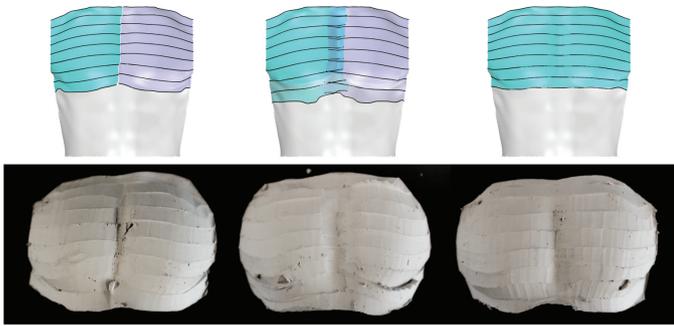


Fig. 4. Seam comparison models. Left: sculpt paths intersecting at seam area without overlapping; Middle: sculpt paths intersecting at seams with overlapping; Right: continuous sculpt paths across the whole surface.

will result in an area remaining unsculpted in the target geometry. As the number of intersection locations is largely decided by the number of toolpaths, we favor long toolpaths to reduce this aggregation. Two extreme cases are shown in Figure 5, where one contains randomly generated short toolpaths in various directions and the other contains only aligned toolpaths across the entire surface.

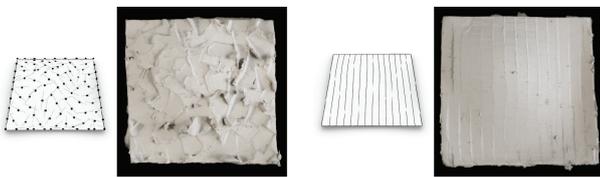


Fig. 5. Left: surface sculpted with 100 toolpaths of 18 mm length in random direction, generated from 100 randomly sampled points; Right: the same surface sculpted with 15 parallel toolpaths across the whole width of the patch.

Toolpath Direction (path-level). This parameter affects the toolpath generation process, and is the most important one for defining the artistic style the user wishes to achieve. It affects the visual effects of the sculpted stripe patterns on the final surface as well as the cutting depth into the clay. We use a Laplacian-based algorithm to generate evenly-distributed parallel-aligned toolpaths on top of each path, and the details are explained in Section 4.3.

Tool Direction (path-level). As shown in Figure 10, the three rotation parameters define the local pose of the tool. Our experiments confirmed that the *aligning direction* affects the precision of the target surface, and the *facing direction* affects the amount of material cut by each toolpath. These parameters together also affect the final surface quality (Section 4.4).

Secondary Parameters. Besides the parameters described above, we also experimented with several other parameters. These were found to be generally less effective in influencing the design and fabrication results compared those above, but still have an impact on the final sculpted appearance depending on the styles we choose (Section 7). For completeness, we list them below:

- *Density of toolpaths:* This parameter needs to ensure that the sculpted area covers the whole patch. Beyond that, increasing the density only increase optimization time with little gain for a selected tool. However, this parameter is still exposed to the user to compensate for any change in the tool.
- *Incline direction:* This parameter does not affect the result as much as the other two listed in the *Tool direction* categories, as long as it does not cause any collisions.
- *Tool shape:* As shown in Figure 6, we experimented with various tool shapes. However, we do not allow for tool changes during a sculpting task in general, so we exclude this parameter from the design stage. Note that the same optimization pipeline is applicable to different tool shapes.

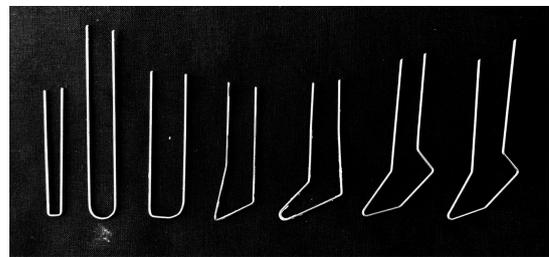


Fig. 6. Customized loop tool heads of different shapes.

3.3. Material Properties

As briefly mentioned in Section 3.2, we discovered that the plasticity and viscosity of the clay affect, on a local scale, how the clay behaves when the tool enters, sculpts, and exits it—which then directly affects the final appearance of the sculpted model. There are two main effects: 1) The clay demonstrates visible plastic behaviors when pushed by cross-section of the tool during the sculpting process. Although we use a thin 1 mm steel wire to reduce the cross-sectional area in order to reduce the plastic effect as much as possible, leftover clay is still noticeable along the moving paths of the tool. 2) Due to viscosity, the forces introduced by the tool cause a “pulling” effect when leaving the clay, and can even cause failure to detach when the remaining material is unable to withstand these forces. This effect mainly happens between the clay subtracted by the tool and the clay model, resulting in small accumulations on the target surface.

Complete modelling of the clay is extremely challenging, as its material properties change over time when the contained water evaporates gradually. However, for a thin (1 mm) tool made of steel wire, we found that by limiting the cutting speed to within 3 – 8 cm/s, we could reduce these visible defects to an acceptable level. We thus decided to conduct the fabrication under these settings and formulate the optimization using a purely geometric approach, resolving robot motion and collision issues without involving any simulation of the material behaviour.

3.4. Style as an Aesthetic Feature

One of the core contributions of our work is to deviate from a conventional path planning task by bringing the designer into the loop—to embed the user’s design expression as the sculpting styles in an automated robotic process. It provides a different perspective to robotic processes by adding manually-controlled elements into the “design-to-fabrication” process, and provides users with more freedom and control over their design expressions. Although not designed to be a computer-human interaction system, our system embeds design preferences and choices in a predefined manner.

While conventional CNC milling prefers precision, our system favours the possibility of creating various visual styles with a minimum amount of effort. It identifies a set of design parameters abstracted from fabrication experiments, and transfers the designed styles from the digital environment to physical artefacts with ease. As evaluating the aesthetics of a sculpture is difficult and inherently subjective, we leave it to the user to realize their creative intention by providing them with a considerable amount of freedom to explore this design space.



Fig. 7. Sculpting styles created with different decomposition schemes and toolpaths.

Figure 7 shows the sculpting style variations of a torso model created by different decomposition schemes or toolpaths. The visual styles formed by the sculpting toolpaths define a unique feature of the sculpting process. We believe these style variations provide new opportunities to explore new forms of robot control and to open discussions in human-robot interaction. More details are discussed in Section 4 and Section 5.

4. User-Driven Toolpath Generation

4.1. Design Parameters

One important goal of our research is to embed human design choices and expressions as “styles” into the automated robotic fabrication process. This requires the system to maintain a certain magnitude of precision, and at the same time deviate from the homogeneous look typical for the results of CNC milling. We rely on the key parameters selected based on the design experiments described in Section 3.2 to allow users to generate toolpaths creatively and transfer essential features into the fabrication process. Following the pipeline described in Section 1, we assume that we can describe the input mesh with a quadrilateral topology (for ease of geometry processing, we define four corners and four edges to the mesh regardless of its shape) and interpret the selected 5 parameters into variables that the user can access and modify in the GUI:

1. Free stroke locations drawn for model decomposition.
2. Offset distance at the overlapping area between patches.
3. Locations on patch boundaries as conceptual “corners”.
4. Distribution of start and end points of the toolpaths.
5. Number of toolpaths generated on each patch.

(1), (2) are *patch-level parameters* and relate to the *Decomposition* process; (3), (4), (5) are *path-level parameters* and relate to the *Toolpath Initialization* process. We will explain both of them in detail below.

For both CNC milling and our system, one necessary step of the toolpath generation is to develop toolpaths that can cover the whole surface of the input model. While common milling tasks use widely applied strategies including the *parallel*, *scallop*, *radial* and *flow-line* methods, we require a different procedure to generate toolpaths as the robot end effector (i.e. the customized loop tool) is not axisymmetrical, as normal milling bits are. The normal of the cutting plane must be aligned towards the cutting direction for an effective cut (though it doesn’t need to be aligned fully), so standard strategies would be insufficient.

Therefore, we developed a global-to-local strategy that decomposes the input model into small patches that can incorporate different sculpting intentions. Treating each patch individually, we generate toolpaths based on the isolines of a scalar field, which in turn is defined through user-provided boundary conditions for each patch. If no decomposition is given, the system will treat the whole mesh as a single patch, and conducts the toolpath generation over the whole area.

4.2. Decomposition



Fig. 8. Left: distance field calculated from the drawn strokes; Middle: decomposed patches without overlapping boundaries; Right: decomposed patches with overlapping boundaries of 15 additional triangle loops.

The *Decomposition* aims to allow the user to select different areas that can be treated separately for the toolpath generation. We developed a GUI to facilitate this task. The user can draw strokes on the model using a mouse, and the system will compute a distance field for each disconnected stroke. This field measures the distance between mesh vertices and the strokes, and later helps to compute separate surface patches using a priority queue based on the measured distances. Once the result is visualized, the user can accordingly decide to either draw additional isolated strokes to create more patches, or to intersect existing strokes with new stroke(s) to modify the shape of the corresponding patches (Figure 8).

