

Interactive Fabrication of CSG Models with Assisted Carving

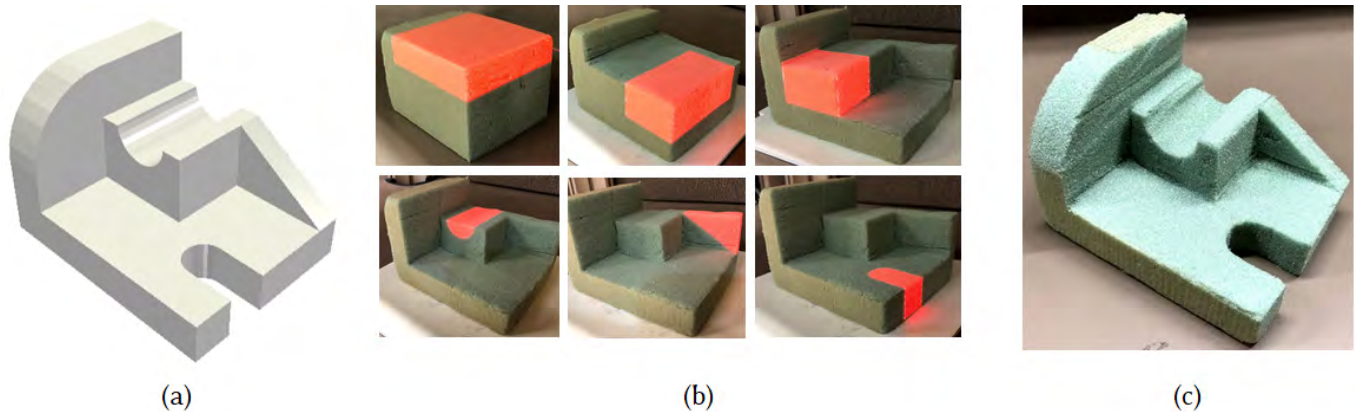


Figure 1. We assist users in carving a physical CAD model. (a) Given a target CAD model. (b) we generate a series of cutting instructions and use a projector to project them to a block of material to guide the user in carving them. (c) Until the user reaches the target physical replica.

ABSTRACT

We propose a method that helps an unskilled user to carve a physical replica of a 3D CAD model while only using manual cutting tools. The method starts by analyzing the input CAD model and generates a set of carving instructions. Then using a projector, we project the instructions sequentially one at a time to a block of material to guide the user in performing each of them. After each cutting step, we use the projector-camera setup to 3D scan the object after cutting. And automatically align the scanned point cloud to the CAD model, to prepare the position for the next instruction. We demonstrate a complete system to support this operation and show several examples manually carved while using the system.

ACM Classification Keywords

H.5.2. Information Interfaces and Presentation (e.g. HCI): User Interfaces; J.6. Computer Applications: Computer-Aided Engineering

Author Keywords

Personal Fabrication, Assisted Carving, Constructive Solid Modeling (CSG), Spatial Augmented Reality

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

TEI'19, March 171–20, 2019, Tempe, AZ, USA

© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM. ISBN 123-4567-24-567/08/06...\$15.00

DOI: http://dx.doi.org/10.475/123_4

INTRODUCTION

In the early stages of prototyping, users usually use digital and physical tools to create their design. They might start by creating a digital CAD model, then creating a physical mock-up prototype by hand using foams. This allows them to feel the shape of the model and to make adjustments, before going back to digital again. The process of transforming a digital CAD model to a physical mock-up is very important at this stage. Some users use 3D printers for that, but they cannot use a prototyping material like foam or wood. CNC milling machines, on the other hand, could work on foam or wood, but they are usually large and expensive or they require some process planning to produce the 3D model from all directions. For this reason, a large amount of fabrication at this stage is done using hand tools.

But the process of manually carving a digital model out of a block of material is still hard for novice users. Precise cutting is a painstaking job and figuring out the cutting steps in 3D is not trivial especially for complex models. Motivated by this problem, we present a method that can automatically analyze a CAD model and generates a series of manual cutting instructions. Then using a projector to project these instructions to guide the user in performing the cut until he reaches the target shape. The system has two main components; a method to generate manual cutting instruction, and a projector-camera calibrated setup to project those instructions, and to perform 3D scanning.

