Closed-Loop Shape Control of a Haptic Jamming Deformable Surface

Andrew A. Stanley[1], Kenji Hata[2], and Allison M. Okamura[1]

***Abstract*— A Haptic Jamming tactile display, consisting of an array of thin particle jamming cells, can change its shape and mechanical properties simultaneously through a combination of vacuuming individual cells, pressurizing the air chamber beneath the surface, and pinning the nodes between the cells at various heights. In previous particle jamming devices for haptics and soft robotics, shape has typically been commanded open-loop or manipulated directly by a human user. A new algorithm was designed for the three types of actuation inputs for a Haptic Jamming surface, using the depth map provided by an RGB-D sensor as shape feedback for closed-loop control to match a desired surface input. To test the closed-loop control accuracy of the system, a mass-spring model of the Haptic Jamming device generated three unique surface shapes as desired inputs into the controller with a mean height range of 25.1 mm. The average correlation coefficient between the desired input shape and the experimental output generated by the controller on the actual device across four trials for each shape was 0.88, with an average height error of 2.7 mm. When the desired input surface is generated from a 3D model of an object that a Haptic Jamming surface cannot necessarily re-create, the difference between the input and the output increases substantially. However, simulation of a larger array suggests that a Haptic Jamming surface can provide a compelling match for these more complicated shapes.**

I. INTRODUCTION

Haptic interfaces add a richness to human-computer or human-robot interaction not experienced by purely visual or auditory interfaces. Traditional haptic interfaces that consist of robotic linkages or tactile actuators can render forces or vibrations from a virtual, remote, or teleoperated environment to the user through a single point of contact. However, shape changing interfaces promise a more immersive experience by allowing multi-point contact in which the user can use his or her entire hands. For applications ranging from medical simulation to product design, the natural use of the entire hand for haptic exploration enables a more natural form of interaction through the sense of touch. The Haptic Jamming concept [1] in particular presents a promising form of such an encountered-type interface, as the combination of particle jamming and pneumatics allows for simultaneous control of mechanical properties and deformable shape across a continuous surface. Separate vacuum lines run to each cell in a multi-cell Haptic Jamming device to allow varied rigidity in each region of the surface. In this interface, particle jamming of individual cells in the array serves not only to adjust the mechanical stiffness that the user feels but also as a means to control shape. When positive pressure is applied to the air chamber below, soft cells balloon outward while rigid cells maintain their current curvature, and pinning the nodes between cells allows greater shape output variability. This methodology was originally developed for simulation of medical procedures that require direct exploration with the bare hand to make a diagnosis. For these scenarios, a Haptic Jamming device can display tissue properties ranging from soft to hard in any region of its surface and in a variety of form factors.

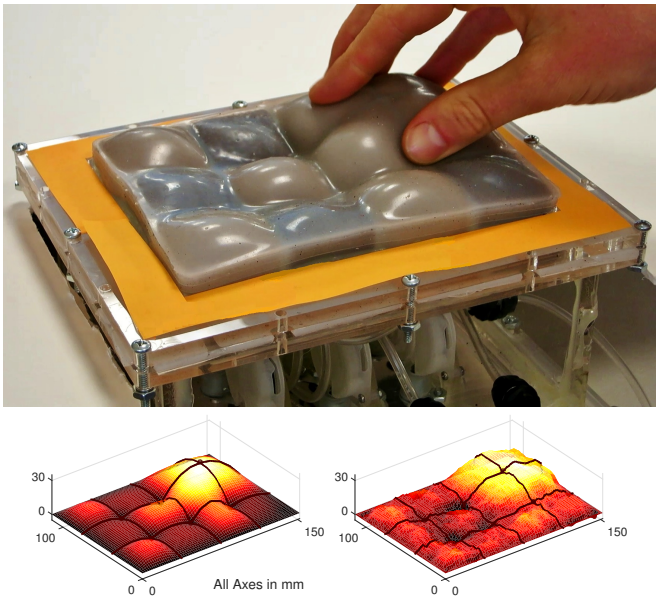


Fig. 1. A user interacts with a 12-cell Haptic Jamming surface (top) after it has reconfigured to match a desired surface (bottom left) using an overhead depth sensor for shape feedback (bottom right). Cells become darker as their density increases when they are jammed, making them more rigid to the touch as well as allowing them to hold the shape of their current configuration.

While previous research around Haptic Jamming focused largely on device design [1], including some basic open-loop control for shape outputs and controlling the mechanical properties across the surface [2], this work demonstrates the first closed-loop control of the output shape. The following section of this paper discusses prior work in the fields of soft robotics, particle jamming, and shape changing interfaces. Section III describes the setup of the system we use to demonstrate closed-loop control on a 12-cell prototype, including mechanical design improvements, additional system components, and an improved method for regulating the

*This work was supported in part by U.S. Army Medical Research and Materiel Command (USAMRMC; W81XWH-11-C-0091), the National Science Foundation (NSF; 1217635), and by the National Science Foundation Graduate Research Fellowship Program.