Once the *patch-geometry* has been defined, the user can modify the overlapping areas around the borders where patches intersect. As the dimension of the input model may vary, this is achieved by adjusting the number of facets in the overlap areas (Figure 8). This adjustment aims to prevent the aggregation of

entry/exit locations of the loop tool, which will produce inferior surface quality due to the material behaviour discussed in Section 3.3.

4.3. Initialization

The *Initialization* process aims to provide intuitive toolpath generation for each decomposed patch. We treat each patch as a “quad-like” patch and ask the user to provide four “cutting” points near the boundary of each patch. These points are used to segment the closed boundary curve into four segments, i.e. two facing pairs. We assign the vertices of the two segments of one of the pairs with the value 0 and 1 respectively and assign that of the other pair with values interpolated from 0 to 1.

To generate the isolines, we use a technique similar to those described in Ma et al. [51] and Pereira et al. [52]. For each surface patch with n vertices, we compute a scalar field by solving the common Laplacian equation with boundary constraints:

$$\begin{cases} \mathbf{Lz}(x) = \mathbf{0}, & x \in \Omega \\ \mathbf{z}(x) = \mathbf{z}_0(x), & x \in \partial\Omega \end{cases} \quad (1)$$

where \mathbf{L} is the $n \times n$ discrete Laplacian and \mathbf{x} are the coordinates of the mesh vertices. Variables $\mathbf{z}(\mathbf{x})$ and $\mathbf{z}_0(\mathbf{x})$ are vectors of per-vertex values of all the vertices and boundary vertices, respectively. Those elements of \mathbf{z} that corresponds to the interior vertices are unknown, while the elements corresponding to the boundary vertices are given as constraints. We allow the user to modify the path direction and orientation by adjusting the position of the cutting points, the distribution of the assigned values, and the number of toolpaths (Figure 9).

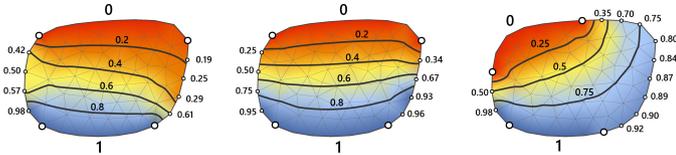


Fig. 9. Toolpath initialization: Left & Middle: same cutting point location, different distributions of assigned boundary values; Right: different cutting point location, distribution of assigned boundary value and density of paths.

We then interpolate a series of isolines from the scalar field. Users can set parameters interactively to find a path initialization that matches their vision. We found that an overlap of more than 30% of the tool width between adjacent paths is needed to allow for the optimization to modify the paths sufficiently in order to avoid collisions or match the target geometry more closely.

4.4. Tool Direction Modification

Although the *Decomposition* and *Initialization* processes succeed in transferring the design intention to initial toolpaths, we can further improve our initialization through local adjustments of the tool direction. While we can generally rely on the optimization to compute the locally optimal results, the experiment below demonstrates that better initialization leads to better surface quality (Figure 11), especially in high curvature areas where local minima may occur during the optimization.

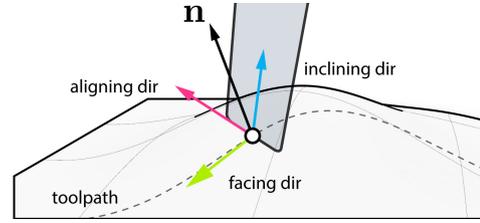


Fig. 10. During a sculpting movement, the tool pose is defined by three vectors: the facing direction, the aligning direction, and the incline direction.

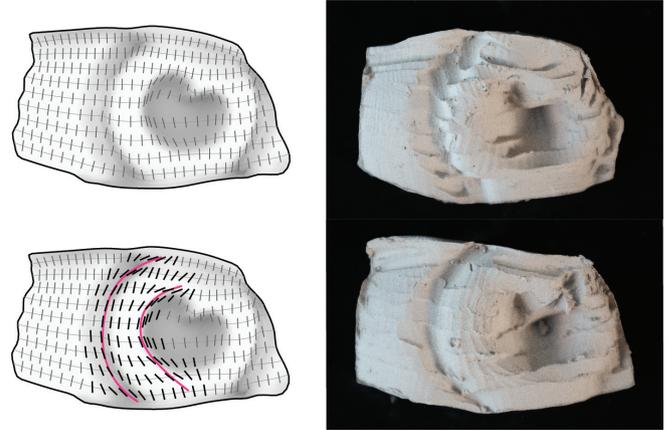


Fig. 11. Local adjustment of the facing direction using curvature information. The fabrication results illustrate noticeable improvements of the surface quality.

As the loop tool has 6 DoF, we define the 3 directions that are not constrained by a given toolpath (in fact, a series of tool positions) as *facing direction*, *aligning direction* and *incline direction* (Figure 10). A milling bit has no facing direction as it always cuts at the width of the tool’s diameter. For the loop tool, the cutting profile depends on the projection of the tool profile to the material along the toolpath direction and can be adjusted by its relative angle to the tangent direction of the toolpath.

For a sampled tool location along a toolpath, we initialize the *incline direction* using the normal direction of the patch, and project the tangent direction of the toolpath to the tangent plane of the patch at the referenced point to initialize the *facing direction*.

Additionally, we re-align some of the tool’s *facing directions* perpendicular to the averaged principal curvature [53] directions near high curvature areas:

$$\mathbf{n}_f = \frac{1}{N} \sum_{r < r_{near}} \mathbf{n}_p \quad (2)$$

where N is the number of samples of the principal curvature \mathbf{n}_p within a pre-defined sphere of radius r_{near} around the tool location. We illustrate the benefits of this post-processing step in Figure 11.

With the above procedures, we obtain a general initialization of both toolpaths and tool directions. However, there is no guarantee that these results can be executed with a specific robot without any collision or reachability problems. It would thus require the user to manually modify the paths iteratively

for a specific robot in use to resolve all collision issues, or use a simplified version or allow certain collisions (Figure 15 middle, Figure 16 upper right)—this is one of the main reasons that led us to develop the optimization process described in Section 5.

5. Optimal Path Planning

The toolpath generation in the previous sections defines a path that sweeps the target surface closely and expresses the aesthetic preferences of the user. It is, however, not guaranteed to be feasible, in the sense that it cannot be executed by a given robot without causing collisions or exceeding the robot’s reach. Therefore, given a patch of the target surface and the associated toolpaths (collectively referred to as the *input toolpath* in this section), we need to find a robot trajectory that 1) is feasible, 2) produces a cut surface that best approximates the target surface and 3) maintains the overall aesthetics of the cut surface implied by the input toolpath.

We follow an approach similar to the one proposed by Duenser et al. [39] for computing cut trajectories for an elastically deformable tool, manipulated by a two-armed robot. At the core of this approach lies the formulation of an optimization problem which matches the surface swept by the tool during movement (*toolsurface*) with the surface of the input model (*target shape*). In particular, we use similar formulations for the physical model of the system, the final *primary objective*, the *constraint objectives* and the last two of the *secondary objectives*, as introduced below.

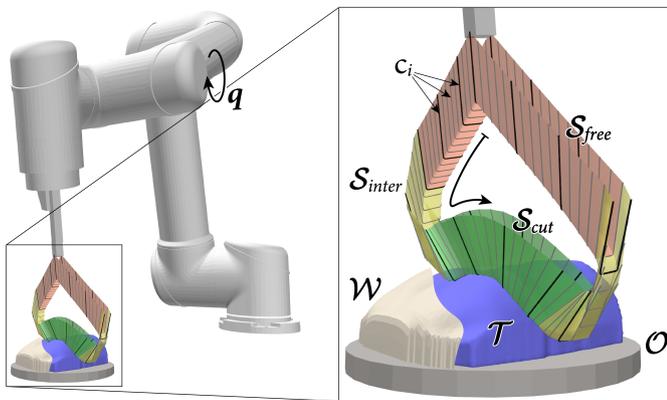


Fig. 12. An overview of the main components of the optimization model. The robot is shown in its rest pose, from where it traverses towards the workpiece (toolpath S_{free} and S_{inter}) and performs the cut (S_{cut}). The robot then moves back to its rest pose—although typically it would loop around and perform a number of successive cuts, optimized simultaneously, to carve out the entirety of a given surface patch.

Model Description. The robot trajectory is represented through a sequence of robot poses (*trajectory points*), each defined by the set of joint angles q_i , collectively forming the full trajectory $q = (q_i)$. In the case that a turntable is used, we simply view it as an additional robot joint and include its orientation in q , so that our optimization will treat the whole system as 7-DoF. The tool is rigidly attached to the robot end effector and modeled by its center line c_i , such that the path swept by

the tool forms the *toolsurface* S . Between the discrete steps of the trajectory we approximate this surface as piecewise linear. Using a kinematic model for the robot, the toolsurface is then fully defined through the joint angles as $S = S(q)$. For a full description of the setup, we further consider the target shape \mathcal{T} and its currently processed subsection \mathcal{T}^* , the current shape of the workpiece \mathcal{W} , as well as any other obstacles \mathcal{O} in the scene, such as the turntable. If the target mesh is split into several patches, the shape of the workpiece is updated after applying each of the corresponding cuts. See Figure 12 for an overview of the simulated setup.