There are many 3D CAD representations, the most popular are boundary representations (B-Rep) and constructive solid geometry (CSG). In CSG, the CAD model is built using a tree of Boolean operations (like union, intersection, and subtraction) applied on simpler objects (like a sphere, cylinder, box, etc). In

this work, we want to generate a series of subtraction instructions that remove parts of a block of material. So, we decided to use CSG for its volumetric representation. Starting from a block of material, each cutting instruction removes some part of the material until we reach the target model. Then using a projector-camera system, we project each instruction to show the part of the material that should be removed allowing the user to perform that cut.

RELATED WORK

Personal Interactive Carving

A few previous papers proposed techniques to help novice users in carving, the closest to our work are the "Sculpting by Numbers" [10], "ShapeShift" [12], and Marnier's work [8] where they used a projector to help users in sculpting clay or carving foam. The main difference between these papers and our work is that they provide a general guidance for the user by projecting the depth or the goal object on the remaining material and let the user figure out how to reach that goal. Whereas our method leads the user one step at a time through their process. Being more prescriptive than the other approaches, our method is more suitable for novice users with little experience in carving or 3D. While the other papers might work better for more experienced users. In fact, one of the recommendations of the expert review in [8] is to develop "an algorithm to automatically generate a list of cuts needed to produce a specific 3D model, rather than playing back a previously recorded session would allow the system to completely replace the current process of producing stencils manually". Which is exactly what our paper is doing. We also allow the user to move the model freely by hand to perform the cut, and then we automatically align the model for the next cutting step.

Besides using a projector, in the "FreeD" project [13] they proposed a hybrid carving technique, where they developed a hand-held digital milling device that helps the user in carving the 3D model. Although the carving could be faster than our work, they normally produce rougher output, which is more suitable for free-form sculptures rather than CAD models.

Automatic Process Planning

The method described in this paper starts by analyzing the CAD model and generates a list of cutting instructions, then it leads the user one step at a time to perform these cutting operations. While this is suitable for beginner users, it's also very similar to the fabrication machines, where usually the CAD model is analyzed and a list of additive or subtractive instructions are generated and sent to the machine. In general, we call this process planning.

Additive machines like 3D printers, analyze the model by slicing it first and generates tool-paths from these slices. Milling machines, on the other hand, might require a more complex analysis, especially if the milling is to be performed from multiple directions, or in multiple stages using different machines. There has been a lot of work to automatically plan the fabrication process using different machines. Part of this work was focused on automatically discovering manufacturable features in the CAD model like holes, slots, and pockets, etc. See [5]

for a survey of this work. The planning process might require segmenting the model into different surfaces, especially if a hybrid additive/subtractive fabrication is used like in [3].

In our work, we propose a method to segment the negative volume (the remaining material) of the model into different parts, where each part is projected as a cutting instruction. The user can hold and transform the model in any direction using his hands to perform the cutting of the projected volume. A very similar approach to segmentation is proposed by [7] to segment the remaining stock in laminated manufacturing, with the goal to help the user in extracting the resulting object more easily, by optimizing the cuts performed in the remaining stock. The cuts are still performed on the stock automatically layer by layer by the machine, their method only have to segment the model into drawable parts that could be removed by the user in some direction, while in our paper we segment the remaining material into simpler parts that the user would be able to cut.

Spatially Augmented Reality

In this paper, we use a projector to project cutting instructions onto a block of material. This creates an augmented reality on top of a real physical object. An approach that is called spatially augmented reality. See for example [2]. In general, this approach uses the projector intrinsic and extrinsic parameters for the rendering cameras, such that the rendering is performed from the same position and orientation of the projector with respect to the environment. The projector itself is considered as an inverse camera. Many previous papers used this approach, especially for educational purposes. Some papers used this approach for guiding the user to perform certain tasks, such as sculpting [12, 10], or painting [4], or placing objects in a certain position. The benefit here is that the user does not need to wear any special devices to see the virtual reality.

Synchronizing the Digital and the Physical

In this paper, we propose a method to make a physical replica of a 3D CAD model while guiding the user using a sequence of cutting instructions. But, designers use a more fluid approach, moving from physical to digital and back to physical in different iterations. At any step of the carving process, the designer might decide to change something in the design, to perform this step physically or digitally. This work could be combined with other methods [6] to support such fluid process.