[1]A. A. Stanley and A. M. Okamura are with the Department of Mechanical Engineering, Stanford University, Stanford, CA 94305 USA. astan@stanford.edu, aokamura@stanford.edu

[2]K. Hata is with the Department of Computer Science, Stanford University, Stanford, CA 94305 USA. khata@stanford.edu

air pressure inside the chamber beneath the surface of the device. Section IV outlines the methods and algorithms we developed to control the shape of the device, which we then evaluate with both quantitative and qualitative metrics in Section V before concluding and discussing future work in Section VI.

II. BACKGROUND

The Haptic Jamming interface builds upon prior work in various fields ranging from soft robotics and particle jamming to shape changing interfaces. We discuss some of the most relevant previous research here.

A. Soft Robotics and Particle Jamming

The recent emergence of flexible and compliant actuation and sensing techniques has sparked extensive innovation in the field of soft robotics. In particular, soft robotics has greatly advanced biologically inspired robots by allowing deformable elements with much greater maximum strain while reducing the weight of the embedded actuators [3]. Particle jamming greatly increases the control over the mechanical properties of an object. These features have led to the development of a sphere for robotic locomotion [4], novel robotic manipulators [5], universal end effectors [6], a flexible endoscope with controllable rigidity [7], novel user interfaces [8] [9] [10], and a wearable force display [11]. For the majority of these systems, either the user controls the deformation of the particle jamming components or the exact shape is not a critical aspect of the functionality. For those that do rely on computer-controlled deformations, like the locomotive sphere and robotic manipulator, such control is conducted open-loop.

B. Shape Changing Interfaces

An idealized form of 3D user interface for input into and output from a computer would be a tactile display or deformable object that can change shape under the control of both the user and the computer with both precise actuation and sensing, conceptualized as "Digital Clay" [12]. Implementations of such displays often fall under either a "bed of nails" or "deformable crust" approach, although Haptic Jamming partially hybridizes the two with its array of pinnable nodes and jamming cells and compliant surface. The actuators in the "bed of nails" can range from shape memory alloys [13] and pneumatics [14] [15] to motor-driven linear actuators [16], so the shape control consists primarily of down-sampling the desired input shape to the resolution of the array and positioning each element accordingly. Thus, the number of actuators scales quadratically as the size of the array grows, but one implementation with hydraulic actuators arranged along the rows and columns of the array [17] allows the number of actuators to scale linearly with increased resolution with a controller based on singular value decomposition [18]. "Deformable crust" implementations, like the SmartMesh multi-loop mechanism [19] and the Formable Object [20], rely on extendable links arranged in a double-layer square grid and kinematics optimized to render cylinders or spheres.

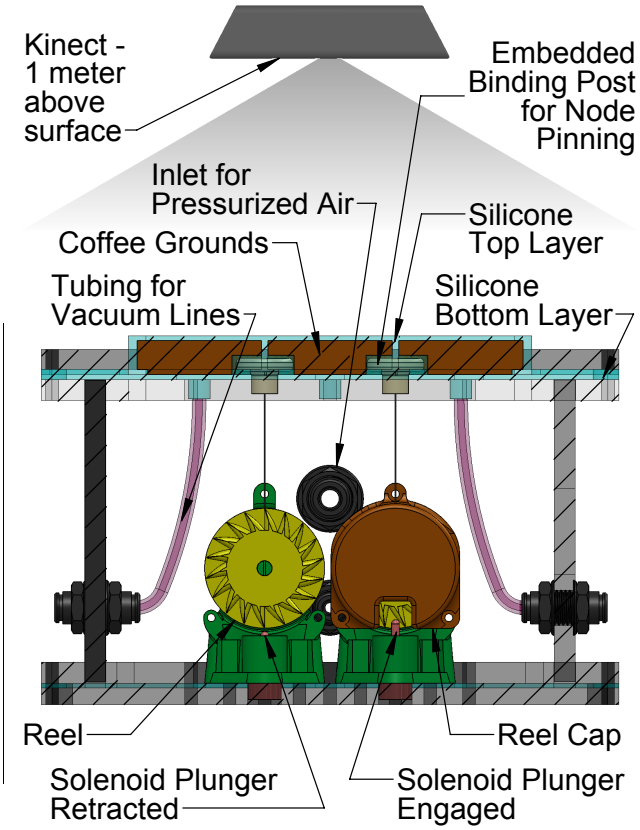


Fig. 2. A cross section of the 12-cell Haptic Jamming device shows one column of cells and nodes with the underlying mechanisms for cell jamming, node pinning, and chamber pressurization.

III. SYSTEM SETUP

The system used in this work included several design changes from the multi-cell prototypes in [1] and [2] to increase robustness and improve repeatability between trials, as well as various new components to add shape feedback and closed-loop control.