Optimization Problem. Similarly to Duenser et al. [39], we formulate an unconstrained optimization problem of the form

$$\min_q E(q) = E_{prime} + E_{constr} + E_{sec}, \quad (3)$$

where all physical constraints are enforced through penalty terms, collectively denoted E_{constr} . The principal design objective E_{prime} defines a cost for the distance between the toolpath and its target, while E_{sec} collects several secondary objectives, as laid out in more detail below. We solve this minimization problem using Newton’s method with line search and a Levenberg-Marquardt type regularization.

The trajectory we optimize, and correspondingly the toolsurface, consists of several distinct, predefined subsections: One or more cut portions S_{cut} , in accordance with individual cuts of the input toolpath, which are designated to carve out the target shape. Transitional portions S_{free} , which describe the free movement in-between individual cuts, as well as from and to a fixed robot rest pose. And finally, intermediate portions S_{inter} , which are short connecting sections at the interface between S_{cut} and S_{free} . While the toolsurface of these sections may take part in cutting through the material, it is not optimized to match the target shape.

5.1. Primary Objective and Constraints

Surface Matching. The primary objective E_{prime} measures the closeness between the toolpath and the given target. We view this as a non-rigid surface registration problem and match the target surface \mathcal{T}^* with the toolsurface S_{cut} . Starting from a dense set of sample points on the target surface \mathcal{T}^* , we penalize the absolute distances to their respective closest points on the toolsurface S_{cut} . In principle, a simple quadratic penalty could be used for this. Although in a case where portions of \mathcal{T}^* can not feasibly be cut, this choice can lead to an undesirable overemphasis on these regions. Instead, we turn to a smooth step function of the form

$$H_\tau(d) = \begin{cases} 3\left(\frac{d}{\tau}\right)^2 - 2\left(\frac{d}{\tau}\right)^3 & 0 \leq d < \tau \\ 1 & d \geq \tau \end{cases} \quad (4)$$

This function acts similar to a quadratic penalty for a distance d close to zero, but smoothly transitions to a constant penalty over a transitional region of size τ (Figure 13, right). Thereby, regions that are definitely uncuttable, i.e. with a distance larger than τ , are simply ignored.

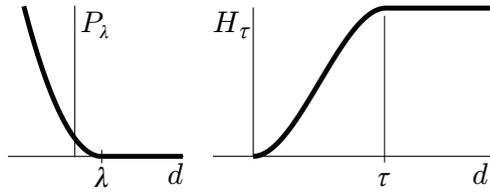


Fig. 13. Penalty functions on distance used for collision avoidance (left) and surface matching (right).

Initialization Procedure. Due to the relatively fine-scaled geometry of the toolsurface and its low rigidity, the outlined surface matching is prone to a large number of undesirable local minima. It therefore relies on a fairly good initialization, for which we use the input toolpath. To this end, we split the optimization process into two distinct stages. During the first, we do not apply the surface matching objective as the primary objective. Rather, we match the cut portion of the toolpath to the input toolpath, with regards to the position and orientation of the tool, using a quadratic penalty. Once a toolpath is found which resembles the input path as close as possible but has a feasible trajectory, we gradually drop this initial objective and apply surface matching instead. Using the input toolpath as initialization also establishes the desired global path layout, and in our experiments we found that this layout was generally well preserved during the surface matching stage, even once the initial objective had been removed entirely. At the same time, matching only the toolsurface provides a larger degree of freedom for the robot trajectory, allowing it to gracefully avoid collisions even in challenging situations.

Physical Limits. The constraints we consider are the robot's limitations on joint angles, as well as collisions of the robot and the tool. These collisions are namely: 1) self-collisions of the robot, 2) collisions between the robot and the workpiece \mathcal{W} and obstacles \mathcal{O} , 3) collisions between the toolsurface \mathcal{S} and the obstacles \mathcal{O} , 4) collisions between \mathcal{S}_{free} and the workpiece \mathcal{W} , and 5) penetration of \mathcal{S}_{cut} and \mathcal{S}_{inter} into the target shape \mathcal{T} .

For the implementation of robot collisions, the robot model is equipped with a number of spherical collision primitives, typically eight per link. From each collision primitive the signed distance is computed to all of the other collision spheres, as well as to the closest point on each of the objects in the scene. The latter are accurately represented through triangle meshes. A negative sign of the distance thereby signifies penetration. Similarly, proximity of the toolsurface \mathcal{S} is evaluated on a dense set of sample points on the surface, for each of which the smallest distance to the relevant objects is computed. We then penalize these distances with the one-sided quadratic function

$$P_\lambda(d) = \begin{cases} (d - \lambda)^2 & d < \lambda \\ 0 & d \geq \lambda, \end{cases} \quad (5)$$

where λ is a safety margin ($\lambda > 0$) or tolerance ($\lambda < 0$) (Figure 13, left). The same type of penalty is applied directly to the joint angles of the robot. The weighted sum of all penalties constitutes the full constraint objective E_{constr} , whereby the

weights are chosen to be large compared to any of the remaining objectives, such that the constraints are enforced rigidly.

5.2. Secondary Objectives

We identified several additional criteria for the quality and practicability of a toolpath, enforced through additional objectives E_{sec} .

Orthogonal tool orientation. For the fabrication process, it is favorable to keep the cutting direction orthogonal to the tool plane. While a cut can be produced when the tool plane is aligned with the cutting direction, this would produce only a narrow slit, often without fully removing a portion of clay from the workpiece. There is a high risk the clay will subsequently reattach, effectively undoing the cut. By only cutting orthogonal to the tool plane, long, narrow shavings are produced which can be removed immediately. Let \mathbf{c}_{ij} be the sample point j of the tool of time step i . For each \mathbf{c}_{ij} we penalize the deviation of the tool facing direction \mathbf{u}_i from the local cut direction \mathbf{v}_{ij} , for those sample points on the tool engaged in the cutting, as

$$E_{orth}^{ij} = l_{ij} \sin^4(\angle(\mathbf{u}_i, \mathbf{v}_{ij})) H_{a,\tau}^*(d_{\mathcal{W},ij}). \quad (6)$$

The symbol $\angle(\cdot, \cdot)$ is the angle spanned by two vectors. The cut direction is computed as $\mathbf{v}_{ij} = 1/2 (\hat{\mathbf{v}}_- + \hat{\mathbf{v}}_+)$, where $\mathbf{v}_- = \mathbf{c}_{i,j} - \mathbf{c}_{i-1,j}$, $\mathbf{v}_+ = \mathbf{c}_{i+1,j} - \mathbf{c}_{i,j}$, and $\hat{\cdot}$ represents a normalized vector. The associated step size $l_{ij} = 1/2 (\|\mathbf{v}_-\| + \|\mathbf{v}_+\|)$ is used to weight the objective. Finally, the last term of the equation represents a weight in the range $[0, 1]$ indicating whether the sample point is inside or close to the workpiece \mathcal{W} and therefore is relevant for the cut. Herein $H_{a,\tau}^*(d) = 1 - H_\tau(d - a)$ is an inverted smooth step function shifted by a tolerance a , and $d_{\mathcal{W}}$ is the signed distance between the sample point and \mathcal{W} .

Smooth discrete toolpath. To ensure smoothness of the discretized toolpath we penalize the angle spanned by the piecewise linear path of a tool sample point at each time step through

$$E_{smooth}^{ij} = l_{ij} \alpha_{ij}^2 H_{a,\tau}^*(d_{\mathcal{T},ij}), \quad (7)$$

where $\alpha_{ij} = \angle(\mathbf{v}_-, \mathbf{v}_+)$. This angle can essentially be viewed as the ratio between the local, approximated curvature of the toolpath (i.e. α_{ij}/l_{ij}) and the sampling density (given by $1/l_{ij}$). Thus, the objective does allow for an arbitrarily large curvature of the path, provided that the temporal resolution is adequate locally. As above, we weight the objective with the path length l_{ij} , and also according to the closeness $d_{\mathcal{T}}$ to the target shape, such that only portions of the cut are affected which may be visible in the final object.