METHOD

System Design

In terms of hardware, the system is composed of a camera and a projector calibrated and fixed with regard to each other. The projector is used to project the cutting instructions on a block of material where it will illuminate the regions that should be removed by the user. The camera is used to capture the 3D shape of the object -together with the projector, see Figure 2.

In terms of software, the system is composed mainly of four components:

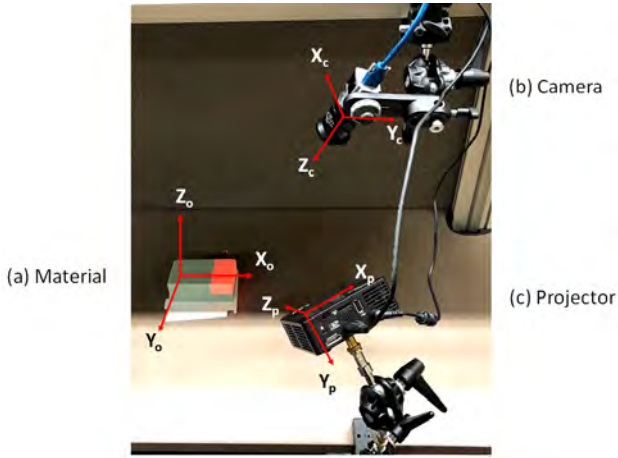


Figure 2. A projector-camera setup is used in this work, we notice that there are three coordinate systems; one for the camera, one for the projector and one for the 3D object.

Instructions Generator

Starting from the input CAD model, we need to generate the cutting instructions to be projected to guide the user. Section 3.2 is dedicated to this part.

3D Scanning

We need to capture the shape of the current material to find the correct alignment in order to project the next instruction. And to show the user how far he is from the target instruction. There are many 3D scanning techniques, but since we are using a projector to project the instructions, we can use it for 3D scanning as well. So, we decided to use structured-light gray-code 3D scanning, and we used the method of [9] to calibrate the projector-camera setup. The result of 3D scanning is a 3D point-cloud sampling of the object.

Notice in Figure 2 that we have three coordinate systems, one for the camera, one for the projector and one for the 3D model. We choose to use the projector coordinate system as the main coordinate system and to convert all points and transformations to this coordinate system. Because the projector is the common thing between the two operations; 3D scanning with the camera, and 3D rendering with the 3D model. The point cloud resulting from scanning is converted to projector coordinate system by a simple transformation using the rotation and translation matrices (These matrices are defined between the camera and the projector and we obtain them during the calibration process):

$$P_p = RP_c + T$$

While the 3D model is transformed to the projector coordinate system by the next 3D alignment step.

3D Alignment

Fully carving a CAD model requires the ability to cut the model from several different orientations to cover the model details from all directions. Also, the user might want to hold the model in his hands or move it freely in any direction to use

a specific tool. So, to allow a fluid process, we cannot restrict the model to a certain location or orientation.

For this reason, after each cutting step, we must perform a 3D alignment step between the scanned 3D point-cloud and the current instruction 3D model. We used the iterative closest point (ICP) method to perform the automatic alignment, see [1].

The iterative closest point (ICP) method requires a good initial alignment to succeed. Otherwise, it could easily get stuck on some local minima. For this reason, after each cutting step, we ask the user to place the object roughly in the same position and orientation of the current instruction model. We do that by projecting a 2D guidance contour on the table and ask the user to place the object roughly on that contour. We generate that contour by slicing the bottom of the current instruction model. This usually provides a good initial alignment for the ICP to succeed. Otherwise, we allow the user to fix the initial alignment manually.

3D Rendering

Now, after alignment of the scanned point cloud and the current instruction model. The model is ready for projecting the next instruction. So, we use WebGL to render the next instruction while using the projector intrinsic parameters for the rendering camera [11], and we show the rendering on the projector screen. At the same time, we show a rendering of the whole model with the current instruction highlighted on the computer screen to allow the user to move around it and explore it.

Generating Manual Cutting Instructions

The input to this step is a CAD model, and the goal is to generate a series of cutting instructions to cut that CAD model out of a block of material.

Starting from a block with the same dimensions as the bounding box of the CAD model. Each cutting instruction should remove a part of that block. And each cutting instruction should be simple enough to be done using hand tools.