A. Hardware and Mechanical Design

Fig. 2 shows a diagram of the components in a cross section of the Haptic Jamming device. The surface itself consists of two layers of Ecoflex 30 silicone (Smooth-On, Inc.) cured in custom-cut acrylic molds and glued together. Coffee grounds fill the 38 mm (1.5 in.) square cells in the membrane with separate vacuum lines running in from below to jam individual cells. The air chamber includes twelve inlet ports for the vacuum lines that each flow through a separate V1-C04-BW1 electronic 3-way valve (Mead Fluid Dynamics), one inlet port for pressurized air that flows in through an EV-P-20-4050 air flow regulator (Clippard Minimatic), and one port to connect a SSCDANN005PGAA5 pressure sensor (Honeywell International, Inc.). A custom-designed binding post with a threaded insert is embedded into the top layer of silicone during the curing process. A vented screw connects the binding post to the nylon node-pinning line that wraps around the reel in the node-pinning mechanism below. The plunger of a S-63-38S pull-type tubular mini solenoid

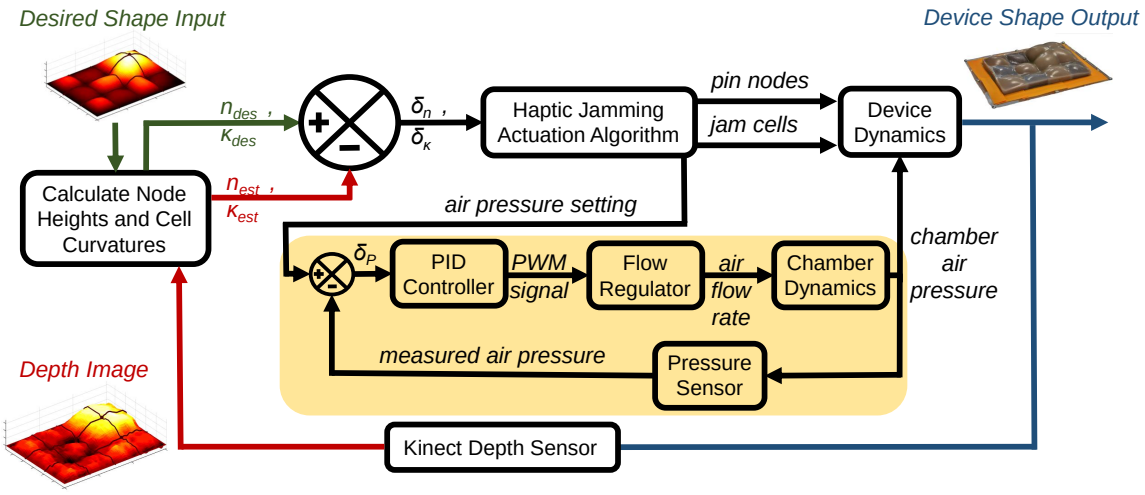


Fig. 3. A block diagram of the system control architecture. The outer loop runs once per iteration of Algorithm 1 while the inner pressure control loop, highlighted in yellow, runs continuously.

(ElectroMechanicsOnline) extends into the teeth of the reel to lock it in place or retracts to allow it to spin freely when an increase in chamber pressure balloons the surface outward. A torsional spring in the reel winds up the line when the air pressure is released and the surface is unjammed. The tooth count on the ratchet and the diameter of the section of the reel around which the nylon line wraps result in a height resolution of 4.7 mm per tooth. While this limits the resolution of the shape display, the mechanism allows the use of a low-power actuator that performs no direct work on the system; rather, the passive mechanical components perform the work. A Kinect for XBox 360 (Microsoft Corporation) mounts one meter above the surface, the minimum distance that its depth sensor can measure. At this distance, each pixel of the connect covers about 2 mm by 2 mm of the surface with a depth resolution around 3.5 mm [21]. The ratcheting method can result in the Haptic Jamming surface deforming above the desired surface height, and the noise and limited resolution of the Kinect estimation can result in a device output with errors on either side of the desired surface height.

B. Electronics and Data Acquisition

The 12-cell prototype requires 19 total actuators, including 12 valves for the vacuum lines, 6 solenoids to pin the nodes, and a flow regulator to control the air chamber pressure. In order to control all 18 of these valve and solenoid actuators simultaneously from three digital output lines of a USB X-Series 6343 DAQ (National Instruments), we use a daisy chain of 74HC595 shift registers. The output lines of the shift registers connect to the inputs of ULN2803 Darlington array transistors that power the actuators in low side drive from a 24V power supply. The flow regulator is controlled by sending a pulse width modulation output from the DAQ through another input to one of the Darlington arrays to get a 0-24V analog signal. The 0-5V analog output of the pressure sensor connects directly to an analog input of the DAQ. The camera data from the Kinect is read into Matlab using the Image Acquisition Toolbox and analyzed with the Image Processing toolbox.