Limited joint angle step size. While for the optimization we assume the toolpath is given by a linear interpolation of the tool geometry at discrete time steps, during fabrication the robot trajectory is interpolated linearly in joint angle space. For the k^{th} joint of the robot, a step of $\beta_{i,k} = \mathbf{q}_{i,k} - \mathbf{q}_{i-1,k}$ in joint angle space induces a maximum interpolation error of

$$\epsilon_{i,j,k} = r_{i,j,k} \left(1 - \cos\left(\frac{\beta_{i,k}}{2}\right)\right), \quad (8)$$

where $r_{i,j,k}$ is the distance between a tool sample point $\mathbf{c}_{i,j}$ and the k^{th} robot axes. For simplicity, we assume a rough, fixed estimate \tilde{r}_k for this distance for each joint angle, and penalize the corresponding approximation error through

$$E_{joint}^{i,k} = (\tilde{r}_k (1 - \cos(\frac{\beta_{i,k}}{2})))^2. \quad (9)$$

Limited tool step size. Collision avoidance of the tool surface is carried out with a fixed number of sample points. In order to maintain an adequate sampling density, it is necessary to limit the step size of the tool. Again, we apply a one-sided quadratic penalty

$$E_{step}^{i,j} = P_{-\delta}(-\|\mathbf{c}_{i,j} - \mathbf{c}_{i-1,j}\|) \quad (10)$$

to roughly ensure an upper bound of δ .

Quadratic regularization. Finally, we apply a weak quadratic regularization to the tool step size, such that all portions of the toolpath which are not governed by any of the above objectives remain short and smooth:

$$E_{reg}^{i,j} = \|\mathbf{c}_{i,j} - \mathbf{c}_{i-1,j}\|^2. \quad (11)$$

6. Results

To demonstrate the versatility of our system, we designed and fabricated four prototypes featuring different geometric characteristics. The decomposition of the input model by drawing strokes in the GUI and generating toolpaths for each patch takes around 0.5 h on average, depending on the number of decomposed patches and the number of attempts made to match the user's intention. The optimization takes 1 h to 4 h on average for the models we present here (torso, eye, face, 3D Möbius ring). The fabrication takes around 1 h on average with a joint velocity of 1 rad/s for the leading axis (the *movej* command [54]). After fabrication, the clay needs around one day to air-dry until its surface solidifies, and at least two days to be fully dried¹. Since the focus of our method is on sculpting and not on a complete ceramic workflow, we did not bake our model, though we do not see any barrier for doing so.

The optimization framework is implemented in C++, making use of the Eigen library [55] for matrix algebra. Searches for closest points on surfaces, as required for collision avoidance, are performed through an axis aligned bounding box tree, using the libigl library [56]. This operation accounts for the largest part of the computational costs in the procedure, with roughly 50%. Another 15%-20% of costs can be attributed the forward kinematics of the robot, and the respective first- and second order derivatives. For context, it should be noted that collisions between tool surface and target surface are tested on 140 sample points per trajectory point, and the distance function for surface matching is evaluated with similar density. Computation times for all examples are reported in Table 1, obtained on a standard PC with a 3.4GHz Intel Core i7-3770 CPU.

¹The clay we use for this paper is a typical fine-grain pottery clay.



Fig. 14. Left: our customized tool attached to the UR5; Right: the Arduino-controlled turntable.

6.1. Fabrication Setup

We designed a custom loop tool with a metal handle and a 3D printed ABS base. For the examples shown in this paper, we chose a rectangular profile with 10 mm cutting width and 30 mm cutting depth, made of 1 mm steel wire. The cutting depth is limited by the stiffness of the wire to avoid visible deformation while cutting. The loop is screwed onto the customized handle (10 mm × 10 mm cross-sectional area), which is attached to a UR5 through the ABS base.

For all the models fabricated in Section 6 and Section 7, we obtain the starting clay shape by sculpting over an initial clay block using parallel toolpaths on a geometry offset from the target model. This strategy is similar to the *parallel* roughing strategy used in conventional CNC milling, but does not need to be precise, since its purpose is to obtain a clay shape that is within the cutting depth limits (see aforementioned paragraph) to start the fine-scale sculpting process described in this paper. This roughing step can take advantage of the whole depth of the tool. The main sculpting process then conducts both the fine cut and the fabrication of the styles simultaneously.

We use a custom turntable controlled by an *Arduino Uno* to compensate for the limited reach of the robot. It rotates in both directions with 1.8° resolution, and acts as the 7th axis of our system to rotate the model to a position within the robot's reach. (Figure 14). The positions of the turntable during the sculpting process is obtained from our optimization, together with the other six joint angles of the robot arm.

6.2. Fabricated Models

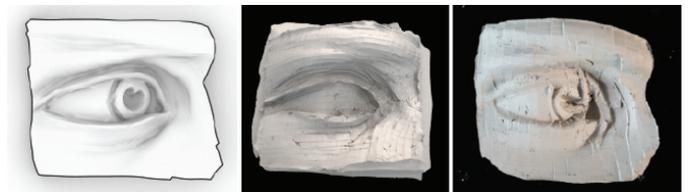


Fig. 15. The eye model. Left: Input geometry; Middle: model by executing CAD-modelled toolpaths; Right: model by executing trajectories generated from our system.

Table 1. Statistics of presented examples.

model	# patches	avg trajectory pts/patch	optimization time	fabrication time
Torso	5	334	1h 57m	7m
Face	6	445	4h 11m	26m
	7	473	4h 30m	31m
	9	412	5h 33m	33m
Eye	3	624	1h 21m	16m
Möbius	8	339	1h 39m	31m

Besides the torso model (Figure 7), the simplest of our examples is the eye model (Figure 15), which contains concave features that are nearly impossible to generate collision-free toolpaths for. We made several attempts through our CAD-modelling process, but fell back to use a smoothed version of the model as the collisions cannot be fully resolved. However, our optimization component resolves all the collisions and generates toolpath trajectories that achieve fabricated results with reasonable quality, even with a customized tool that is oversized for the details around the iris area.

We further use our interactive, user-guided design method to decompose and generate toolpaths for a face model that contains more challenging geometric features around the eye area (concave with large curvature) and the nose area (sharp edges). Similarly, CAD-modelled toolpaths failed to resolve collisions around the eye corner, but our system successfully fabricates the different styles we desire (Figure 16).

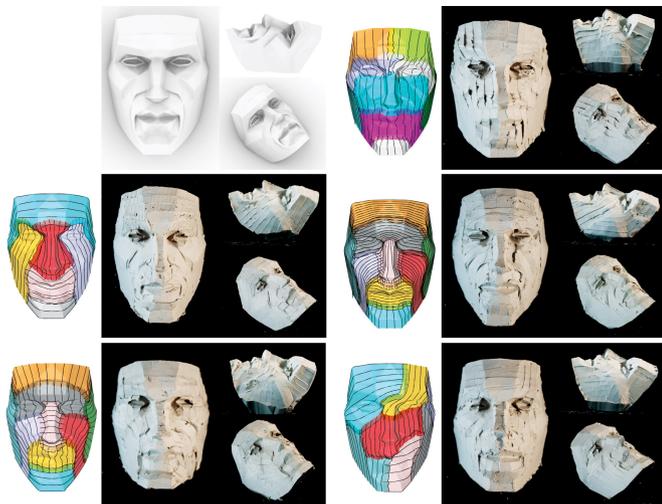


Fig. 16. Sculpting results of the face model. Top-left: input geometry; Rest: initialized toolpaths and the results with different styles by executing robot trajectories generated from our system.

Our system even allows the use of different parts of the tool for the sculpting process. In the 3D Möbius ring example (Figure 17-right), we use the *bottom blade* to sculpt the outer patches, and the *side blade* for the inner patches which are inaccessible to the *bottom blade* due to collision issues. However, we noticed two limitations: 1) Models with a thin connection to the base are likely to be deformed during the fabrication,

which causes lower precision. In this example, we compensate it by manually supporting the model. 2) Sculpting with the *side blade*, the maximum cut depth naturally cannot exceed the length of the tool. This can become a limiting factor when cutting the innermost portion of the ring, and constraints the possible size of the model.

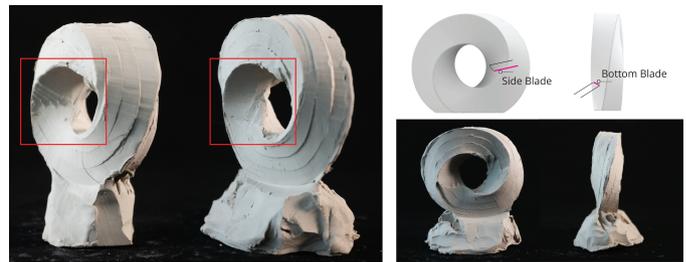


Fig. 17. Left: Reachability limitation from inadequate side blade length. Right: Illustration of model areas cut by side blade or bottom blade.

Preference for using a specific edge of the tool can be set by choosing the appropriate tool-local frame used for the initialization phase. The subsequent path optimization on the other hand is agnostic to the notion of distinct blades. That is, it treats the entire tool as one blade. Similarly to the overall path layout, we find that any preferences implied by the initialization are typically well preserved. As shown in Figure 17 (left), we verify the reachability limitation by fabricating two Möbius models with different thickness.