Each cutting instruction removes a part of the block of material until we get the required shape. The union of all removed parts is equal to subtracting the CAD model from its the bounding box. So basically, to generate these cutting instructions, we turn the problem into a 3D segmentation problem of the negative space (The space resulting from subtracting the model from its bounding box), see Figure 3 for an explanation.

To perform this segmentation process, we need first to segment the initial input CAD model. The segmentation method will depend on the nature of the CAD model. In this paper, we focus on CSG CAD models due to their volumetric representation which makes the segmentation process much easier.

The algorithm works in a backward way, starting from the CAD model, we generate the last instruction to be performed by the user until we reach the first instruction, See Figure 4.

The algorithm performs the segmentation in two stages. In the first stage, it generates instructions to cut a union of bounding boxes of the simpler shapes in the CSG tree. In the second

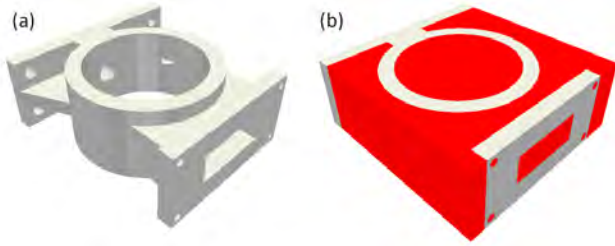


Figure 3. The negative space of a CAD model shown in red color. We get the negative space by subtracting the CAD model (a) from its bounding box.



Figure 4. Cutting instructions are generated in a backward way starting from the CAD model (b). While the actual cutting is done starting from a block of material (a).

stage, it generates instructions to cut the final CAD model out of this union of bounding boxes. See Figure 5 for a general overview of the algorithm.

In the first stage, we generate a cutting instruction for each simple geometry shape in the CSG tree (cube, cylinder, cone, etc). At first, for the negative shapes (shapes that are subtracted in the CSG tree), they are added directly to the list of instructions as the last cutting instructions. While for the positive shapes, if they are simple we add the negative space of the positive shape (its bounding box subtract the shape itself).

We do this to generate simple cutting instructions that the user can perform manually. The output of stage 1 is a shape that is a union of bounding boxes, see Figure 5.

In the second stage, we take the remaining volume to be cut, and we define a regular volumetric grid using each of the small bounding boxes that it contains. The goal of the second stage is to segment this remaining volume by merging the small cells in the volumetric grid to get the remaining regions/instructions to be cut. To do so we use a 3D voxel grid array, and we mark the filled voxels in this grid to be segmented. And then we run a region grown algorithm on this voxels grid starting from the top z-level.

Since the user usually starts cutting from the top, we need to look to the model from above and segment the remaining volume from the top according to each region depth in the z-level. See Figure 6.

The algorithm iterates over the z-levels starting from the top, and for each z-level, it will get the voxels in this level. And for each voxel, it computes the depth under this voxel (how many connected voxels under this voxel are marked as filled by the remaining volume). Then we select any voxel in this level and start region growing from that voxel (add the voxels that have the same depth to the same segment). After segmenting the voxels of certain z-level according to the depth, we start growing the segments down to the bottom levels (if they are

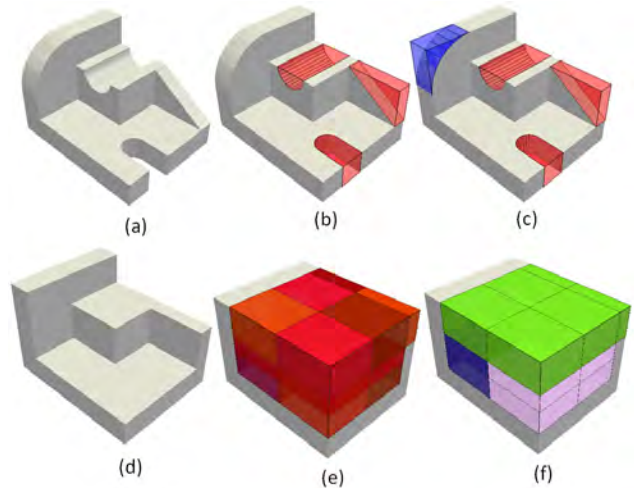


Figure 5. Method steps. (a) Given a CAD model. (b) We add negative objects instructions. (c) And we add simple positive objects instructions. This finishes stage 1 which results in a (d) union of bounding boxes object. (e) We construct a voxel grid of the remaining space. (f) And we segment the voxel grid by merging voxels.

