Augmented Foam Sculpting for Capturing 3D Models

Michael R. Marner* Bruce H. Thomas†

University of South Australia

ABSTRACT

This paper presents a new technique to simultaneously model in both the physical and virtual worlds. The intended application domain for this technique is industrial design. A designer physically sculpts a 3D model from foam using a hand-held hot wire foam cutter. Both the foam and cutting tool are tracked, allowing the system to digitally replicate the sculpting process to produce a matching 3D virtual model. Spatial Augmented Reality is used to project visualizations onto the foam. Inspired by the needs of industrial designers, we have developed two visualizations for sculpting specific models: Target, which shows where foam needs to be removed to produce a model, and Cut Animation, which projects the paths for cuts to be made to reproduce a previous artifact. A third visualization of the wireframe of the generated model is projected onto the foam and used for verification. The final visualization employs 3D procedural textures such as a wood grain texture, providing a simulation of volumetric rendering. Volumetric rendering techniques such as this provide a more natural look that is projected onto the foam. Once the object has been modeled physically and virtually, the designer is able to annotate and paint the finished model. The system has been evaluated through a user study conducted with students from the School of Industrial Design at the University of South Australia.

Keywords: Spatial Augmented Reality, User Interfaces, 3D modeling.

Index Terms: H.5.2 [Information Interfaces and Presentation]: Graphical User Interfaces—Input Devices and Strategies; I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction Techniques

1 INTRODUCTION

Industrial design typically involves an iterative process where the designer progresses from hand drawn sketches, through to handmade mockups, CAD models and finally rapid prototype mockups before arriving at a final design. This paper presents a new system aimed at improving this process through the use of Spatial Augmented Reality (SAR). SAR allows digital objects, images, and information to be added real world artifacts by projecting onto surfaces in the environment with digital projectors.

The aim of our new technique, Augmented Foam Sculpting, is to simultaneously capture a digital model while a user sculpts a physical version. A designer physically sculpts a 3D model from foam using a hand-held hot wire foam cutter. Both the foam and cutting tool are tracked, allowing the system to digitally replicate the sculpting process to produce a matching 3D virtual model. SAR is employed to project visualizations on to the foam to enhance the modeling process.

Constructing mockups and prototypes by carving light weight Styrofoam with a hot wire foam cutter is a common technique for industrial designers. This aids in developing ideas and design concepts. Designers can also find problems with designs that are difficult to identify without a physical mockup. Once these mockups are made, they are used as a basis for CAD models.

Our system allows designers to create 3D models of designs with traditional techniques already in use. Using traditional foam sculpting tools, the designer creates a foam mockup of their design. At the same time, the system captures the designer's actions as they are being performed. The system models the designer's actions to produce a matching digital model of the object and is able to playback these actions for additional tasks, such as training.

Previous work has investigated reverse engineering the CAD models from physical mockups [7]. This process involves scanning the object and processing the resulting point cloud to produce a usable model. Our system completely removes the need for this step, as the 3D model is generated while the sculpting is taking place. Virtual models created with the foam cutter are often less complex than the scanned model, as vertices are only added where a cut occurs.

This system also provides new abilities and more flexibility during the design phase. The designer can save versions of the CAD model at different stages of sculpting, allowing them to explore several ideas in the digital realm, based on a single physical mockup. The foam model is enhanced by projecting information onto it. The system allows the designer to digitally annotate the model with a stylus, and airbrush paint onto it. The paint textures can be saved with the model for future reference. Furthermore, SAR can be used to instruct the user how to produce a specific model. Two visualizations have been developed to support this instructive process: Cut Animation, which projects cutting paths onto the object, and Target Visualization, which colors the foam to illustrate areas to be removed. SAR is also be employed to test different materials by lighting the foam mockup with projected textures.

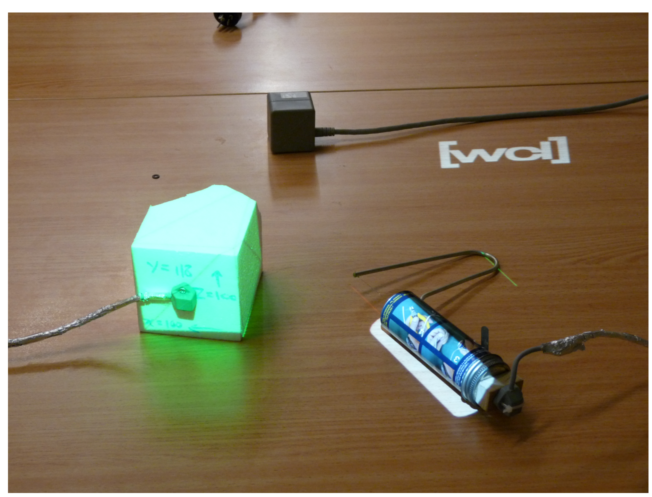

Figure 1: The Augmented Foam Sculpting System

The contributions of this paper are the Augmented Foam Sculpting technique for capturing 3D models, and the SAR visualizations

*e-mail: marnermr@cs.unisa.edu.au

†e-mail: bruce.thomas@unisa.edu.au

used to aid the user during the sculpting process. The augmented sculpting technique is novel and intuitive. It uses tools already in widespread use. Augmented Foam Sculpting can be integrated with very little change in industrial design workflow. The other contribution is the set of SAR visualizations, which aid the designer to create specific models with the system. Many more visualizations could potentially be incorporated into this system. Therefore, the visualizations presented in this paper provide a foundation of visualizations which could be extended in the future.

The remainder of this paper is organized as follows: first, we detail the rationale for employing SAR over other AR display technologies. This is followed by a review of relevant research. We then present our Augmented Foam Sculpting system and the SAR visualizations implemented to enhance the system, with a detailed description of their implementations. The limitations of our implementation are discussed, along with suggested modifications that can overcome these limitations. We describe our prototype implementation, and report on feedback from an informal user study. Finally we conclude with a look towards future work and a summary of this project.