C. Pressure Control

To maintain smooth pressure control inside the air chamber beneath the surface, we implemented a PID controller on the flow rate of air into the chamber with a pressure sensor attached directly to the chamber to provide feedback, as shown in the inner loop of the full system's controller in Fig. 3. This setup has two main advantages over the commercial pressure regulators used in previous prototypes, which typically implement a bang-bang (hysteresis) controller with a pair of solenoid valves based on feedback from a pressure sensor on the outlet of the regulator. First, the derivative controller smooths out the popping sensation felt by the user when a bang-bang controller regulates the pressure, because when a user presses on the surface it cause small pressure changes inside the box, requiring adjustments by the regulator. Second, the integral controller has the added benefit of negating the effects of any small leaks in the box without sacrificing smoothness in the pressure control. A commercial regulator maintains the pressure on its outlet, which does not match the pressure inside the box if there are any leaks.

IV. SHAPE SENSING AND CONTROL

Previous implementations of Haptic Jamming surfaces did not include any means for shape sensing, and thus relied on preprogrammed sequences of operations to create a desired shape. This limited both the complexity and accuracy of shapes that the device could create. Adding a sensor into the loop allows feedback of the output shape to expand the capabilities of the device.

A. Depth Sensor Registration to Surface Heights

As shown in Fig. 1, we lined the outside of the Haptic Jamming surface with a bright orange border to facilitate registration of the Kinect's RGB-D sensor to the coordinate

frame of the device. For the purposes of this initial prototype, the Kinect was assumed to be mounted parallel above the surface to simplify the three-dimensional registration into a two-dimensional problem. In each frame triggering, the Kinect provides an RGB color image and a separate depth image, both at a resolution of 480×640 pixels. Because the two cameras are not perfectly colocated, we must first transform the color image to align with the frame of the depth image. After performing this image transformation we then convert the RGB image to an HSV image and separately threshold each of the channels based on a tolerance around the values for the bright orange paper. We then open the resulting image mask with a disk structuring element to remove any noise, leaving just a mask of the orange border. We then fill the interior of this mask and subtract out the orange border, resulting in a mask that corresponds to the actual surface of the display. Finally, we calculate the orientation of the major axis of the mask.

The registration of the device location to the camera is only necessary once each time the system is powered on or either the camera or device positions are adjusted. After this initial frame, we collect only the depth data and ignore the color camera. We use the mask generated from the color image to crop the depth image to include only the surface of the device and then rotate each frame by the orientation of the major axis calculated during the registration to align the surface edges with the image edges. To obtain the heights of each point on the surface, we invert the depth image and subtract the height of the outer boundary so that it is zeroed about the edge of the surface. Knowing the actual length and width dimensions of the surface allows us to scale a three-dimensional surface plot accurately. Finally, we convolve the image with a Gaussian filter to smooth some of the noise and pixelation features that result from the limited resolution of the Kinect.

Algorithm 1 Haptic Jamming Actuation

while $pressure_{set} <$ maxPressure **do**
  $pressure_{set} \leftarrow pressure_{set} +$ increment
  allNodesArePinned $\leftarrow$ true
  for each *pinnableNode* in *array* **do**
    if $\delta_{n,i} <= 0$ **then**
      pinnableNode.fixed $\leftarrow$ true
    else
      allNodesArePinned $\leftarrow$ false
  allCellsAreJammed $\leftarrow$ true
  for each *cell* in *array* **do**
    if $\delta_{\kappa,i} <= 0$ **then**
      cell.jammed $\leftarrow$ true
    else
      allCellsAreJammed $\leftarrow$ false
  if *allNodesPinned* && *allCellsJammed* **then**
    break

B. Closed-Loop Shape Control

To close the loop on the shape control, we needed to extract a measured error for the aspects of the surface that the Haptic Jamming system can directly actuate, which include the cell curvatures and the node heights. The height measurements collected by the RGB-D sensor provide a direct estimate of each node height, $n_{est,i}$, by averaging the nine pixels surrounding the location of each pinnable node, so we can calculate a measured error of each node height, $\delta_{n,i}$, as

$$\delta_{n,i} = n_{des,i} - n_{est,i}, \tag{1}$$

where $n_{des,i}$ is the height of the corresponding node in the desired input surface. Similarly, the measured error of each cell curvature, $\delta_{\kappa,i}$, can be found by subtracting the measured curvature of each cell, $\kappa_{est,i}$, from the curvature, $\kappa_{des,i}$ of the corresponding cell in the desired input:

$$\delta_{\kappa,i} = \kappa_{des,i} - \kappa_{est,i}. \tag{2}$$

However, unlike the node heights, the cell curvatures do not come directly from the surface height data. We estimate the curvature of a cell by calculating the difference between the height of the center of the cell and the heights of the midpoints of the diagonals that connect opposite corners of the cell, where positive curvature denotes ballooning outward.