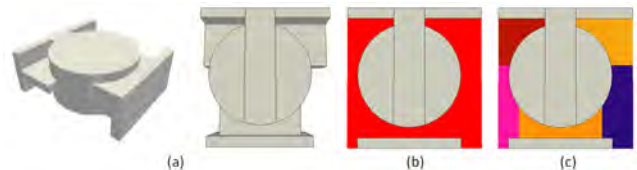


Figure 6. Segmenting the remaining volume by depth. (a) Bounding boxes object the result of stage 1. (b) Remaining volume shown in red color. (c) Segmenting the remaining volume by depth by counting how many connected voxels under each voxel are occupied by the remaining volume.

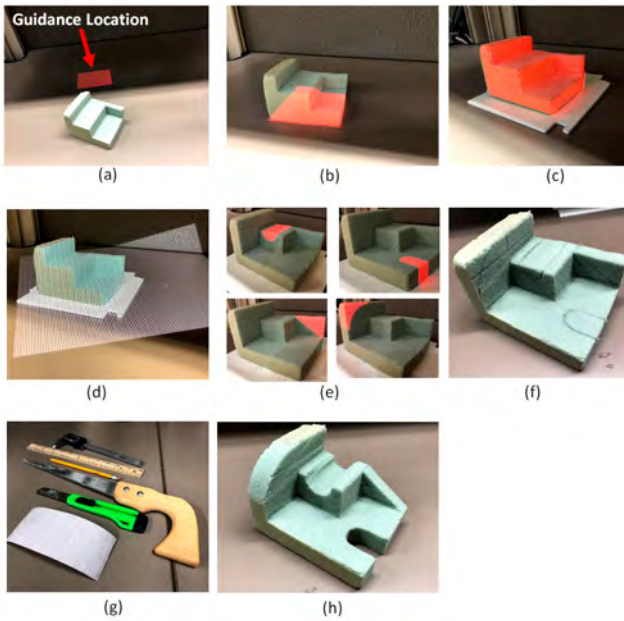


Figure 7. General process step. For a specific instruction, first, we find the best orientation. Then (a) We ask the user to place it in a guidance location. (b) The user roughly places the object in the guidance location (c) We project the whole object again for a better rough alignment. (d) We perform 3D Scanning. (e) We project the instruction(s). (f) The user marks the places to be carved. (g) The user performs the carving using hand tools. (h) result ready for next instruction.

marked as filled by the remaining volume). And we do so for the remaining voxels until no voxels are left not marked.

After segmenting the remaining volume into regions. We order them as cutting instructions for the user. The ordering is done using the uncovered surface area. So, we show the instruction with the largest uncovered surface area first. Since that the more surface area uncovered for a segment, the easier it would be for the user to cut it.

EXPERIMENTS AND DISCUSSION

Carving Process

The carving process starts from a block of material with the same dimensions as the bounding box of the CAD model, then we project a sequence of cutting instructions and ask the user to perform them, for each cutting instruction we need to perform the following steps (See Figure 7):

Finding the best orientation

We want to find the best orientation to show the instruction that gives the best projector visibility, such that we can show the instruction to the user. We simply want to choose from the six basic orientations; front, back, left, right, top and bottom. We do so by shooting rays starting from each polygon on the surface of the instruction in each of these directions. And choosing the direction that gives the best visibility (gives a larger visible area where these rays do not intersect the remaining volume). After choosing the best orientation we show it to the user and ask him to orient the part in the specified orientation.

Alignment

After finding the best orientation, we project a guidance location on the table (we slice the bottom of the model to get the guidance shape). We ask the user to place the object roughly in the specified location. This usually gives us a good initial alignment to be used for ICP algorithm. Then after the user places the object roughly in the specified location, we project the whole object to allow the user to enhance the rough alignment more. In case the alignment is still not good, we ask the user to do manual custom alignment.

3D Scanning

We start structure light scanning on the object, to capture its shape, the 3D scanning will give us a point cloud in the projector coordinates. Since we know the table plane equation (we get it from the calibration process where we placed a checkerboard on the table) we use the table plane to only keep points above the plane. This will remove all points on the table and thus keep only the points that belong to the object. We use the result point cloud and run the ICP algorithm to align it with the current instruction model. The resulting alignment will help us in projecting the next instruction correctly. Another possibility is to use the result of 3D scanning to verify the previous cut. We could calculate the distances between the point cloud and the current instruction model, and visualize the differences using colors as in [1,2]. This could help the user in fixing any errors introduced during the last cut operation.