1.1 Spatial Augmented Reality

As previously mentioned, Spatial Augmented Reality is a technique for adding digital objects, images, and information onto physical objects by projecting onto them with digital projectors. SAR benefits from the natural affordances offered by physical objects; the user can physically touch the objects virtual images are projected onto. SAR also benefits from the natural depth cues of physical objects, as SAR systems change the appearance of a physical object's 'skin', rather than adding purely virtual objects. As the display technology of SAR is placed into the environment, individual users are not required to wear or carry the display. Where head worn displays often provide a narrow Field Of View (FOV), and handheld devices provide a small window into the augmented world, the range of SAR systems can easily be extended by adding more projectors. The FOV of the overall system is the natural FOV of the user, allowing them to use their peripheral vision.

The nature of SAR also has limitations. SAR systems require physical surfaces to project onto; it is not easy to place a virtual object in 'mid air' in a SAR environment. It is not possible to achieve high quality rendering with view dependent lighting without tracking the user, and with a group of users, these effects would only appear correct from the specific tracked viewpoint of a single user.

1.2 SAR as a Design Tool

Projector based SAR systems have the potential to improve processes in a variety of application domains. We have chosen to use SAR over other display technologies for several reasons. SAR places the display into the environment, allowing the user to experience visualizations in an unencumbered manner and from any angle. Users must be able to interact with the systems without their movement being restricted, such as if they need to wear or carry equipment. The number of users for a particular SAR system is only limited by the size of the space and not the amount of equipment, allowing the system to easily accommodate multiple users simultaneously. This is not the case for other AR display types, such as head worn displays and handheld systems, which require equipment for each user. This is especially useful for group collaboration. People are able to view each other, the physical world, and the virtual world unencumbered by technology. The passive haptic nature of SAR can enhance the user experience, as users are able to touch the design artifact with their hands. Our system takes advantage of the physical nature of SAR to accomplish 3D modeling. The user sculpts with a natural, bimanual technique. The foam is real, meaning the user benefits from natural depth cues, which are often missing in other AR systems.

1.3 Physical-Virtual Tools

As previously mentioned, the nature of SAR requires physical objects to project onto. Unlike HMD and handheld devices, it is not possible to draw an object in mid air, or display menus and other information at a predictable location on the image plane of the display. SAR systems require new user interface techniques to address this issue. Our approach to this problem is to construct the user interface from Physical-Virtual Tools (PVT) [14], which encompass the entire UI for SAR applications. The user interacts with objects in the system through these tools, and the system provides feedback by projecting information on the tools. Several tools with different form factors are provided, so the best one can be chosen for a specific task. Projecting onto tools allows for a meaningful overloading of functions. This is a tradeoff between a large number of single purpose tools, and one tool responsible for all interaction. A secondary monitor or a fixed projection screen is not required, as the current state of the tools is displayed on the tools themselves.

Augmented Foam Sculpting employs four different PVT's with varying levels of virtualization. The Airbrush, Stencil, and Stylus have been presented previously. This system uses these three tools, and introduces a new tool for foam cutting. The primary tool in this application is the foam cutter. This is an example of a highly specialized tool, serving only a single purpose. The stylus is another single purpose tool used for annotation. The tool has been designed with a flat surface to project onto. The tool is virtualized in that the active color is projected onto the tool. The airbrush is used both for airbrushing onto objects, and as a virtual laser pointer. Here, the projection changes depending on the mode of operation. In paint mode, the spray angle and paint color is projected onto the tool. The area around the top of the device illuminates when a paint operation is active. The projection changes to an arrow to indicate virtual laser pointer mode. Finally, the stencil tool is a tracked board. This acts as a Personal Interaction Panel [28] for our system, but also to mask areas of the paint when airbrushing. The stencil mask is virtual; the user can create new shapes by drawing onto the board.

2 Related Work

This project builds upon previous work in 3D modeling for desktop and AR systems, Tangible User Interfaces (TUI), and SAR research.

An early form of TUI was introduced by Fitzmaurice et al. [5]. Graspable user interfaces consist of physical objects the user manipulates in order to interact with the system. This was extended [29] to identify physical analogues to traditional GUI components. Previous research has shown that both two handed interaction techniques [13], and the use of physical props [6], can enhance the user experience.

AR has been used for verifying 3D models, and to test design elements such as materials and colors. Augmented Foam [11] uses AR to overlay 3D models on top of matching rapid prototypes. SAR has also been used for design verification. WARP [30] allows designers to select and modify materials for designs by projecting onto rapid manufactured models. Our system moves from using SAR for verification and changing cosmetic properties to actually creating 3D models.

Spray Modeling [8] offers an intuitive way of developing 3D models. While traditional applications require the user to create a model through explicitly adding and modifying polygons, Spray Modeling allows the user to build up 3D models by spraying virtual material onto a base mesh, in a similar way to how an artist sculpts with clay. SKETCH [31] uses a stylus based interface. A user builds a 3D model by successively drawing and refining an object from different angles. ModelCraft [26] also uses natural interaction for annotating and changing 3D models. Paper or rapid prototype models are printed with special patterns. A user annotates changes for the design using special pens that track their position on the object's surface. A gesture based command system allows the user to specify changes for the 3D model. This is a simple, intuitive interface, but the user cannot directly modify the 3D model. Our work builds on the idea of mimicking physical tools for 3D modeling.

Surface Draw [22] allows a user to sculpt 3D models in a VR environment. The user defines a path for a surface through the motion of their hand. Our system shares a similar motivation to Surface Draw. While their process is additive, ours is subtractive in that we are removing geometry from an existing object. In addition, our system benefits from a physical object, where as Surface Draw is a completely virtual system.