As shown in the block diagram of the system control architecture in Fig. 3, these measured errors $\delta_{n,i}$ and $\delta_{\kappa,i}$ provide the necessary inputs for Algorithm 1 to actuate the device via cell jamming, node pinning, and adjusting the pressure in the air chamber beneath the surface. This algorithm iteratively increases the pressure in the chamber by a set increment of 69 Pa (0.01 psi), pins any node that has reached the height of the corresponding node in the desired surface, and jams any cell that has reached the curvature of the corresponding cell in the desired input surface.

When a cell is jammed and its mechanical properties change from soft to rigid, it locks that region of the surface at its current curvature so that it does not deform with further adjustments of the air chamber pressure. Curvature and height are independent features because an unpinned node's height can continue to rise with increasing chamber pressure regardless of whether the cells around it are jammed, and a cell's curvature can continue to increase with increasing chamber pressure regardless of whether the nodes around it are pinned. However, due to the continuity of the surface, the curvatures of neighboring cells are dependent on each other, and the heights of neighboring nodes are dependent on each other. Surface stiffness is not entirely independent of shape control. Cells that remain soft throughout a shape configuration can have their rigidity controlled independently by jamming afterwards, but cells that are jammed in the process cannot be unjammed afterwards without sacrificing the integrity of the shape.

The controller turns off when either the chamber reaches a maximum pressure of 2.8 kPa (0.4 psi) or all of the nodes and cells are pinned and jammed, whichever occurs first. Each

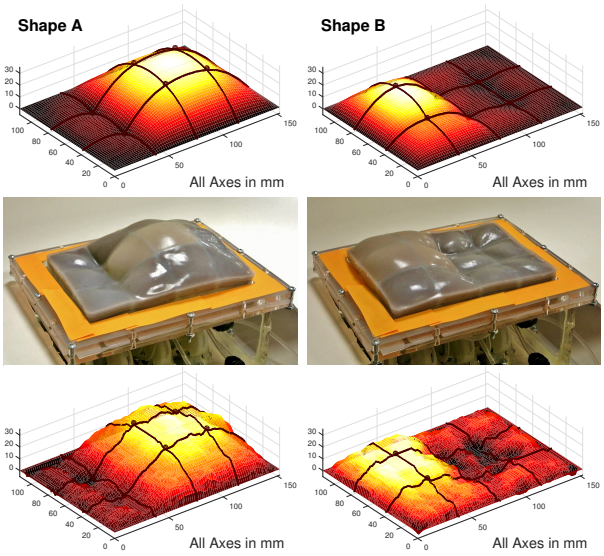


Fig. 4. Two of the desired input shapes (top) used to evaluate the system, their corresponding outputs on a 12-cell array (middle), and their surface shape measurements from the depth camera (bottom).

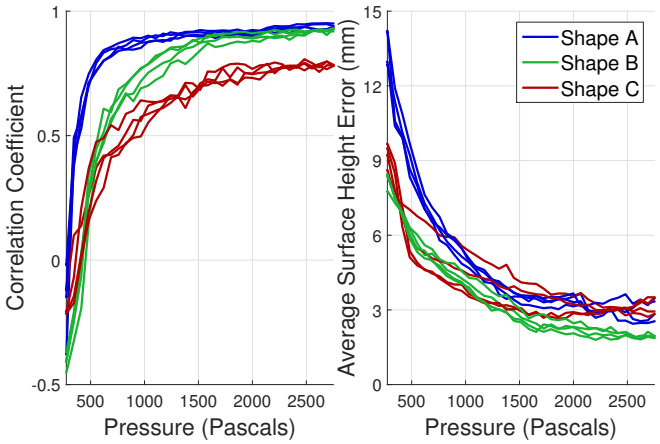


Fig. 5. As the shape control algorithm incrementally increases chamber pressure and jams cells or pins nodes when they reach their respective desired curvature or height, the correlation between the desired input shape and the output shape increases and the average height error across the surface decreases.

cycle through the outer loop of the control block diagram includes one iteration of Algorithm 1 and one measurement of the surface heights with the depth sensor. In the current setup, this outer loop runs at about 0.5 Hertz due to the processing time required for the Kinect images in Matlab. The inner control loop for the air chamber pressure control runs continuously at about 20 Hertz, even when the outer loop is not running because the air pressure is necessary to maintain the shape of any cells that remain unjammed.

V. EVALUATION

We took both quantitative and qualitative approaches to evaluate the performance of the control techniques described in Section IV. To quantitatively evaluate the performance we needed to generate desired input shapes that the 12-cell Haptic Jamming surface could feasibly re-create, whereas the qualitative evaluation permitted more freedom in the selection of a desired surface shape.