7. Style Exploration

The initialization process described in Section 4 demonstrates a generalized method to obtain a set of toolpaths to create a specific type of sculpting styles (visual patterns). While the decomposition improves the diversity of the styles over the whole sculpted surface, we see limitations in using only one strategy multiple times during a sculpting process. Additionally, the artistic expression of the sculpting gesture is not fully explored, if compared to the results sculpted by a human sculptor. In this section, we draw our inspiration from the brushstroke of various paintings and create a larger set of initialization techniques, enlarging the possibilities of our robotic sculpting techniques.

Brushstroke styles were often used to classify the genres of paintings [57] or attribute works of art to certain artists [58].

Using personalized brushstroke styles, famous painters created their specific signatures, which have also been analysed and reproduced with various modern techniques [59, 40, 41]. While it is not our aim to conduct a “style transfer” from specific paintings to a 3D sculpted object, the similarity between a brushstroke and a sculpting toolpath indeed inspires us to explore the possibility of expressive styles for our system.



Fig. 18. The two models used for style exploration demonstration: left – Height-field image for generating the “G” model; middle – the “G” model; right – the abstract face model.

For demonstrations shown in this section, we enlarge the tool sets for the sculpting process by adding a round-end loop tool and a half-width flat-end loop tool (the first two tools in Figure 6), expecting different comparisons of resolutions (large/small) and single stroke style (flat/round). For models, we used a relatively flat model generated from a height-field image of the letter “G” and an abstract 3D face model to compare the visual appearance of the developed styles on different types of surfaces (Figure 18, flat model – low curvature surface with geometric details, abstract face model – a volumetric model with various curvatures).

7.1. Style Classification & Toolpath Generation

While there exists a large collection of brushstroke techniques for paintings, including hatching, cross-hatching, brush ruling, flat wash, stippling, dry brush, etc., some are only reasonable for painting brushstrokes and color pigments. We selected paintings with three different types of representative brushstrokes based on the stroke length-to-width ratio (Figure 19), and developed multiple sub-styles for each type by varying the parameters tool width, tool shape, and toolpath length. For sculpting, we define the “length-to-width ratio” as the ratio of the toolpath length over the width of the tool. Instead of adding surface textures to sculpted objects as many human artists do (for instance, a “feather” texture for a sculpted bird), our aim in this paper is to create effective sculpting toolpaths, meaning the “styles” are created while conducting the sculpting process.

Stippling. The stippling technique uses short sculpting toolpaths (“scoops”) of length-to-width ratio around 1 to sculpt over the target area. Based on our experiments, the sculpting style created by toolpaths of this length-to-width ratio is mostly direction-independent to the alignment of the toolpaths—we don’t need to take care of the toolpath direction, and can even use toolpaths with randomized directions. Figure 20 shows two sculpted results with different alignments, using a 5 mm width flat-end tool and 8 mm length for all toolpaths. Thought the



Fig. 19. The referred three classes of brushstrokes for sculpting toolpaths: stippling (left, “Haymaking” by Camille Pissarro), short straight stroke (middle, “Starry Night” by Vincent Van Gogh), long continuous stroke (right, “The Scream” by Edvard Munch).

appearances of the two results look somewhat different, the difference caused by the alignments of the toolpath directions are almost negligible, as the alignments contribute no additional visual effects. The two examples both contain 411 toolpath segments, and the midpoints of the segments follow a Poisson disk distribution over the whole sculpted area.

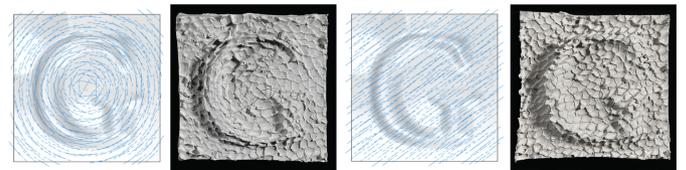


Fig. 20. Comparison model using different alignments of the toolpaths.

As named, this technique is inspired by the similar technique used in paintings (Figure 19-left) yet is also similar to the “dither” technique used in painting or pixel art, where dots of different densities are used to create pattern effects. The utilization of this technique in a sculpting process allows no space between the scoops, as the whole area needs to be covered by the sculpting paths.

Short Straight Stroke. The short straight stroke technique uses *directed* short sculpting toolpaths of length-to-width ratio about 1.5 – 3 to sculpt over the target area. Unlike the *stippling* technique, the direction of the toolpaths will impact the visual perception of the styles. Thus, we can create various sub-styles by using different strategies to generate the alignment map.

While artists use different sizes of paintbrushes to create different level of details in their paintings and different level of abstractions, sculptures can similarly present the potential of non-realistic expressiveness by creating different levels of abstraction of the model. We demonstrate the ability of our system by using sculpting tools of different width with the same alignment of the toolpaths on the same model.

To our surprise, curved toolpaths at this length-to-width ratio (1.5 – 3) have no visible effects on the result appearances with the current technique. Though the overall perception of the “G” is improved, this is mostly due to the exceeding number of toolpaths rather than the curvature of the toolpaths. In addition, the non-negligible number of turns in the toolpaths causes the tool to scrape on the surface with many unexpected small defects (Figure 21-D, top inner area) due to the material properties of clay described in Section 3.3. We thus limit the technique to

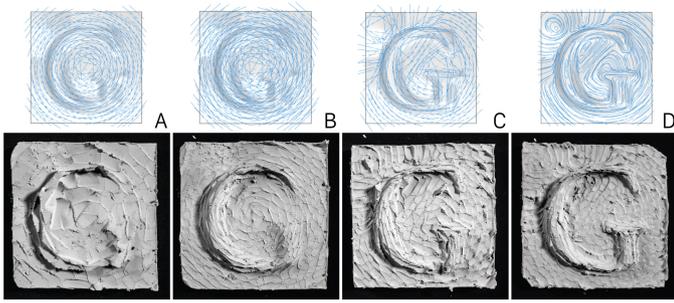


Fig. 21. Comparison model using different tool widths and alignment map parameters: A vs. B – 10 mm vs. 5 mm; B vs. C – radial vs. contour; C vs. D – straight vs. curved toolpaths.

only straight paths, as suggested already by the name of this technique.

Figure 21 demonstrates the sculpted results by varying parameters discussed above using the short straight stroke.

Long Continuous Stroke. The long continuous stroke technique uses continuous sculpting toolpaths of length-to-width ratio > 4 . We don't set an upper limit since toolpaths in this category are long enough to be treated as curves on the surface and the final appearances are mostly caused by how the curve flows along the surface. Similar to the *short straight stroke*, this technique also allows almost infinite possibilities to generate toolpaths. Notice that the strategy described in Section 4.3 also falls into this category.

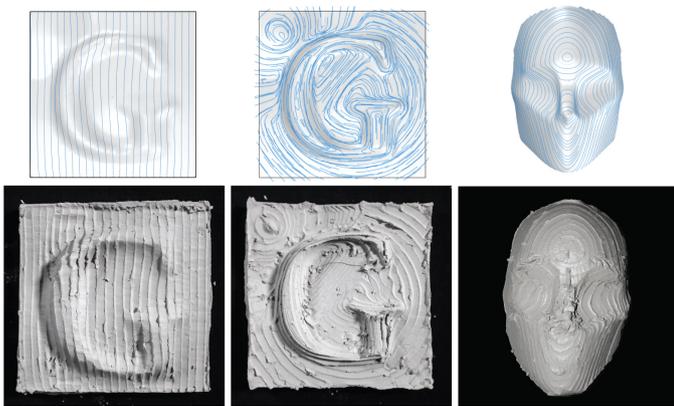


Fig. 22. Different models sculpted using the *Long continuous stroke* technique.

We show sculpted examples using the *long continuous stroke* technique with toolpaths generated with two different strategies (isoline and contour) in Figure 22, as well as the same strategy (contour) on different models. We leave the additional exploration to the readers for developing creative toolpath-generation strategies using this technique.

7.2. Style Analysis

We demonstrated the characteristics of three different sculpting style categories in Section 7.1 with sculpted examples, but how each style category should be used and what parameters will affect the final results still remains unanswered. Due to

the unquantifiable characteristic of artistic styles, we conduct a qualified analysis to explain the inner relations between the sculpting styles and the important parameters.

Table 2. Parameter sensitivity of the three style categories.

style	density	direction	tool profile
stippling	✗	✗	✗
short straight stroke	✓	✓	✓
long continuous stroke	✗	✓	✓

Parameter Sensitivity. Based on the examples we sculpted with different styles, we summarize the parameter sensitivity of each style in Table 2. Understandably, the *stippling* technique is the most simple and stable style generation strategy, as each sculpting “scoop” is similar to drawing a “point” on the paper. The resulting effects are predictable since the visual appearance largely depends on the accumulated effects of these “points”. As the number of the toolpaths are significantly higher than in the “isoline” strategy described in Section 4.3, the material effect described in Section 3.3 has more impact on the final appearance, and even overwhelms the intensity difference caused by the tool profile, which is explained in Figure 24.