Digital Foam [25] is a unique input device using conductive foam that is manipulated by the user's hands. As the foam is compressed, the resistance through the foam changes. This change is measured by an array of sensors. The user manipulates 3D models by gesturing into the foam, such as pinching. The main difference between digital foam and our work, is Augmented Foam Sculpting has the user sculpt a physical piece of foam to create a 3D model, where as Digital Foam is a computer input device allowing gestures as a means of modeling in a traditional CAD environment. Illuminating Clay [16] is a system that allows the user to sculpt a landscape with clay. A ceiling mounted laser scanner captures the height of the clay to produce a depth map. Visualizations are then projected onto the clay. The main limitation of this system as a modeling tool is it can only create height maps of the surface. Our system can be used to create arbitrary 3D objects.

Our software is inspired by Shader Lamps [19], a system for changing the appearance of physical objects by projecting computer graphics onto them. Shader Lamps has been used to simulate different material properties, or to light scenes with non-photorealistic shading models [20]. Large scale systems have been used to create human scale virtualized scenes [17]. Shader Lamps has been extended to support interactivity [2]. A tracked stylus is used to digitally paint onto objects, with a color palette projected onto the desk. Shader Lamps requires a calibration step to find the intrinsic and extrinsic parameters of the projectors. While the original Shader Lamps system uses a manual calibration process involving finding projector coordinates for known points in 3D, it is possible to automate this process. Raskar et al. [18] introduce techniques for automatic projector calibration through the use of cameras.

Recently, SAR has been used in a variety of interactive systems. CADcast [15] gives instructions to a worker by projecting onto surfaces. SAR has been introduced to medical surgery [24], in which it is used for projecting procedure information directly onto the patient, such as marking the location for an incision. Manufacturing has benefited from SAR [23], where a laser projector is used to display the location for weld points onto car parts. SAR has also been used as a teaching tool. For example, teaching a novice how to play billiards by projecting onto the table surface [27]. In a presentation at HCI [1], Akaoka and Vertegaal demonstrate how SAR can be used to prototype user interfaces on foam design mockups.

3 Augmented Foam Sculpting

Augmented Foam Sculpting (see Figure 1 for the physical setup) allows a designer to create 3D models using traditional sculpting techniques and tools already used in the industrial design process. The user begins with a block of Styrofoam, and uses a hot wire foam cutter to remove pieces of foam to produce the design artifact. Both the foam and the cutting tool are tracked with six degree of freedom sensors. The system digitally replicates the sculpting as a series of Constructive Solid Geometry [10] Boolean difference operations. The geometry removed from the digital model is the same as what is removed from the physical foam. The end result is a physical foam design artifact and a matching 3D virtual model that can then be used for further work in a traditional CAD environment. This ties in with existing industrial design processes, where a CAD model is often created based on a physical prototype. The foam mockup can immediately be used within a SAR system. We have integrated Augmented Foam Sculpting with our previous airbrushing system allowing the designer to paint and airbrush onto their design.

This approach fits in well with current design techniques. Previously, designers would sculpt mockups to test design ideas, then model these in a CAD package, or use a 3D scanner to obtain geometry from the mockup. In our system, 3D modeling is combined with sculpting. The user does not need to learn a CAD program, and can utilize the sculpting skills they already have.

To create a model, the user first enters the dimensions of the foam block so an initial virtual model can be created. This model is then projected at a known location on the workbench. The user places a tracking sensor anywhere on the foam, and places the foam over the projection. The system records the offset from the tracker to the foam. This allows the user to quickly change pieces of foam without having to perform a time consuming calibration process at startup. The user can then create the virtual model by sculpting the foam.

3.1 Performing CSG Operations

Sculpting is modeled as a series of CSG Boolean difference operations. The cutting tool and foam are tracked with six degrees of freedom. A cut is initiated when a line segment representing the cutting wire intersects with a bounding box of the foam. The position of the line segment is recorded as it passes through the foam. Due to limitations in the tracking system, we only record points that are at a distance of 1mm or more from the previously recorded point. When the line segment leaves the bounding box, the cut is complete, and the recorded points are used to modify the 3D model.

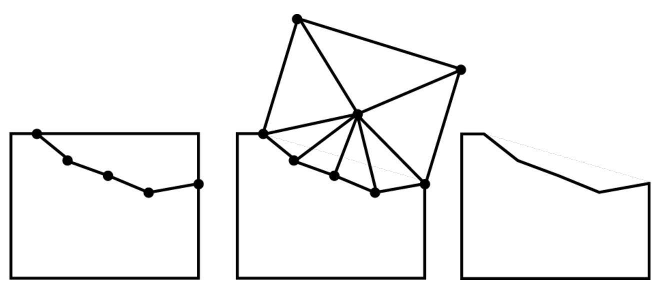

Figure 2: The cut path through the foam (left), constructed cut geometry (center), and result of Boolean operation (right)

Geometry must be generated from the cut path so that the Boolean operation can be performed. New vertices must be added in a position outside the foam geometry. The position of these vertices is based on the first and last recorded line segment. A direction vector for the position of the new vertices is calculated from the center of the foam to the average between the four points of the line segments. The direction vector is used to place the new vertices away from the original line segments and outside the foam geometry. For example, if the points representing the line segment are stored in an array, the new positions are calculated like so:

```
Pavg = (P[0] + P[1] + P[nPoints -2]
+ P[nPoints -2]) / 4

Direction = Pavg - Centroid

NP0 = P[0] + Direction*Offset)
NP1 = P[1] + Direction*Offset)
NP2 = P[nPoints -2] + Direction*Offset)
NP3 = P[nPoints -1] + Direction*Offset)
```

In the equation above, Offset is simply a number large enough to ensure the new vertices will be outside the foam. This value does not need to be calculated for each cut, as the physical dimensions of the cutting tool limit the maximum size of cuts. Based on the size of the cutting tool, our prototype uses a value of 100mm. These vertices are then used to generate a triangle strip representing the path the cutting wire took through the foam. Finally, side walls are constructed. The center of all vertices comprising the side is found, and a new vertex is inserted. A triangle fan is then constructed to form the side wall. The final product of this construction is an enclosed 3D volume suitable for CSG operations. A Boolean difference operation is performed, simulating the volume of foam that was removed by the cutting tool. A 2D cross section of this operation is illustrated in Figure 2. The new foam geometry matches the foam within the limitations of the tracking system, and is used for further sculpting operations.