A. Accuracy of Generated Shapes

A deformable body simulation of a Haptic Jamming surface [22] generated sample input shapes that the device could feasibly re-create accurately. This simulated surface consists of an array of point masses connected in a grid pattern by normal and torsional springs, with the outer boundary of point masses fixed in space. By applying outward forces normal to the surface at each point mass, the body deforms in such way that simulates pressurizing the air chamber beneath the surface. To simulate pinning a node, the point mass at that node's location is simply fixed in space. Increasing the stiffness of the torsional and normal springs within a cell inhibits that cell's deformation to simulate jamming. Various sequences of cell jamming, node pinning, and chamber pressurization created the three output shapes we used to quantitatively evaluate the control of the device's shape output. The first two shapes, Shape A and Shape B, are shown in Fig. 4 with their corresponding outputs on the device, and Shape C appears in Figs. 1 and 3.

The simulator output consists of a matrix of heights of point masses, which we resized using bicubic interpolation to match the resolution of the Kinect depth sensor's point cloud of the actual surface. This allows direct comparison between the desired and actual shapes using two quantitative metrics. First, Pearson's r correlation coefficient, commonly used for template matching in images [23], is calculated as

$$r = \frac{\sum_i \sum_j \left(A_{ij} - \overline{A}\right)\left(B_{ij} - \overline{B}\right)}{\sqrt{\left(\sum_i \sum_j \left(A_{ij} - \overline{A}\right)^2\right)\left(\sum_i \sum_j \left(B_{ij} - \overline{B}\right)^2\right)}}, \quad (3)$$

where A_{ij} and B_{ij} are the heights of each point in the two surfaces and $\overline{A}$ and $\overline{B}$ are the mean heights of each image. A correlation coefficient of $r = 1$ indicates an exact match, whereas $r = -1$ indicates an exact inversion, and $r = 0$ indicates no correlation. The mean of the absolute values of the height difference between each corresponding point of the two surfaces yields the second evaluation metric. A high correlation coefficient and low average height error indicate a good match in slightly different ways, as two surfaces with similar curvatures but different height scales or positions would have a strong correlation coefficient but also a large average height error.

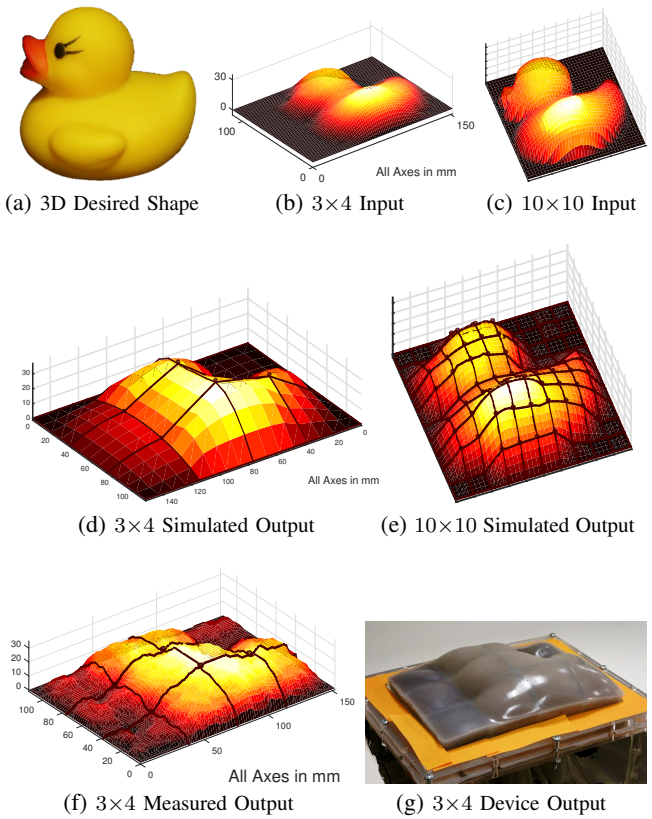
We ran the closed-loop controller on each of the three sample surfaces four times with a final pressure of 2.8 kPa (0.4 psi). We calculated the correlation coefficients and average height errors across the surface for each iteration through the outer loop of the controller and Fig. 5 shows plots of these values for all trials. As we would expect, correlation increases and error decreases as the controller progresses, although not entirely monotonically due in large part to the noise in the Kinect depth sensor. The video that accompanies the submission shows one of the four trials for each of the three sample shapes. The correlation coefficients

level off at mean values of 0.94, 0.93, and 0.78 for Shapes A, B, and C, respectively. The heights within each desired surface had ranges of 32.3 mm, 23.1 mm, and 22.4 mm, with corresponding average height errors of 3.1 mm, 1.9 mm, and 3.2 mm from the output surface.