3D Projection

We then use the projector to project the instruction, the user can choose to project multiple instructions (to cut them at once) as in the example in Figure 7.

Marking

Before performing the cut, the user could use a pencil or other tool to mark the cut lines. A process that is usually done in general before any manual cutting.

Cutting

The user is free to use any manual cutting tool in any orientation to cut the part. Then the user could start the next cut operation. And repeat the previous steps.

Instructions Examples

In Figure 8, we show several CAD models and the generated instructions by our algorithm. Some instructions steps are merged for illustration purposes.

Carving Experiments

In Figure 9, we show some examples of CAD models and the manual carving process, with the projected instructions on the top row and the actual result on the bottom row. To perform these experiments, we used a low-resolution Optoma projector (960 X 540 pixels) and a Basler camera (1928 X 1208). The projector and camera were fixed at a certain angle of the workspace table to allow projecting the instructions from several directions.

Using a method to clamp the object while in place could help in the experiments.

Input Model

Instructions

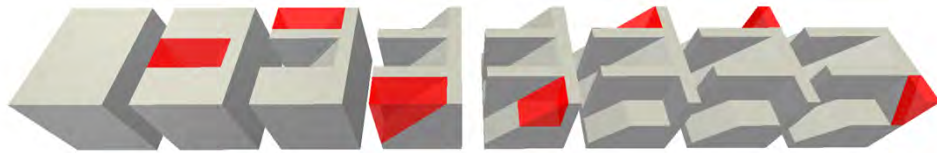
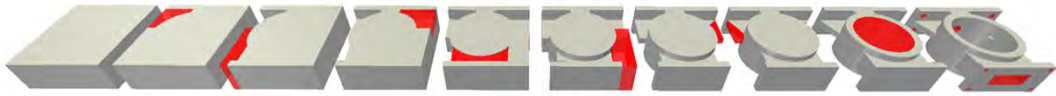
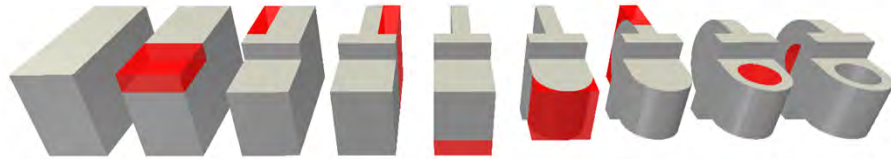
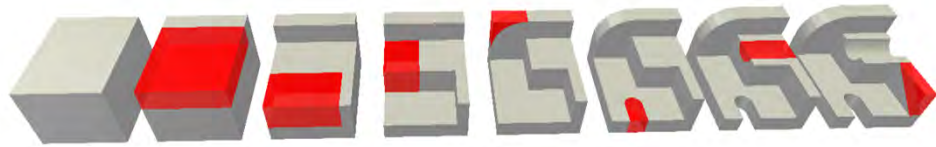


Figure 8. Examples of cutting instructions for CAD models. Some instructions steps are merged for illustration purposes.