3.2 Handling Difficult Cuts

Augmented Foam Sculpting functions correctly with concave cuts with several curves. The algorithm will generate correct cut geometry even if the cut path passes through the centroid of the bounding box. The degenerate case is if the user attempts to remove more than half the foam in a single cut. Here, *Direction* must be reversed so the correct section of geometry is removed. This condition is easily tested for by comparing the distances of *Centroid* and *Pavg* to the position of the tracking sensor, ensuring *Centroid* is closer. In practice, this rarely occurs, since the physical size of the foam cutter prevents such drastic cuts.

The CSG operation may fail for extremely concave shapes due to the way side walls of the cut geometry are generated. The centroid calculated for the side walls can be outside of the cut path, causing an invalid surface to be constructed. A simple modification to our algorithm using a more advanced triangulation technique [9] would resolve this problem. This scenario is illustrated in Figure 3. In testing, we have not found this to be an issue, as users generally make small, rather than drastic, complex cuts into the foam.

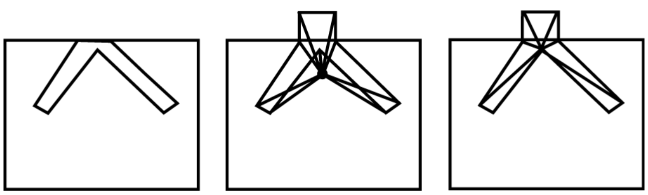

Figure 3: Highly concave cut paths (left) cause invalid geometry to be constructed (center). In this scenario, side walls should be triangulated as shown (right)

As the entire cut through the bounding box is recorded, cuts into small, concave areas will be modeled correctly. This is because the user must remove the foam cutter from the bounding box without touching the physical foam. However, the system will generate invalid cut geometry in cases where the path through the foam crosses itself. In a future iteration we would like to modify our algorithm to swap vertices that cause the path to cross itself before performing the Boolean operation. This will result in the correct cut being performed.

4 SAR Visualizations

While the modeling system itself can function without any video output, the usability of the system can be greatly increased through the use of SAR. The following section describes the visualizations we have implemented to enhance the system: Cut Animation, Target, and volumetric rendering. Additionally, we describe how the models can be virtually painted onto with a SAR system. All of these visualizations are projected onto the foam and updated in real time as the user sculpts.

4.1 Cut Animation

Cut Animation (Figure 4) displays the cuts required to reproduce a model, one after the other. This visualization can be used for instruction. For example, an expert can use the system to produce an object, with the system recording the cuts made. The cuts can then be played back to a novice user, instructing them how to produce the object. The Cut Animation technique can guide the designer to reproduce any of the previously saved versions of a design.

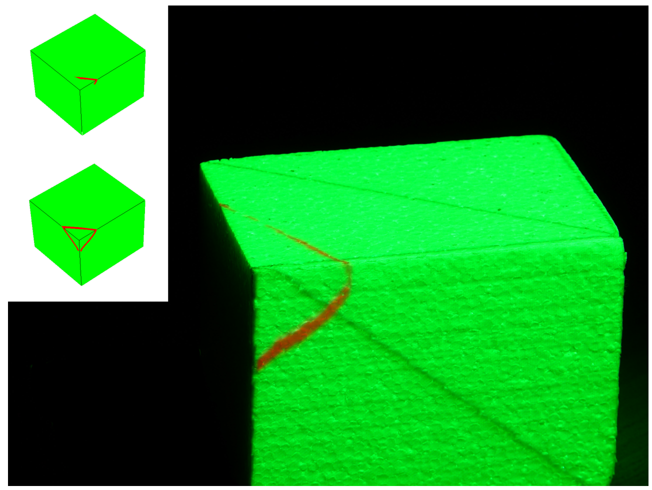

Figure 4: Animating the path for a cut by projecting onto the foam

This visualization projects lines representing the path taken by the cutting tool through the object. To aid in understanding, the line is animated such that the cut appears at the start and moves through the object. The user follows the lines to make the cut, and the system then moves to the next cut. This process continues until the model is complete.

The visualization is implemented using two Frame Buffer Objects (FBO) and GLSL shaders. The rendering process consists of three stages:

1. The foam is rendered as normal to a FBO. This captures both the color and depth information of the foam.

2. Geometry representing the cut through the foam is drawn to another FBO. However, we only draw to the FBO if the current fragment depth is similar to the depth recorded for the foam at the same pixel location. This ensures that we only draw on the surface of the foam, and prevents self occlusion of the cut geometry for complex paths.

3. The two FBO are combined in a final shader. Here, we draw the foam to the screen. However, for fragments containing the cut, we draw the cut. This has the effect of superimposing the cut path on top of the foam geometry.

By increasing the number of cut points drawn over time, we can animate the path the cutting tool took through the foam, illustrating to the user how the cut should be made. This approach is quite flexible, and much less computationally intensive than if we were to calculate intersection points between the foam and cut path. As the visualization only involves drawing the path as geometry, it can handle complex cut paths with many curves.

4.2 Target

Target (Figure 5) visualizes the sections of foam that must be removed from the foam to match a target 3D model. This visualization would also be useful to create a modified version of an earlier design mockup.