B. Re-creating 3D Objects

For a more qualitative evaluation of the controller's performance and to explore the capabilities of a Haptic Jamming surface for various applications, we tested the system with a desired shape input generated from a real 3D object. The duck chosen as a sample object for this work appears in Fig. 6a. The output capabilities of the Haptic Jamming surface are not truly three-dimensional in the sense that the surface cannot display the underside of objects to the user or render any shapes with overhanging features, an effect we describe as 2.5D. To account for these limitations when generating the desired shape, we first z-buffer a 3D model of the object and, zero the heights to the lowest visible point, and resize the resulting image, again with bicubic interpolation, to the size of the Haptic Jamming array. The corresponding desired input shapes for a 3×4-cell prototype and a 10×10-cell surface are shown in Figs. 6b and 6c, respectively.

Running the closed-loop controller on the 3×4-cell prototype with the desired input shape in Fig. 6b results in the output shape shown in Fig. 6f with the corresponding measured surface shown in Fig. 6d. This output shape highlights further limitations of the Haptic Jamming surface beyond the 2.5D constraints. First, the surface cannot sufficiently recreate sharp peaks, particularly when located near the edge of the array. Furthermore, sharp valleys are limited to the locations of the nodes between the cells that can be pinned at lower heights. As a result, the shape output by the 3×4-cell surface does not very closely resemble the desired input shape, let alone the 3D object from which it was generated. We can use the same spring-mass deformable body simulator that we used to generate the desired input shapes in Section V-A to test the feasibility of using larger arrays to create more compelling shapes. The output of the simulated larger array in Fig. 6g shows much closer resemblance to the input shape and suggests that increasing the array size and resolution can reduce many of the limitations of the Haptic Jamming surface.



(a) 3D Desired Shape (b) 3×4 Input (c) 10×10 Input

(d) 3×4 Simulated Output (e) 10×10 Simulated Output

(f) 3×4 Measured Output (g) 3×4 Device Output

Fig. 6. A Haptic Jamming surface can render various 3D objects, like a duck (a), by z-buffering an image to convert to a desired shape input for a 12-cell array (b) or 100-cell array (c). While the simulated output shape (d) and actual output from a 3×4-cell array prototype (g) results in limited output correlation (f) to the input, a simulation of a 10×10-cell array promises more compelling outputs from larger arrays (e).

VI. CONCLUSION AND FUTURE WORK

This work demonstrated the first closed-loop shape control of a Haptic Jamming surface, using a 3×4-cell prototype with the Kinect depth sensor for feedback. This initial controller implements an algorithm that incrementally increases the chamber pressure beneath the surfaces in each control loop while pinning nodes at their desired heights and jamming cells at their desired curvatures. The main drawback of the current implementation stems from the limited loop rate of the controller. While this slow loop rate does not affect stability due to the incremental nature of Algorithm 1, it does limit many applications of a Haptic Jamming surface. The surfaces tested in this work had an average height error of 2.7 mm across all trials, which is within both the resolution of the Kinect's depth sensor as well as the resolution of the ratchet system on the node-pinning mechanism.

We will explore increasing the resolution of the node pinning mechanism by switching from the ratchet-based design to a method that would allow pinning a node at any height along a continuous range. Future implementations will work towards developing a system that can process a desired input shape, for example a real object placed near the device where the depth sensor can measure its surface topology as well as the device's, and recreate it in real-time on the Haptic Jamming surface. This will first require moving the depth sensor acquisition and image processing from Matlab to a more real-time processing software, and transferring the inner control loop and actuation control from a computer controlled DAQ to a microcontroller with intermittent commands received serially from the computer. Furthermore, since adjusting the chamber pressure represents the limiting factor in terms of the dynamic response of the physical system, we plan to develop more model-based control algorithms that make larger chamber pressure adjustments before using the closed-loop shape feedback to make the finer adjustments, thus limiting the number of pressure changes per shape reconfiguration. This may also facilitate

transferring directly from one shape to another without requiring returning to a flat, depressurized, and unjammed state as an intermediate step.

To validate that a larger array can create more compelling output shapes as suggested by the simulation output in Fig. 6g, we are constructing a 100-cell array. This will not only serve as a test bed for the more advanced control algorithms we plan to develop but also for a variety of interaction features, expanding those explored with the pin array of the inFORM interface like object manipulation and telepresence [16] to similar features on the continuous, deformable surface of a Haptic Jamming interface. In addition, we plan to embed flexible capacitive sensors into the silicone layers of the surface, which could simultaneously provide shape feedback for the controller without the concerns of occlusion from user interaction as well as touch or proximity feedback to enable a greater variety of user interaction features.

REFERENCES

[1] A. A. Stanley, J. C. Gwilliam, and A. M. Okamura, "Haptic jamming: A deformable geometry, variable stiffness tactile display using pneumatics and particle jamming," in *IEEE World Haptics Conference*, pp. 25–30, 2013.

[2] A. A. Stanley and A. M. Okamura, "Controllable surface haptics via particle jamming and pneumatics," *IEEE Transactions on Haptics*, vol. 8, no. 1, pp. 20–30, 2015.

