A Gesture-free Geometric Approach for Mid-Air Expression of Design Intent in 3D Virtual Pottery

Vinayak[a,*], Karthik Ramani[a,b]

[a]School of Mechanical Engineering, Purdue University, West Lafayette, Indiana 47907
[b]School of Electrical Engineering (by courtesy), Purdue University, West Lafayette, Indiana 47907

Abstract

The advent of depth cameras has enabled mid-air interactions for shape modeling with bare hands. Typically, these interactions employ a finite set of pre-defined hand gestures to allow users to specify modeling operations in virtual space. However, human interactions in real world shaping processes (such as pottery or sculpting) are complex, iterative, and continuous. In this paper, we show that the expression of user intent in shaping processes can be derived from the geometry of contact between the hand and the manipulated object. Specifically, we describe the design and evaluation of a geometric interaction technique for bare-hand mid-air virtual pottery. We model the shaping of a pot as a gradual and progressive convergence of the pot's profile to the shape of the user's hand represented as a point-cloud (PCL). Thus, a user does not need to learn, know, or remember any gestures to interact with our system. Our choice of pottery simplifies the geometric representation, allowing us to systematically study how users use their hands and fingers to express the intent of deformation during a shaping process. Our evaluations demonstrate that it is possible to enable users to express their intent for shape deformation without the need for a fixed set of gestures for clutching and deforming a shape.

Keywords: hand-based shape modeling, mid-air interactions, virtual pottery, gestures, natural user interfaces, mesh deformation.

1. Introduction

Mid-air interactions have received significant attention in the context of 3D shape conceptualization. These interaction approaches offer intuitive and direct manipulation of virtual shapes by using free hand movements. The core challenge in mid-air interactions for computer-aided design (CAD) is to map user input to virtual operations, such that the user can directly focus on the design task rather than spending time in learning the tool itself. Three-dimensional user input has been extensively studied, evaluated, and reviewed [1, 2, 3, 4] in the context of 3D selection, manipulation [5], control, and navigation, in virtual environments. Mid-air interactions have found significant use in gaming [6] and art [7, 8, 9]. Within the context of 3D conceptual shape design, we find two broad classes of mechanisms that enable mid-air user input.

The first class comprises of instrumented controllers such as gloves [10, 11, 12], hand-held trackers [13, 14], and haptics devices [15, 16, 17]. Special devices and setups [18, 19, 20, 21, 22, 23] have also been demonstrated for 3D interactions. These hardware systems offer great control, feedback, and unambiguity to the user while interacting in mid-air. In these approaches, the user provides explicit commands via the hand-worn or hand-held controller to indicate design intent such as starting, stopping, or selecting a modeling operation when desired. However, such systems are not accessible to the common user outside a lab environment. Further, wearing or holding can be intrusive to the user during a focused modeling task.

The second class of mechanisms for mid-air user input are the so-called *bare-hand* gesture-based interactions [24, 25, 26, 27, 28, 29, 30, 31]. With the recent commercialization of depth cameras, gesture-based interactions have become accessible to the common user. Creative applications for free-form shape modeling [32] in mid-air have gained significant popularity. The user input in these applications is represented as a combina-

---

*corresponding author
Email addresses: fvinayak@purdue.edu (Vinayak),
ramani@purdue.edu (Karthik Ramani)

tion of a hand posture (such as pointing with a finger) and the motion of a representative point (such as the palm or fingertip) on the hand. We call this the *symbolic gesture* approach. Here, converting user input into a meaningful shape modeling task involves (a) acquisition, segmentation, and processing of hand data, (b) extracting a virtual representation of the hand, (c) mapping the gestures to a shape modeling task, and (d) generating an appropriate response of the shape as intended by the user's input. Such approaches involve estimation of the skeletal structure of the hand [33, 34, 35] or classification of the hand's image as a pre-defined posture [36]. In this work, we aim to utilize the non-intrusiveness and accessibility of camera-based mid-air interactions for 3D shape deformation.

Both classes of mid-air interactions (instrumented controllers and bare-hand gestures) have one common characteristic, namely, there is a clear distinction between interaction and geometric modeling. In this work, we take a different approach; we pose mid-air interaction itself as a geometric problem. Our focus is the geometric investigation of spatial interactions in the specific context of shape deformation. Particularly, we address the problem of determining how the shape and motion of a user's hand and fingers geometrically relates to the user's intent of deforming a shape.