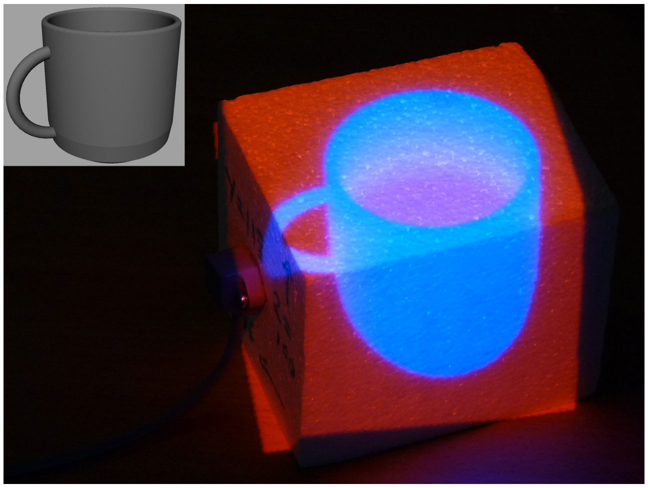

Figure 5: Target Visualization onto foam block, inset: target model

This visualization represents sections to be removed as a color gradient projected onto the foam. Green areas on the foam indicate that no more cutting is required. The remainder is colored using a gradient from red to blue. Red indicates a large area should be removed, and blue indicates that only a small area needs to be removed.

This visualization also utilizes depth information to generate the color gradient. The scene is organized with the target model placed inside the bounds of the foam. Two FBO are used to individually record the depth information of the foam and target model. The two depth textures are then utilized in a fragment shader. For each fragment, the two depths are compared. If the foam depth is greater or equal to the target no more foam should be removed. Otherwise the color is mixed between red and blue, depending on the difference in depths. Due to the non-linear nature of the depth buffer, the differences in recorded depths are very small. To achieve a reasonable color blend, the mix value is multiplied by a constant. The value of this constant varies depending on the distance from the projector. In testing, We found a value of 0.005 to be an appropriate constant. Pseudocode for the fragment shader is shown below:

```
void main()
{
  /* discard empty fragments */
  if (foam == 0)
    discard;
  if (foam >= target)
    output GREEN;
  else
  {
    mixValue = (target - foam) * dConst;
    output mix (BLUE, RED, mixValue)
  }
}
```

All calculations for this visualization are performed from the point of view of the projector. This limits the use of this visualization to single projector systems, as the color calculated for a single point on the foam will be different depending on the location of the projector. In our prototype we have used an over the shoulder projection setup, making the point of view of the user is similar to that of the projector. This limitation can be resolved by performing calculations from the point of view of the user. The resulting gradient texture can be re-projected from the point of view of the projectors, such that each projector displays the same color for each point on the foam. However, our current system does not track the user.

4.3 Volumetric Texturing

The SAR projection system is also used to project different textures and information onto the foam while the system is in use. The wireframe (Figure 6) of the 3D model is projected onto the foam during the sculpting session. This allows the designer to easily verify that the generated 3D model is correct.

Figure 6: A user sculpts into the foam, with the wireframe projected onto the physical model

In simple testing, we found that the wireframe can be distracting to have projected all the time. We have experimented with projecting 3D textures onto the foam, with the ability to switch to the wireframe when needed. An example is retexturing the foam as wood using a 3D procedural texture (Figure 7). The texture is implemented as a GLSL shader, using the approach described by Rost [21]; we have simply removed the view dependent lighting component as we are not tracking the user. Using procedural textures has the advantage that no 2D texture coordinates are needed, the position of the fragment in 3D space can be used. Generating 2D texture coordinates is a computationally expensive process, and these must be updated each time a cut is made. The wood appearance also remains consistent as cuts are made, giving the illusion that pieces of wood are actually being removed. New textures and materials are easily added through the pluggable GLSL shader system.

4.4 Painting the Prototype

An important requirement for this system is that once the designer has finished sculpting a design artifact it can be used for later stages of the design process. This could be in traditional CAD packages, however we have developed a system that allows for iterative design entirely in the SAR domain.

Our previous work [14] introduced a SAR digital airbrushing system. One major limitation of this system is it required 3D models of the design artifact before the designer was able to paint. The models used were created in a 3D modeling package, which can be quite time consuming.

Augmented Foam Sculpting addresses this by allowing the designer to sculpt a design, and have the 3D model immediately avail-

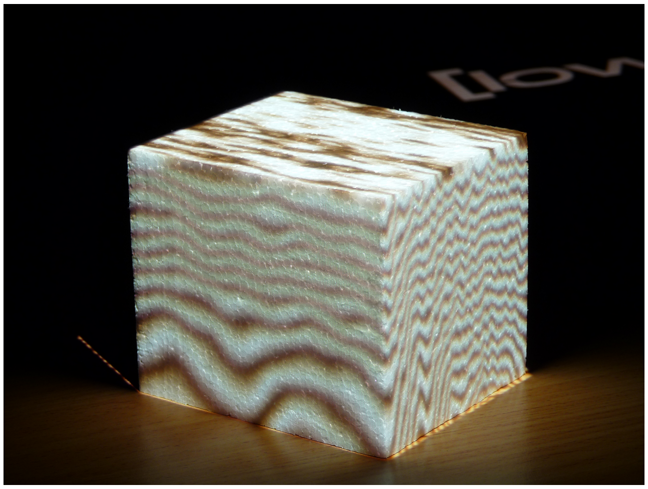

Figure 7: Projecting a 3D procedural wood texture onto the foam

able for further work. We have provided the interface between the sculpting and painting functions, so the designer is able to airbrush and annotate on their design. 2D texture coordinates are generated for the object. Multi-texturing is used to provide a paint and annotation layer. These textures can be saved to disk independently, along with the 3D model using common file formats for later use. A designer can then use these to produce a high quality ray traced render of their design using existing software.