[3] D. Trivedi, C. D. Rahn, W. M. Kier, and I. D. Walker, "Soft robotics: Biological inspiration, state of the art, and future research," *Applied Bionics and Biomechanics*, vol. 5, no. 3, pp. 99–117, 2008.

[4] E. Steltz, A. Mozeika, N. Rodenberg, E. Brown, and H. Jaeger, "JSEL: Jamming Skin Enabled Locomotion," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5672–5677, 2009.

[5] N. G. Cheng, M. B. Lobovsky, S. J. Keating, A. M. Setapen, K. I. Gero, A. E. Hosoi, and K. D. Iagnemma, "Design and Analysis of a Robust, Low-cost, Highly Articulated manipulator enabled by jamming of granular media," in *IEEE International Conference on Robotics and Automation*, pp. 4328–4333, 2012.

[6] E. Brown, N. Rodenberg, J. Amend, A. Mozeika, E. Steltz, M. R. Zakin, H. Lipson, and H. M. Jaeger, "Universal robotic gripper based on the jamming of granular material," *Proceedings of the National Academy of Sciences*, vol. 107, no. 44, pp. 18809–18814, 2010.

[7] A. Loeve and O. van de Ven, "Vacuum packed particles as flexible endoscope guides with controllable rigidity," *Granular Matter*, vol. 12, pp. 543–554, 2010.

[8] S. Follmer, D. Leithinger, A. Olwal, N. Cheng, and H. Ishii, "Jamming User Interfaces: Programmable Particle Stiffness and Sensing for Malleable and Shape-Changing Devices," in *ACM Symposium on User Interface Software and Technology*, pp. 519–528, 2012.

[9] A. Mazzone, C. Spagno, and A. Kunz, "The HoverMesh: A deformable structure based on vacuum cells," in *Proc. ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, pp. 187–193, 2004.

[10] N. Aihara, T. Sato, and H. Koike, "Highly deformable interactive 3D surface display," in *ACM Symposium on User Interface Software and Technology*, p. 91, 2012.

[11] T. Mitsuda, S. Kuge, M. Wakabayashi, and S. Kawamura, "Wearable Force Display Using a Particle Mechanical Constraint," *Presence: Teleoperators and Virtual Environments*, vol. 11, no. 6, pp. 569–577, 2002.

[12] J. Rossignac, M. Allen, W. Book, A. Glezer, I. Ebert-Uphoff, C. Shaw, D. Rosen, S. Askins, P. Bosscher, J. Gargus, I. Llamas, and A. Nguyen, "Finger sculpting with Digital Clay: 3D shape input and output through a computer-controlled real surface," in *Shape Modeling International*, pp. 229–231, 2003.

[13] P. Taylor, A. Moser, and A. Creed, "A sixty-four element tactile display using shape memory alloy wires," *Displays*, vol. 18, pp. 163–168, 1998.

[14] C.-H. King, M. O. Culjat, M. L. Franco, J. W. Bisley, E. Dutson, and W. S. Grundfest, "Optimization of a pneumatic balloon tactile display for robot-assisted surgery based on human perception.," *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 11, pp. 2593–600, 2008.

[15] H. Iwata, H. Yano, and N. Ono, "Volflex," in *ACM SIGGRAPH 2005 Emerging technologies*, p. 31, 2005.

[16] S. Follmer, D. Leithinger, A. Olwal, A. Hogge, and H. Ishii, "inFORM: Dynamic physical affordances and constraints through shape and object actuation," in *User Interface Software and Technology*, pp. 417–426, 2013.

[17] H. Zhu and W. J. Book, "Control Concepts For Digital Clay," in *IFAC Symposium on Robot Control*, 2003.

[18] R. Winck, J. Kim, W. J. Book, and H. Park, "Command generation techniques for a pin array using the SVD and the SNMF," in *10th IFAC Symposium on Robot Control*, vol. 10, pp. 411–416, 2012.

[19] A. Mazzone and A. Kunz, "Sketching the future of the SmartMesh wide area haptic feedback device by introducing the controlling concept for such a deformable multi-loop mechanism," *Links*, vol. 3, p. 248, 2005.

[20] S. Klare and A. Peer, "The Formable Object: a 24-degree-of-freedom shape-rendering interface," *IEEE/ASME Transactions on Mechatronics*, no. 99, pp. 1–12, 2014.

[21] K. Khoshelham and S. O. Elberink, "Accuracy and resolution of kinect depth data for indoor mapping applications," *Sensors*, vol. 12, no. 2, pp. 1437–1454, 2012.

[22] A. Stanley and A. Okamura, "Deformable model-based methods for shape control of a haptic jamming surface," *IEEE Transactions on Visualization and Computer Graphics*, p. Accepted, 2016.

[23] J. P. Lewis, "Template matching using fast normalized cross correlation," in *Vision Interface*, vol. 95, pp. 15–19, 1995.