1.1. Motivation

Consider a mid-air interaction scenario of selecting and displacing a mesh vertex for deforming the mesh. Unlike a controller-based approach, there is no explicit physical mechanism for triggering events. Here, gestures serve two fundamental purposes. First, they help define a beginning (e.g. reaching and clutching some region of interest) and end (e.g. de-clutching the region after required deformation) of an interaction [37, 38]. Secondly, they help define the exact operation from a set of operations defined in the context of an application. For example, the type of deformation could be selected by using different gestures (e.g. fist to pull, point to push, open palm to flatten).

There are two issues with this approach. First, gesture-based interactions rely heavily on (a) the robustness of gesture recognition to hand occlusions and (b) stability of recognized gestures over time. In our previous work [30], we observed that even slight instability in the accuracy of gesture recognition can disrupt the modeling process leading to user frustration. Secondly, predefined gestures impose rigid constraints on users. Having to learn too many gestures is memory-intensive in a creative activity, but too few gestures can result in ambiguity while expressing design intent.

Hand and finger movements in real-world shaping processes (such as pottery or clay sculpting) are complex, iterative, and gradual. Such processes are essentially governed by the geometry of contact between hand and clay. Works by Sheng et al. [19], Kry et al. [20], and Pihuit et al. [21] have leveraged finer finger level movements and grasping for 3D shape deformation by designing ingenious hardware systems. In their systems, users can actually grasp and deform virtual objects. However, for marker-less camera-based systems, the true expressive potential of finger movements remains underutilized despite advances in hand pose and skeleton estimation. We aim to determine user's intent from fine finger-level movements while retaining the non-intrusiveness and accessibility of depth cameras. In doing so, we demonstrate that it is unnecessary to *prescribe* shape modeling interactions with a predefined set of rules using classified hand gestures or full-scale hand skeletons.

1.2. Goals

There are two factors that demand critical attention while designing mid-air interactions, particularly for shape modeling. These are **intent** and **controllability**. The general term *intent* is literally defined as "*the thing that you plan to do or achieve : an aim or purpose*". In our case, intent (*what one wants to achieve*) can be described in terms of the context of shape deformation (*what operations one can perform on the shape*). On the other hand, controllability can be viewed as the quality of intent recognition and disambiguation *as perceived by the user*. In general, we find that the prior works in mid-air shape modeling primarily focus on the development of a feature-rich geometric modeling system. However, there is currently no literature that studies problems of intent and controllability in mid-air interactions for shaping operations such as deformation. Our broader aim in this paper is to study these critical aspects of mid-air shape deformation.

Considering intent and controllability as our central themes, we have two specific goals in this paper. First, we seek a concrete geometric method that takes a general representation of the user's hand (PCL) and allows the user to deform 3D geometry. Secondly, we want to investigate this geometric method in light of intent and controllability. Thus, our focus here is not to build a comprehensive and feature-rich 3D modeling system. Instead, we intend to investigate spatial interactions for 3D shape deformation with a raw representation of the hand.

Considering a full-fledged 3D modeling system makes it prohibitively complicated to rigorously study

the aspects of the interaction. There are two basic components to our problem - the representation of the deforming shape and that of the user's hand (PCL in our case). In a typical shape deformation scenario, an arbitrary triangle mesh is the ideal and generic shape representation. The use of an arbitrary mesh makes it prohibitively challenging to perform a controlled study of the geometry of interactions. There are two reasons for this. First, the hand PCL data obtained from a single depth sensor is partial (due to occlusions) and noisy. Secondly, we want the user to approach a 3D shape from any direction and grasp it in any way. In this case, the user's dynamic and complex finger motions further add complexity to the occlusions and noise. Subsequently, designing interaction tasks for a quantitative evaluation is difficult, particularly for users with no prior experience with mid-air interactions for free-form 3D modeling. Hence, it is essential to constrain the geometric representation of the object being modified.

To cater to our goal of investigating mid-air deformation, we chose the example of virtual pottery. Our choice of pottery offers a well-defined and intuitive relationship between the use of hands and the shaping of pots to a user. The simplicity of the geometric representation and deformation in pottery lends itself to quantitative measurement of the user's response to our system (Section 6.3). Our concrete and practical implementation enabled us to study how hand and finger motions can allow users to deform a shape *as they see fit* (Section 6.4.3). We note that virtual pottery using direct hand manipulations is a niche area within the hand-based modeling paradigm [10, 17, 22, 39, 25, 28, 29]. We do not aim at contributing towards pottery itself, but using pottery to reveal the research questions pertaining to the geometry of spatial interactions.