5 Implementation Details

Our prototype is implemented as a C++ application using OpenGL, running on Linux. The system runs on a single desktop PC, with a 3GHz Core 2 Duo processor, and Nvidia Geforce 9400 graphics card, driving up to two projectors. The tools and foam are tracked with six degrees of freedom, using a Polhemus Patriot magnetic tracking system. In practice we have found the magnetic tracker to suffer from interference. This reduces the quality of the generated 3D models, as they do not always match the physical version. Sculpting is modeled as a series of Constructive Solid Geometry (CSG) operations. Our software makes use of the CGAL [4] library to perform CSG calculations and to generate 2D texture coordinates for models. In some instances, CGAL will raise an error when performing CSG operations. We have implemented an adaptive resampling technique to avoid this issue. A CSG operation is attempted, and if it fails the cut points are resampled at a higher minimum distance. The distance is incremented at 1mm intervals until the cut succeeds or a maximum number of attempts have occurred.

5.1 System Calibration

Several components must be calibrated before the system can be used. Our calibration process is outlined in this section. Projector calibration involves finding the intrinsic and extrinsic parameters of the projectors, so that projected graphics align correctly with real world objects. Our calibration process is the same as described by Bimber and Rasker [3]. Known points on a reference box are found in the projector's viewport by moving a crosshair into position. These points are used to derive intrinsic and extrinsic matrices for the projector. This process takes approximately five minutes per projector, but only needs to be performed when projectors are moved. This is acceptable for our current setup consisting of two projectors, but will not scale to larger systems. An automated process such as those presented in [18, 12] will need to be implemented.

Transform matrices must be obtained to transform tracking data from sensors into the projector coordinate system. To do this, we project a 3D model of the Polhemus sensor onto known positions on the workbench. This allows us to find the X and Y vectors for the tracking system, and through the cross product, the Z vector. We can then construct a transform matrix. This process must be completed once for each tracking system. The process takes less than one minute to complete, as only three points need to be obtained.

Finally, an offset matrix must be obtained for transforming tracking data to the local coordinate system of the foam. This process must be simple and fast. A user is able to quickly change to a new piece of foam without having to precisely align the sensor. Our system creates these transforms on startup. A 3D model of the foam is projected to a known location on the workbench. The user places the foam at the projection, and the system calculates the offset matrix from the tracking data.

6 Evaluation

This section describes the evaluation of our system. We have provided quantitative data including measurements of system performance, and comparisons of the system output to that of a commercially available laser scanner. We also have qualitative data in the form of an expert review of the system.

6.1 Expert Review

The system was demonstrated to a group of final year undergraduate and honors students from the Louis Laybourne Smith School of Architecture and Design at the University of South Australia. The system and visualizations were demonstrated, with students able to make cuts into the foam. This was followed by an open ended discussion to gain feedback on the system.

The student response to the system was quite positive. The most compelling aspects of the system from their point of view included the real time update of the virtual model during the carving process and the ability for the system to help in planning foam cutting. One student noted "the ease in which a CAD model would be produced, through manual labor and the freedom to possibly carve foam accurately without the need for templates", another noted the system "will help with any planning of foam cutting, making errors more unlikely".

While this paper has focused on a workflow that moves from foam sculpting into CAD, students commented that they also work in the other direction. Sometimes, they use foam sculpting an inexpensive alternative to 3D printing. In this scenario a lot of their time is spent in planning and producing stencils to cut against to produce a physical version of the CAD model. Our system supports this workflow through the Cut Animation and Target visualizations. The addition of 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.

6.1.1 Suggestions for Improvement

Several students stated they would like to see new tools integrated into the system. In particular, students would like to see a rasp or the ability to sand the model, as these tools would allow much finer modifications to be made than what is possible with the cutting tool. Another tool suggested was a soldering iron, which would provide the ability to put holes and other patterns onto the surface of the foam.

One student suggested a more realistic physical simulation, taking into account the properties of the foam. For example, if the cutting wire is held at a constant position during a cut, the area of foam around the wire would begin to melt away.

The students also stated they would like to have the ability to have high resolution textures projected onto the models. The quality of the images projected onto objects is limited by the resolution of the projector, and the distance between the object and projector. This limitation can be reduced by using more projectors closer to the object, which would provide a higher spatial resolution.

6.2 Quality of 3D Models

A common workflow for industrial designers is to use a 3D scanner to produce a digital model of a design mockup for use in CAD software. Therefore, comparing the quality of 3D models generated with our system to those generated by a commercial 3D scanner is important. To compare, we created a simple foam sculpture with many small, concave cuts.. This model was then scanned with a Polhemus FastSCAN handheld 3D laser scanner. The results of the sculpting and scan are shown in Figure 8.

Augmented Foam Sculpting has the advantage that faces are only added to the model where a cut is made. Depending on the sculpture, this can produce a virtual model with vastly fewer polygons compared to the output from the FastSCAN, which produces a high polygon model with uniform vertex density. In the example in Figure 8, the Augmented Foam Sculpting system generated a mesh containing 684 triangles, where as the laser scanner's output contained 81,925 triangles. Augmented Foam Sculpting also captured the sharp edges of the cuts more accurately than the soft edges from the scanner.

The quality of the digital models is limited to the quality of the tracking system used. The spatial resolution of the projection system has no effect on the model resolution. It does however affect the quality of the visualizations, and therefore the ability for a designer to precisely match a target model. In our setup, a pixel is approximately 1.5mm square, which is better than the rated static accuracy of the tracking system (2.54mm).