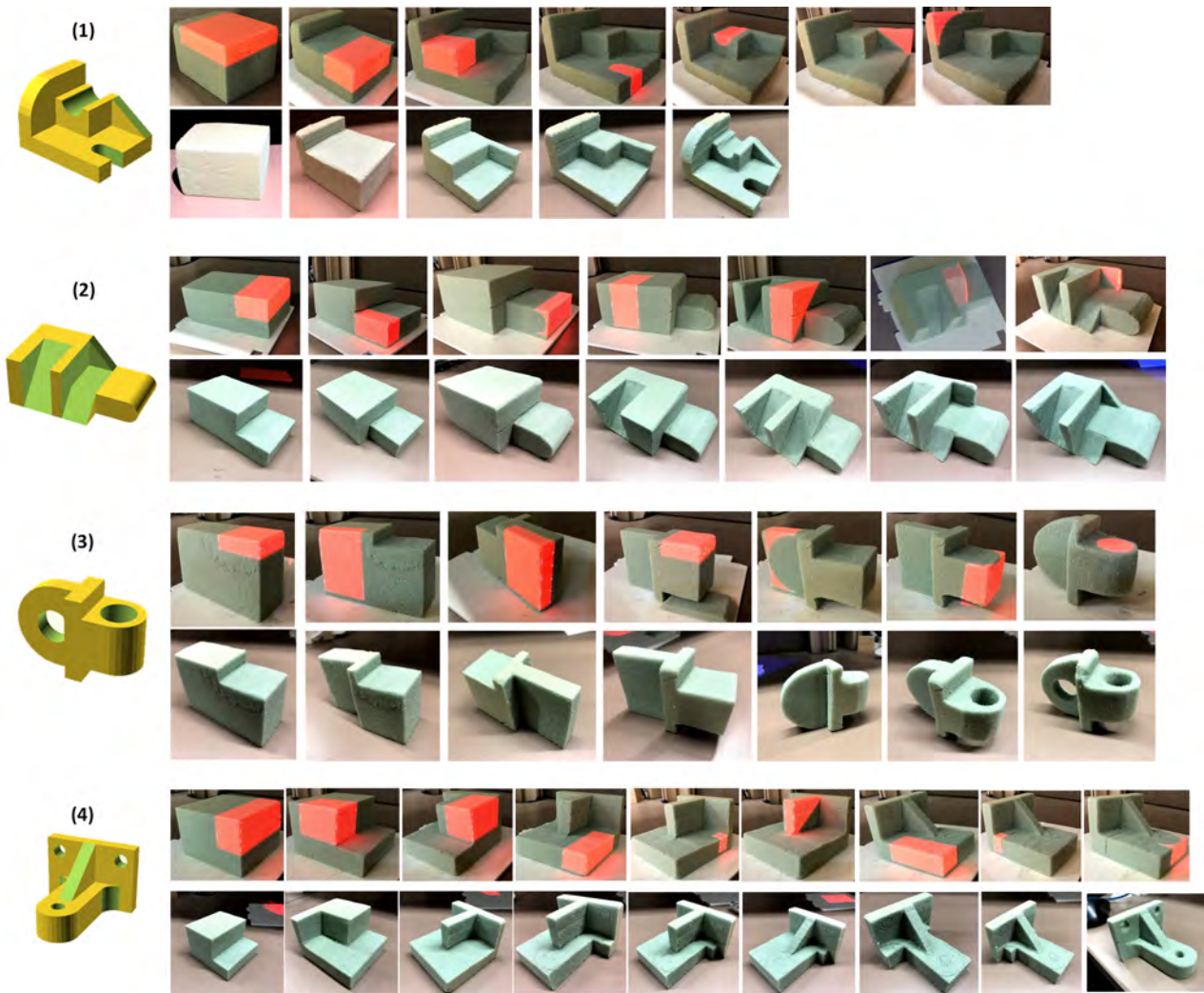


Figure 9. Some examples of CAD models and their manual carving process, with the projected instructions on the top row and the actual result on the bottom row.

Discussion and Future Work

Visibility constraints

There are some constraints on the type of CAD models that the method can handle. At first, we expect each part of the model surface to be visible from some direction, otherwise, the algorithm will generate instructions that cannot be performed.

One possible way to handle these cases is to split the CAD model to make its whole surface visible from some angle.

For each instruction, our method tries to find the best orientation from the six basic orientations that gives the best visibility -for the instruction to be illuminated by the projector and cut by the user. But in the future, we want to develop a more complex algorithm that selects the best orientation in general not only from the six basic orientations. That algorithm should select the best orientation while also taking into account that we want the model to stand by itself while projecting this orientation (in order to be able to project to it correctly).

We should also take into account the visibility constraints in the segmentation step of stage 2, because there are cases that one instruction is better divided into two instructions for enhanced visibility of the projector and the user.

One way to help with the visibility constraints is to have multiple projectors-cameras to cover the model from more directions.

Instructions order

Besides the visibility constraints, we should order the instructions such that we minimize the number of times where the user must move the object. This will help the user in performing multiple instructions at once.

User preference

Another limitation of the method is that it does not always match the user preferences. Sometimes there are multiple possible good segmentations, and in general, the user might have his own preference, and he might not like the instructions given. Some other times the instructions given are too long for the user.

To help with that, we give the user the option to combine multiple instructions, to skip instructions, or to change the order of instructions.