Fig. 23. The abstract face model sculpted with different tools and in different toolpath densities: left – 250 toolpaths, round tool; middle – 400 toolpaths, round tool; right – 400 toolpaths, flat tool.

While the *direction* parameter has been well explained in Section 7.1, the *toolpath density* parameter is highly related to the feed-in/out position of the toolpaths, causing small material accumulation “dots” on the target surface. Since we are optimizing the number and position of the toolpaths for “best coverage” and “minimum overlap”, the overlapping of the toolpaths will cause the feed-in/out position of one toolpath to overlay on another. This effect has little impact on the *stippling* and *long continuous stroke* technique as the former is composed of only “dots” and the long toolpaths on the highest layer will always look long. However, the situation is different for the *short straight stroke* technique, as these feed-in/out “dots” will break the toolpaths they overlay into shorter segments with the length-to-width ratio falling into the *stippling* category—the alignment effects intended by the *short straight stroke* technique are thus sometimes visually disturbed.

In general, the *tool profile* parameter affects the results as the style intensity. This is mostly due to the shape of the *bottom blade* and the local curvature of the sculpted surface, as shown in Figure 24. In most cases, the round-end loop tool causes

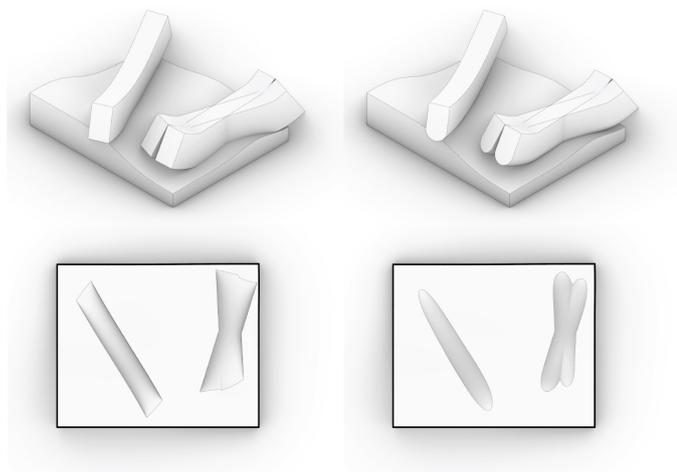


Fig. 24. Two examples of carved geometries by flat-end and round-end tools on an arbitrary doubly-curved surface. As seen in the figure, the round-end tool causes more depth variations in the results due to the profiles of the intersection between the toolpaths and the base geometry.

depth variation along each cut more than the flat-end tool, thus creating stronger and more dynamic visual effects, increasing the style intensity and expressiveness.

Expressiveness vs. Precision. As mentioned in Section 3.4, we set out to develop various sculpting styles as an artistic feature to our robotic sculpting system, providing users with a palette of sculpting styles emulating human sculptor patterns, rather than following the purely precision-driven approach of industrial CNC milling systems.

However, there are several factors affecting the expressiveness of the chosen styles. Treating a 100% matching model as the “most precise” result, we think the different levels of expressiveness come from the different levels of abstraction provided by the sculpting styles. In other words, the redundant material accumulation either created intentionally or caused by the sculpting process unintentionally affects how we perceive the sculpted results.

As shown in Figure 21 and Figure 23, larger tools result in lower resolution but higher abstraction, while smaller tools sculpt with better precision. Additionally, the flat-end tool allows for a better approximation of the target surface in convex regions, while the round tool results in more material leftovers. However, it is worth noticing that the abstraction should not diminish the reading of the base geometry, as in areas with geometric details (for instance, the nose area), stronger style intensity may have negative effects on the overall perception (the eye area of Figure 26-right can also be seen as one example). Additionally, as we’ve already discussed previously, the width variation of the toolpaths caused by the changing curvature of the base model and the profile of the round tool provides extra dynamic effects (Figure 26-middle, forehead), similar to the appearance produced by a human sculptor.

In general, it would be desirable to clarify the feature lines of the base model—for instance, the visual lines that define the shape of the nose—during the selection of the styles, while more choices are available for larger low curvature areas de-

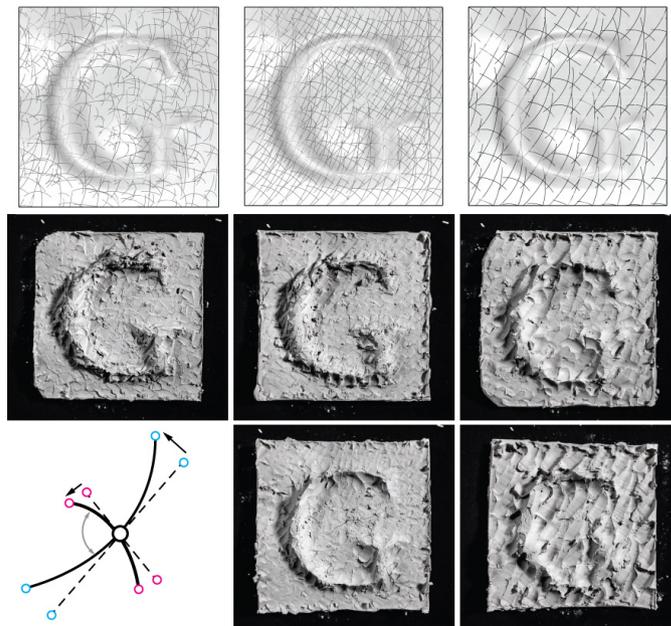


Fig. 25. Various tests using ineffective parameters: top-left – randomized cross-hatch units; middle – aligned cross-hatch units with different tool profiles; right – minimal coverage cross-hatch units with different tool-path depths; bottom-left – each “crosshatching” unit is composed of two 3-control-point NURBS curves, and the whole unit is parametrized with two parameters: the angle between the two curves, and the deviation of the end points perpendicular to the original straight curve in the tangential plane.

pending on the user’s preference of the expressiveness.

Weakly Effective Factors. During the development of the three categories of style, we discovered that certain factors that are effective for paintings or drawings do not work well in the sculpting context. We show the representative ones in Figure 25.

For these experiments, we use a parametrized “cross” stroke unit, inspired from the common “crosshatching” techniques from painting and sketching. We place multiple units over the target area on random positions or a grid (left v.s. middle & right), in different densities (middle v.s. right), with different tools (middle top–flat v.s. middle bottom–round), and in different depth (right top–same depth v.s. right bottom–long-stroke 1 mm deeper than the short one in each unit). Because of the overlapping characteristics of the cross-unit itself, these experiments demonstrate the following conclusions:

- For non-stippling styles, organizing the toolpaths towards a certain alignment is necessary to obtain the desired visual effects;
- For toolpaths falling into the *Short Straight Stroke* category, an excessive number of toolpaths impairs the desired visual effects (both direction and intensity);
- Toolpaths with larger depths may have stronger visual intensity, but it diminishes the visual perception of the other toolpaths.

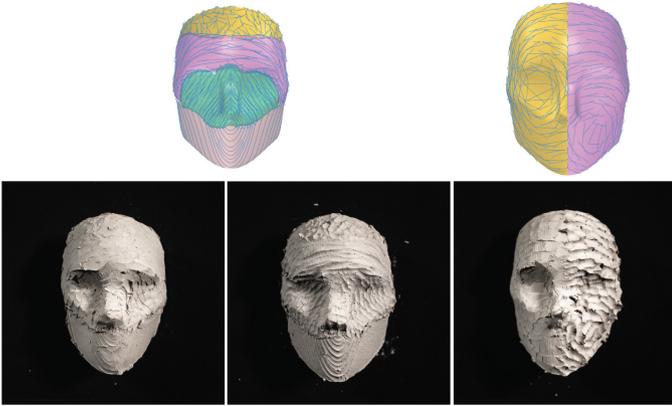


Fig. 26. Style composition based on different decompositions of the abstract face model: left–flat-end tool; middle–round-end tool; right– flat-end tool + round-end tool.

7.3. Style Composition

While we have demonstrated and analysed the three categories of sculpting styles, the power of the artistic expressiveness in our system comes from the style combination supported by the decomposition step (Section 4.2). We show three sculpted examples on the same abstract face model using different tools and decomposition pattern in Figure 26.

While the flat-end tool results in a more precise final appearance with subtle styles, the round tool provides us with a more expressive appearance, or even abstract facial expressions. The model sculpted with both tools (Figure 26-right) shows a strong artistic contrast using almost identical methods to generate the toolpaths (different center point of the radial alignments).

8. Conclusion

We have presented an interactive design and fabrication system that allows users to design different styles for sculpting clay models with a 6-axis robot. We identified and extracted a set of key parameters from a series of sculpting experiments and exposed them to the users in an interactive user interface we developed. The interface allows the user to decompose the input mesh into desired patches by drawing free sketch strokes and embed their design expressions as different sculpting styles individually by generating a set of corresponding initial sculpting toolpaths.