6.3 System Performance

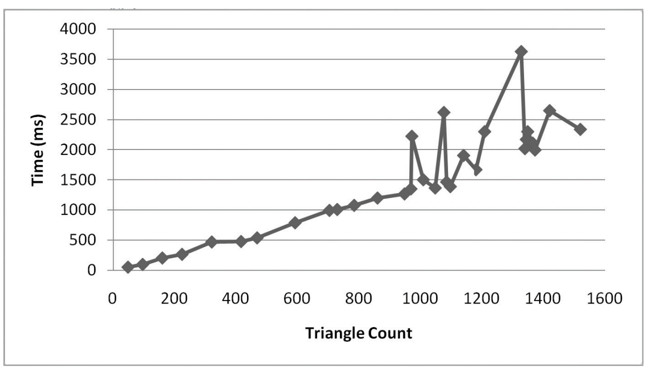

Figure 9: Calculation time for CSG operations as geometry size increases

Augmented Foam Sculpting requires many CSG operations to be performed during the sculpting process. The complexity of the foam geometry generally increases as more cuts are made. The CSG operations must be completed quickly so the user is not left waiting for an operation to complete before continuing to sculpt. The system was tested using blocks of foam approximately 100x100x118mm in size. With blocks of this size the mean cut size was 55 triangles, and mean CSG calculation time was 1466ms. However, this time varies greatly depending on the complexity of both the cut and foam geometry. The graph in Figure 9 illustrates the increase in time as the geometry complexity grows. The spikes in the graph can be attributed to the system resampling and reattempting to perform the CSG operation in the event of CGAL raising an error. In general, users only experience a short delay between cuts. The system frame rate is 60Hz. However, the prototype is a single threaded application, causing the video updates to pause while CSG operations are calculated. Our software uses CGAL, a popular computational geometry library. The CSG algorithm, as described by [10], has aspects that can be executed in parallel. Overcoming the current performance issues would require writing a new multithreaded CSG algorithm. It may also be possible to implement the algorithm with CUDA[1], taking advantage of the massively parallel nature of current graphics cards. These solutions, while desirable for a production system, are beyond the scope of this research prototype.

7 Conclusions and Future Work

This paper has presented a new technique, Augmented Foam Sculpting for generating 3D models using traditional tools for foam sculpting, already in use by industrial designers. By using the tools that designers currently use, designers do not have to change their workflow to begin using the system. By using foam blocks and a physical cutter, users benefit from the passive haptic feedback the objects provide, and the users can create 3D models with an intuitive bimanual interaction technique. The use of SAR allows us the projection of extra information onto the objects as they are being sculpted. We have used this ability to visualize how to produce target models from foam, either by changing the color of the surface to represent areas that should be cut away, or by 'playing back' cuts from a previous session. We have also used the projector to display the wireframe of the 3D model, or a wood grain texture onto the foam, as it is being sculpted. Although we have focused on the industrial design domain, augmented foam cutting could be used in other areas. The system would support any domain requiring physical objects with matching 3D models, or where 3D models are created from hand sculpted designs, for example the computer animation and visual effects industry. In the future we would like to extend this system to support extra tools. For example, it is not possible to add small grooves or indents into the foam using only the hot wire cutter. However, other tools such as a hot tip iron could be modeled using CSG, in a similar way to the cutter. We believe there is much more potential for SAR visualizations to improve our system, and will look towards new functionality and visualizations. We would also like to modify the algorithms that generate the cut geometry to address the limitations described in the paper, making the system more robust.

Acknowledgements

The authors wish to thank the design students from the Louse Laybourne School of Industrial Design who provided valuable feedback on our system.

References

[1] E. Akaoka and R. Vertegaal. DisplayObjects: prototyping functional physical interfaces on 3D styrofoam, paper or cardboard models. Boston, Massachusetts, 2009.

[2] D. Bandyopadhyay, R. Raskar, and H. Fuchs. Dynamic shader lamps: Painting on movable objects. In *IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 207–216, 2001.

[3] O. Bimber and R. Raskar. *Spatial Augmented Reality: Merging Real and Virtual Worlds*. A K Peters, Wellesley, 2005.

[4] CGAL. Computational geometry algorithms library.

[5] G. W. Fitzmaurice, H. Ishii, and W. A. S. Buxton. Bricks: Laying the foundations for graspable user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 442–449, Denver, Colorado, United States, 1995. ACM Press/Addison-Wesley Publishing Co.

[6] H. Hoffman, A. Hollander, K. Schroder, S. Rousseau, and T. Furness. Physically touching and tasting virtual objects enhances the

[1] www.nvidia.com/object/cuda_home.html

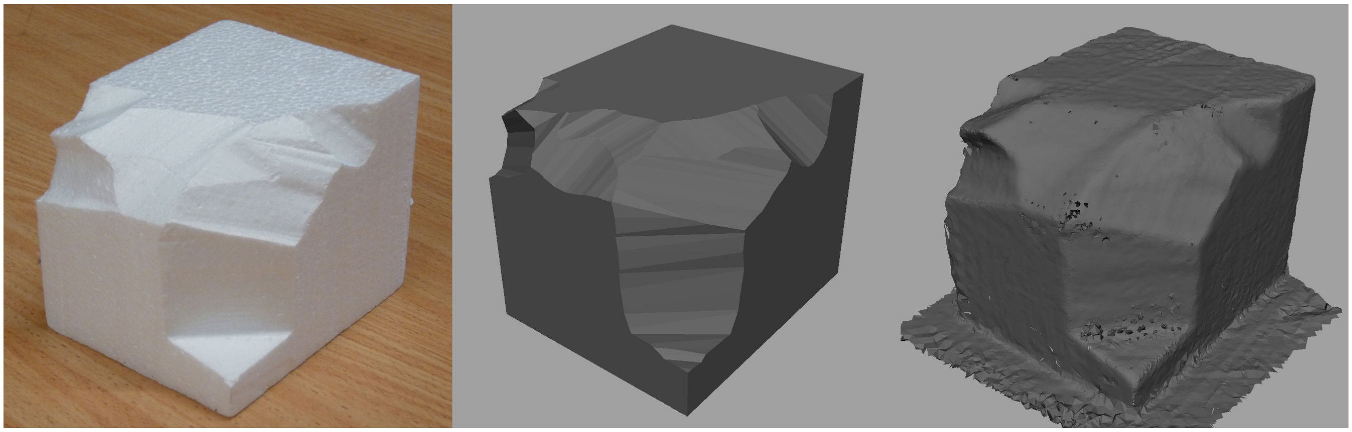

Figure 8: Comparison of physical foam model (left), Augmented Foam Sculpting (center), laser scanned model (right)

realism of virtual experiences. *Virtual Reality*, 3(4):226–234, 1998. 10.1007/BF01408703.