In the future, we want to allow the user to do his own special cuts, and then rerun the algorithm starting from the user operation.

Removing or adding material

In the experiments, we noticed that it is easy for the user to cut more material than required. Since we are dealing with prototyping foam, it is not easy to fix that. So, it is always better if we are conservative in leaving more material. We could handle that by expanding the cutting instruction by adding some margins; for example, 1-2 mm to each cutting instruction from all sides. At the same time, we show the actual cutting instruction to allow the user to start from the expanded instruction until he correctly reaches the target.

Projecting inner structure

Sometimes the cutting instruction is not a simple straight cut, or it does not go all the way through the material. In these cases, it might be better if we project part of the inner structure to guide the user in performing the instruction.

Full system

In the future, we want to develop a fully integrated system, that allows the user to perform any operation digitally or physically and use the 3D scanning and the projector to transfer the operation to the second model. Allowing a fluid synchronization between the digital and physical worlds. And we could also allow the user to select a specific point or region in the screen and let system highlight it physically using the projector. On the other hand, we can also allow the user to physically draw or mark something on the model and we can capture these marks and transfer them to the digital model.

CONCLUSION

In this preliminary work, we demonstrated a system that helps unskilled users in carving CAD models. We noticed that carving is a hard process for novice users, as they need help figuring out the steps to cut a specific 3D model and to measure and mark the locations to perform these cuts. We presented a method to generate carving instructions for an input CAD model and used a projector to project these instructions to a block of material. To generate the instructions, we need to solve a segmentation problem of the negative space of the CAD model. The method must take into account the visibility constraints and some other factors. 3D alignment is an important step in the system that prepares the system for projecting the next instruction. The system flexibility helps the user to match his preference in merging instructions, changing their order or defining custom instructions. The presented system has the potential to be further developed to allow for a fluid synchronization between the digital and physical worlds especially at the prototyping stage.

REFERENCES

1. Paul J Besl and Neil D McKay. 1992. Method for registration of 3-D shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, Vol. 1611. International Society for Optics and Photonics, 586–607.
2. Oliver Bimber and Ramesh Raskar. 2005. *Spatial augmented reality: merging real and virtual worlds*. CRC press.
3. YC Chang, JM Pinilla, JH Kao, J Dong, K Ramaswami, and FB Prinz. 1999. Automated layer decomposition for additive/subtractive solid freeform fabrication. In *Proceedings of the Solid Freeform Fabrication Symposium, The University of Texas at Austin*. 111–120.
4. Matthew Flagg and James M Rehg. 2006. Projector-guided painting. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*. ACM, 235–244.
5. JungHyun Han, Mike Pratt, and William C Regli. 2000. Manufacturing feature recognition from solid models: a

status report. *IEEE Transactions on Robotics and Automation* 16, 6 (2000), 782–796.

6. Ammar Hattab, Ian Gonsler, Daniel Moreno, and Gabriel Taubin. 2017. Differential 3D Scanning. *IEEE computer graphics and applications* 38, 3 (2017), 43–51.
7. KP Karunakaran, Shivamurthy Dibbi, P Vivekananda Shanmuganathan, D Satyanarayana Raju, and Srinivasarao Kakaraparti. 2002. Efficient stock cutting for laminated manufacturing. *Computer-Aided Design* 34, 4 (2002), 281–298.
8. Michael R Marner and Bruce H Thomas. 2010. Augmented foam sculpting for capturing 3D models. In *3D User Interfaces (3DUI), 2010 IEEE Symposium on*. IEEE, 63–70.
9. Daniel Moreno and Gabriel Taubin. 2012. Simple, accurate, and robust projector-camera calibration. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*. IEEE, 464–471.
10. Alec Rivers, Andrew Adams, and Frédo Durand. 2012. Sculpting by numbers. *ACM Transactions on Graphics (TOG)* 31, 6 (2012), 157.
11. Kyle Simek. 2013. Calibrated Cameras in OpenGL. http://ksimek.github.io/2013/06/03/calibrated_cameras_in_opengl. (2013). Accessed: 2018-04-12.
12. Christopher Skeels and James M Rehg. 2007. ShapeShift: A projector-guided sculpture system. In *Proc. UIST*, Vol. 7.
13. Amit Zoran, Roy Shilkrot, and Joseph Paradiso. 2013. Human-computer interaction for hybrid carving. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, 433–440.