After the toolpaths have been initialized, our system conducts optimal path planning to resolve robot collision and reachability issues while still maintaining a maximum match to the given input surface. To increase the robot’s capabilities, we also added an Arduino-controlled turntable and integrated it into the optimization pipeline. We have demonstrated the capacity of our system through a set of fabricated clay models: torso, eye, face, and 3D Möbius ring. Moreover, as evidenced by the wide variety of styles for the same model, our system successfully enlarges the magnitude of expression incorporated during the design stage.

8.1. Limitation and Future Work

It is the combination of initialization and optimization that makes our system not only a robotic extension of the human hand, but a system that can intelligently fulfill certain design intentions. Yet we still have a long way to go to merge the system seamlessly into the human endeavours of design and creation. Many exciting questions are still left open for future work.

First, our system utilizes a subtractive strategy for the sculpting process assuming the clay to be rigid. This assumption works well when sculpting robust areas, but may cause imprecise results in areas with slender features due to the material deformation caused by the sculpting movement.² We plan to investigate methods that incorporate material simulation during the optimization for a better prediction, or that use external sensors to create a closed-loop system. This will contribute to controlling small accumulations caused by the material deformation, and the visual intensity of the styles.

Second, our current system can only predict the sculpted geometry after running the toolpath optimization. As the optimization process is computationally demanding, the current pipeline cannot present a predicted representation of the final appearance to the users instantly. We plan to investigate different methods that can approximate the final appearance independent of the optimization so as to enhance the design process with instant feedback.

Third, we explored various techniques for generating toolpaths to define a collection of sculpting styles. While these techniques employ parameters including tool profile, toolpath length and density, toolpath alignment, etc., techniques with an excessive number of toolpaths are not fully compatible with the optimization component. We plan to improve the optimization with better compatibility in the future.

Fourth, compared to the dexterity of a human hand that can apply additive, subtractive, and formative techniques during the sculpting process, our system only utilizes the subtractive process, with one type of tool. While combining both additive and subtractive techniques in the fabrication process is not difficult, predicting the material behaviour under formative processes (modelling, pushing) to fulfill the optimization tasks will require a simulation component.

Fifth, we developed the customized GUI for non-expert users to work on non-fired clay only. However, similar robotic processes have much larger application, such as foam wire cutting, wax cutting, or even fired clay, and the targeted user group may also extend to experts. Evaluations of the usability of our approach through user studies with experts and non-experts can provide more information about the effectiveness of our design abstraction, and highlight how we can extend the expressiveness of robotic sculpting. Ultimately, our path planning framework can also serve as a platform to fabricate with other materials and processes, through additional physical tests and tool designs.

²Professional sculptors often use inner skeletons (usually bent metal wires) to support such models, such as fingers in a hand model, or the nose tip of our face model. However, the fact that we do not use any skeletons also limits our selection of models.

Acknowledgement

We thank Dr. Espen Knoop for building the turntable and the customized metal pieces for the tool design, and Mr. Ryan Luke Johns for editing support with this manuscript.

References

- [1] Dhokia, VG, Newman, ST, Crabtree, P, Ansell, MP. A process control system for cryogenic CNC elastomer machining. *Robotics and Computer-Integrated Manufacturing* 2011;27(4):779–784. doi:10.1016/j.rcim.2011.02.006.
- [2] Ma, Z, Duenser, S, Schumacher, C, Rust, R, Bächer, M, Gramazio, F, et al. RobotSculptor: Artist-Directed Robotic Sculpting of Clay. In: *Symposium on Computational Fabrication. SCF '20*; New York, NY, USA: Association for Computing Machinery. ISBN 978-1-4503-8170-3; 2020, p. 1–12. doi:10.1145/3424630.3425415.
- [3] P. Faraut, C. Faraut, . *Mastering Portraiture: Advanced Analyses of the Face Sculpted in Clay*. PCF Studios, Incorporated; 2009. ISBN 978-0-9755065-6-1.
- [4] Faraut, P, Faraut, C. *Figure Sculpting: Planes & Construction Techniques in Clay*. v. 1; PCF Studios; 2013. ISBN 978-0-9755065-8-5.
- [5] Nan, IC, Patterson, C, Pedreschi, R. *Digital materialization: Additive and robotical manufacturing with clay and silicone*. 2016..
- [6] Bechthold, M. *Ceramic Prototypes – Design, Computation, and Digital Fabrication*. *Informes de la Construcción* 2016;68(544):167. doi:10.3989/ic.15.170.m15.
- [7] Friedman, J, Kim, H, Mesa, O. *Experiments in Additive Clay Depositions*. In: McGee, W, Ponce de Leon, M, editors. *Robotic Fabrication in Architecture, Art and Design 2014*. Cham: Springer International Publishing. ISBN 978-3-319-04663-1; 2014, p. 261–272. doi:10.1007/978-3-319-04663-1_18.
- [8] Rosenwasser, DM. *Clay Non-Wovens: Robotic Fabrication and Digital Ceramics*. In: *ACADIA 2017: DISCIPLINES & DISRUPTION [Proceedings of the 37th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA) ISBN 978-0-692-96506-1] Cambridge, MA 2-4 November, 2017*, Pp. 502- 511. CUMINCAD; 2017..
- [9] Ko, M, Shin, D, Ahn, H, Park, H. *InFormed Ceramics: Multi-axis Clay 3D Printing on Freeform Molds*. In: *Robotic Fabrication in Architecture, Art and Design 2018*. Cham: Springer International Publishing; 2019, p. 297–308. doi:10.1007/978-3-319-92294-2_23.
- [10] Dunn, K, O'Connor, DW, Niemelä, M, Ulacco, G. *Free Form Clay Deposition in Custom Generated Molds*. In: Reinhardt, D, Saunders, R, Burry, J, editors. *Robotic Fabrication in Architecture, Art and Design 2016*. Cham: Springer International Publishing. ISBN 978-3-319-26378-6; 2016, p. 316–325. doi:10.1007/978-3-319-26378-6_25.
- [11] Bhooshan, S, Ladinig, J, Van Mele, T, Block, P. *Function Representation for Robotic 3D Printed Concrete*. In: Willmann, J, Block, P, Hutter, M, Byrne, K, Schork, T, editors. *Robotic Fabrication in Architecture, Art and Design 2018*. Cham: Springer International Publishing. ISBN 978-3-319-92294-2; 2019, p. 98–109. doi:10.1007/978-3-319-92294-2_8.
- [12] Trilsbeck, M, Gardner, N, Fabbri, A, Haeusler, MH, Zavoleas, Y, Page, M. *Meeting in the middle: Hybrid clay three-dimensional fabrication processes for bio-reef structures*. *International Journal of Architectural Computing* 2019;17(2):148–165. doi:10.1177/1478077119849655.
- [13] Dörfler, K, Ernst, S, Piskorec, L, Willmann, J, Helm, V, Gramazio, F, et al. *Remote material deposition: exploration of reciprocal digital and material computational capacities. What's the Matter: Materiality and Materialism at the Age of Computation*, ed Maria Voyatzaki 2014;:361–77.
- [14] Tan, R, Stylianos, D. *Clay Robotics: Tool making and sculpting of clay with a six-axis robot*. In: *Living Systems and Micro-Utopias: Towards Continuous Designing, Proceedings of the 21st International Conference on Computer-Aided Architectural Design Research in Asia (CAADRIA 2016) / Melbourne 30 March–2 April 2016*, Pp. 579-588. CUMINCAD; 2016..
- [15] Schwartz, M, Prasad, J. *RoboSculpt*. In: Brell-Çokcan, S, Braumann, J, editors. *Rob — Arch 2012*. Springer Vienna. ISBN 978-3-7091-1465-0; 2013, p. 230–237.
- [16] Weichel, C, Hardy, J, Alexander, J, Gellersen, H. *ReForm*. *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology - UIST '15 2015*;:93–102doi:10.1145/2807442.2807451.
- [17] Zoran, A, Paradiso, J. *FreeD: A freehand digital sculpting tool*. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* 2013;:2613–2616doi:10.1145/2470654.2481361.
- [18] Johns, RL. *Augmented Materiality: Modelling with Material Indeterminacy*. In: *Fabricate 2014; dgo - digital original ed. Negotiating Design & Making*; UCL Press; 2017, p. 216–223. doi:10.2307/j.ctt1tp3c5w.30.
- [19] Peng, H, Briggs, J, Wang, CY, Guo, K, Kider, J, Mueller, S, et al. *RoMA: Interactive Fabrication with Augmented Reality and a Robotic 3D Printer*. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. CHI '18*; New York, NY, USA: Association for Computing Machinery. ISBN 978-1-4503-5620-6; 2018, p. 1–12. doi:10.1145/3173574.3174153.
- [20] Braumann, JaBC. *Adaptive Robot Control - New Parametric Workflows Directly from Design to KUKA Robots*. In: Martens, B, Wurzer, G, Grasl T, Lorenz, WE and Schaffranek, R (Eds.), *Real Time - Proceedings of the 33rd eCAADe Conference - Volume 2*, Vienna University of Technology, Vienna, Austria, 16-18 September 2015, Pp. 243-250. CUMINCAD; 2015..
- [21] Clifford, B, Ekmekjian, N, Little, P, Manto, A. *Variable Carving Volume Casting*. In: McGee, W, Ponce de Leon, M, editors. *Robotic Fabrication in Architecture, Art and Design 2014*. Cham: Springer International Publishing. ISBN 978-3-319-04663-1; 2014, p. 3–15. doi:10.1007/978-3-319-04663-1_1.
- [22] Brugnaro, G, Hanna, S. *Adaptive robotic training methods for subtractive manufacturing*. In: *Proceedings of the 37th annual conference of the association for computer aided design in architecture (ACADIA)*. Acadia Publishing Company; 2017, p. 164–169.
- [23] Brugnaro, G, Hanna, S. *Adaptive robotic carving*. In: *Robotic Fabrication in Architecture, Art and Design*. Springer; 2018, p. 336–348.
- [24] Kontovourkis, O, Tryfonos, G. *Integrating parametric design with robotic additive manufacturing for 3d clay printing: An experimental study*. *ISARC Proceedings of the International Symposium on Automation and Robotics in Construction* 2018;:918–925.
- [25] Rael, R, Fratello, VS. *Clay bodies: Crafting the future with 3d printing*. *Architectural Design* 2017;87(6):92–97. doi:10.1002/ad.2243.
- [26] Chiou, CJ, Lee, YS. *A machining potential field approach to tool path generation for multi-axis sculptured surface machining*. *Computer-Aided Design* 2002;34(5):357–371. doi:10.1016/S0010-4485(01)00102-6.
- [27] Feng, HY, Li, H. *Constant scallop-height tool path generation for three-axis sculptured surface machining*. *Computer-Aided Design* 2002;34(9):647–654. doi:10.1016/s0010-4485(01)00136-1.
- [28] Jun, CS, Cha, K, Lee, YS. *Optimizing tool orientations for 5-axis machining by configuration-space search method*. *Computer-Aided Design* 2003;35(6):549–566. doi:10.1016/S0010-4485(02)00077-5.
- [29] Sullivan, A, Erdim, H, Perry, RN, Frisken, SF. *High accuracy NC milling simulation using composite adaptively sampled distance fields*. *Computer-Aided Design* 2012;44(6):522–536. doi:10.1016/j.cad.2012.02.002.
- [30] Tournier, C, Duc, E. *Iso-scallop tool path generation in 5-axis milling*. *The International Journal of Advanced Manufacturing Technology* 2005;25(9):867–875. doi:10.1007/s00170-003-2054-7.
- [31] Zhu, W, Lee, YS. *Five-axis pencil-cut planning and virtual prototyping with 5-DOF haptic interface*. *Computer-Aided Design* 2004;36(13):1295–1307. doi:10.1016/j.cad.2004.01.013.
- [32] Dragomatz, D, Mann, S. *A classified bibliography of literature on NC milling path generation*. *Computer-Aided Design* 1997;29(3):239–247. doi:10.1016/S0010-4485(96)00060-7.
- [33] Elber, G, Cohen, E. *Toolpath generation for freeform surface models*. *Computer-Aided Design* 1994;26(6):490–496. doi:10.1016/0010-4485(94)90070-1.
- [34] Muntoni, A, Livesu, M, Scateni, R, Sheffer, A, Panozzo, D. *Axis-Aligned Height-Field Block Decomposition of 3D Shapes*. *ACM Trans Graph* 2018;37(5):169:1–169:15. doi:10.1145/3204458.
- [35] Zhao, H, Zhang, H, Xin, S, Deng, Y, Tu, C, Wang, W, et al. *DSCarver: Decompose-and-spiral-carve for Subtractive Manufacturing*. *ACM Trans Graph* 2018;37(4):137:1–137:14. doi:10.1145/3197517.3201338.
- [36] Şucan, IA, Moll, M, Kavraki, LE. *The Open Motion Planning Library*. *IEEE Robotics & Automation Magazine* 2012;19(4):72–82. doi:10.