[7] S. Hsiao and J. Chuang. A reverse engineering based approach for product form design. *Design Studies*, 24:155–171, Mar. 2003.

[8] H. Jung, T. Nam, H. Lee, and S. Han. Spray modeling: Augmented reality based 3D modeling interface for intuitive and evolutionary form development. In *International Conference on Artificial Reality and Telexistence*, 2004.

[9] D. G. Kirkpatrick, M. M. Klawe, and R. E. Tarjan. Polygon triangulation in o(n log log n) time with simple data-structures. In *Proceedings of the sixth annual symposium on Computational geometry*, pages 34–43, Berkley, California, United States, 1990. ACM.

[10] D. H. Laidlaw, W. B. Trumbore, and J. F. Hughes. Constructive solid geometry for polyhedral objects. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 161–170. ACM, 1986.

[11] G. A. Lee, C. Nelles, M. Billinghurst, and G. J. Kim. Immersive authoring of tangible augmented reality applications. In *Mixed and Augmented Reality, 2004. ISMAR 2004. Third IEEE and ACM International Symposium on*, pages 172–181, 2004.

[12] J. C. Lee, P. H. Dietz, D. Maynes-Aminzade, R. Raskar, and S. E. Hudson. Automatic projector calibration with embedded light sensors. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 123–126, Santa Fe, NM, USA, 2004. ACM.

[13] A. Leganchuk, S. Zhai, and W. Buxton. Manual and cognitive benefits of two-handed input: an experimental study. *ACM Trans. Comput.-Hum. Interact.*, 5(4):326–359, 1998.

[14] M. R. Marner, B. H. Thomas, and C. Sandor. Physical-Virtual tools for spatial augmented reality user interfaces. In *International Symposium on Mixed and Augmented Reality*, Orlando, Florida, 2009.

[15] B. Piper and H. Ishii. CADcast: a method for projecting spatially referenced procedural instructions, 2001.

[16] B. Piper, C. Ratti, and H. Ishii. Illuminating clay: a 3-D tangible interface for landscape analysis. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*, pages 355–362, Minneapolis, Minnesota, USA, 2002. ACM.

[17] R. Raskar and K. Low. Interacting with spatially augmented reality. In *Proceedings of the 1st international conference on Computer graphics, virtual reality and visualisation*, pages 101–108, Camps Bay, Cape Town, South Africa, 2001. ACM.

[18] R. Raskar, J. van Baar, P. Beardsley, T. Willwacher, S. Rao, and C. Forlines. iLamps: geometrically aware and self-configuring projectors. In *ACM SIGGRAPH 2005 Courses*, page 5, Los Angeles, California, 2003. ACM.

[19] R. Raskar, G. Welch, K. Low, and D. Bandyopadhyay. Shader lamps: Animating real objects with Image-Based illumination. In *Rendering Techniques 2001: Proceedings of the Eurographics*, pages 89–102, 2001.

[20] R. Raskar, R. Ziegler, and T. Willwacher. Cartoon dioramas in motion. In *NPAR '02: Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*, page 7ff, New York, NY, USA, 2002. ACM. inproceedings.

[21] R. J. Rost. *OpenGL(R) Shading Language*. Addison-Wesley Professional, 2 edition, 2006.

[22] S. Schkolne, M. Pruett, and P. Schrder. Surface drawing: creating organic 3D shapes with the hand and tangible tools. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 261–268, Seattle, Washington, United States, 2001. ACM.

[23] B. Schwerdtfeger, D. Pustka, A. Hofhauser, and G. Klinker. Using laser projectors for augmented reality. In *Proceedings of the 2008 ACM symposium on Virtual reality software and technology*, pages 134–137, Bordeaux, France, 2008. ACM.

[24] B. Seo, M. Lee, H. Park, J. Park, and Y. S. Kim. Direct-Projected AR based interactive user interface for medical surgery. In *Artificial Reality and Telexistence, 17th International Conference on*, pages 105–112, 2007.

[25] R. T. Smith, B. H. Thomas, and W. Piekarski. Digital foam interaction techniques for 3D modeling. In *Proceedings of the 2008 ACM symposium on Virtual reality software and technology*, pages 61–68, Bordeaux, France, 2008. ACM.

[26] H. Song, F. Guimbretire, C. Hu, and H. Lipson. ModelCraft: capturing freehand annotations and edits on physical 3D models. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 13–22, Montreux, Switzerland, 2006. ACM.

[27] A. Suganuma, Y. Ogata, A. Shimada, D. Arita, and R. ichiro Taniguchi. Billiard instruction system for beginners with a projector-camera system. In *Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology*, pages 3–8, Yokohama, Japan, 2008. ACM.

[28] Z. Szalavri and M. Gervautz. The personal interaction panel - a two handed interface for augmented reality. pages 335–346, Budapest, Hungary, 1997.

[29] B. Ullmer and H. Ishii. The metaDESK: models and prototypes for tangible user interfaces. In *Proceedings of the 10th annual ACM symposium on User interface software and technology*, pages 223–232, Banff, Alberta, Canada, 1997. ACM.

[30] J. Verlinden, A. de Smit, A. Peeters, and M. van Gelderen. Development of a flexible augmented prototyping system. *Journal of WSCG*, 2003.

[31] R. C. Zeleznik, K. P. Herndon, and J. F. Hughes. SKETCH: an interface for sketching 3D scenes. In *ACM SIGGRAPH 2006 Courses*, page 9, Boston, Massachusetts, 2006. ACM.