- 1109/MRA.2012.2205651; <https://ompl.kavrakilab.org>.
- [37] Edwards, S, Lewis, C. Ros-industrial: applying the robot operating system (ros) to industrial applications. In: IEEE Int. Conference on Robotics and Automation, ECHORD Workshop. 2012.,
- [38] De Maeyer, J, Moyaers, B, Demeester, E. Cartesian path planning for arc welding robots: Evaluation of the descartes algorithm. In: 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). 2017, p. 1–8.
- [39] Duenser, S, Poranne, R, Thomaszewski, B, Coros, S. RoboCut: Hot-wire cutting with robot-controlled flexible rods. *ACM Trans Graph* 2020;39(4). URL: <https://doi.org/10.1145/3386569.3392465>. doi:10.1145/3386569.3392465.
- [40] Gatys, LA, Ecker, AS, Bethge, M. Image Style Transfer Using Convolutional Neural Networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016, p. 2414–2423.
- [41] Gatys, LA, Ecker, AS, Bethge, M, Hertzmann, A, Shechtman, E. Controlling Perceptual Factors in Neural Style Transfer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017, p. 3985–3993.
- [42] Gatys, LA, Ecker, AS, Bethge, M. Texture and art with deep neural networks. *Current Opinion in Neurobiology* 2017;46:178–186. URL: <http://www.sciencedirect.com/science/article/pii/S095943881730065X>.
- [43] Karras, T, Laine, S, Aila, T. A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2019.,
- [44] Xu, H, Chen, B. Stylized rendering of 3d scanned real world environments. In: Proceedings of the 3rd International Symposium on Non-Photorealistic Animation and Rendering. NPAR '04; New York, NY, USA: Association for Computing Machinery. ISBN 1581138873; 2004, p. 2534. URL: <https://doi.org/10.1145/987657.987662>. doi:10.1145/987657.987662.
- [45] Lake, A, Marshall, C, Harris, M, Blackstein, M. Stylized rendering techniques for scalable real-time 3d animation. In: Proceedings of the 1st international symposium on Non-photorealistic animation and rendering. 2000, p. 13–20.
- [46] Pham, B. Expressive brush strokes. *CVGIP: Graphical Models and Image Processing* 1991;53(1):1–6.
- [47] Healey, CG, Tateosian, L, Enns, JT, Remple, M. Perceptually based brush strokes for nonphotorealistic visualization. *ACM Transactions on Graphics (TOG)* 2004;23(1):64–96.
- [48] Hertzmann, A. Painterly rendering with curved brush strokes of multiple sizes. In: Proceedings of the 25th annual conference on Computer graphics and interactive techniques. 1998, p. 453–460.
- [49] DeCarlo, D, Santella, A. Stylization and abstraction of photographs. *ACM transactions on graphics (TOG)* 2002;21(3):769–776.
- [50] Kyprianidis, JE, Collomosse, J, Wang, T, Isenberg, T. State of the” art: A taxonomy of artistic stylization techniques for images and video. *IEEE transactions on visualization and computer graphics* 2012;19(5):866–885.
- [51] Ma, Z, Walzer, A, Schumacher, C, Rust, R, Gramazio, F, Kohler, M, et al. Designing Robotically-Constructed Metal Frame Structures. *Computer Graphics Forum* 2020;39(2):411–422. doi:10.1111/cgf.13940.
- [52] Pereira, T, Rusinkiewicz, S, Matusik, W. Computational Light Routing. *ACM Transactions on Graphics* 2014;33(3):1–13. doi:10.1145/2602140. [arXiv:1204.6216v2](https://arxiv.org/abs/1204.6216v2).
- [53] Meyer, M, Desbrun, M, Schröder, P, Barr, AH. Discrete differential-geometry operators for triangulated 2-manifolds. In: Visualization and mathematics III. Springer; 2003, p. 35–57.
- [54] Robots, U. The urscript programming language. *Universal Robots A/S, version* 2015;3.
- [55] Guennebaud, G, Jacob, B, et al. Eigen v3. <http://eigen.tuxfamily.org>; 2010.
- [56] Jacobson, A, Panozzo, D, et al. libigl: A simple C++ geometry processing library. 2018. [Http://libigl.github.io/libigl/](http://libigl.github.io/libigl/).
- [57] Lee, SG, Cha, EY. Style classification and visualization of art painting’s genre using self-organizing maps. *Human-centric Computing and Information Sciences* 2016;6(1):7. doi:10.1186/s13673-016-0063-4.
- [58] Sablatnig, R, Kammerer, P, Zolda, E. Hierarchical classification of paintings using face- and brush stroke models. vol. 1. 1998, p. 172–174 vol.1. doi:10.1109/ICPR.1998.711107.
- [59] Li, J, Yao, L, Hendriks, E, Wang, JZ. Rhythmic brushstrokes distinguish van gogh from his contemporaries: Findings via automated brush-stroke extraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2012;34(6):1159–1176. doi:10.1109/TPAMI.2011.203.