1.3. Approach

Our basic idea is to progressively conform the shape of a pot (represented as a 3D mesh) to that of the user's hand. This idea, dubbed *proximal persistence*, is a generalization of the notion of *dwell-time* used in 3D object selection [40]. Our algorithm is a combination of *exponential smoothing* and selective *Laplacian smoothing* (Figure 1). Here, each point in the hand's PCL *attracts* a local region on the pot, hence deforming the pot's section. The combination of many such local deformations, due to each point on the hand, amounts to a progressive convergence of the pot-profile to the shape of the user's hands. We found two approaches that appear similar to our method in terms of the representation of the hand and the idea of persistence-based deformation.

The first approach is *Data miming* [27] wherein the hand is modeled as a set of active voxels in a volumetric domain. Thus, this approach uses hands without explicit determination of gestures for recognizing user intent, as in our case. Our method, however, is inherently different from this approach in two ways. Firstly, *Data miming* focuses on the specification existing objects through the user's hand motions rather than creating new shapes through shape deformation. Secondly, it differentiates the relevant voxels from the irrelevant ones by imposing a threshold on the number of times the user's hand visit each voxel. Our method does not require any threshold or statistic to distinguish intentional actions from unintentional ones. Instead, we use the rate of attraction as our parameter to determine when and how much to deform a surface (Section 2.3). We also provide an explicit relationship between our parameters and the responsiveness of shape deformation (Section 3). We determined appropriate parameter values of our algorithm through our pilot experiments. Such a study was not the focus of the *Data miming* approach.

The *magnet tool* demonstrated by Schkolne et al. [11] captures the idea of persistence. It is a custom made hardware addendum that allows a user to change a region of surface locally by *waving the hand close to the surface while holding the tool*. This tool, as mentioned by the authors, enables an overdrawing metaphor for surface deformation. In our case, the input is dynamic and unstructured data (the PCL of the user's hand) and there is no explicit user expression (such as waving) that triggers the deformation of a shape. Further, our method is capable of both local and global changes to the shape depending on how the user makes contact with the surface of the shape.

1.4. Contributions

We make two main contributions. First we demonstrate, with a practical implementation, that it is possible to achieve controllability in bare-hand mid-air shape deformation using unstructured PCL data of the user's hand. We describe a method that does not compute any finite set of gestures or hand skeleton. Instead, our method uses the actual shape of the user's hands for deforming the shape of a pot in 3D space. This directly allows a user to shape pots by using physical objects as tools.

Secondly, we show a detailed user evaluation to understand user behavior and perception in pottery design. Our evaluations help reveal two core aspects of mid-air interactions for shape deformation, namely, intent & controllability. We characterize user behavior in pottery
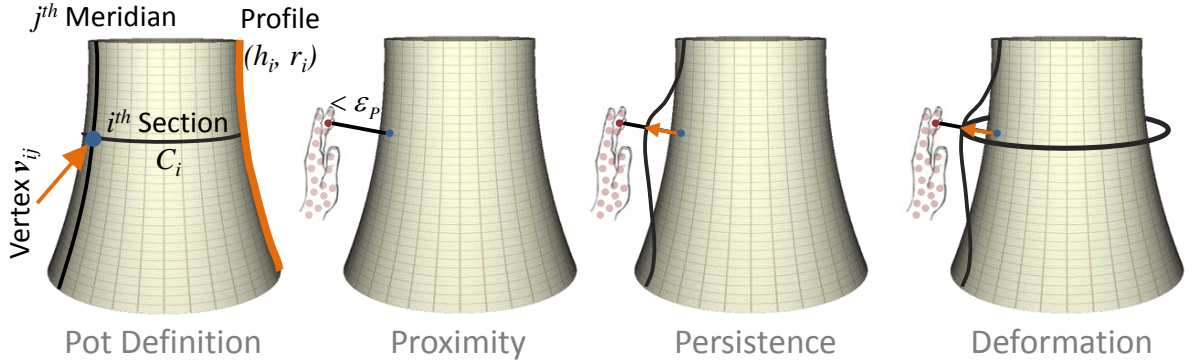
Figure 1: Definition of a pot and the deformation using proximal persistence.

design in terms of (a) common hand & finger movement patterns for creating common geometric features, (b) user perception of intent, and (c) engagement, utility, and ease of learning provided by our approach.

2. Technical Details

Proximal persistence can be described as a spatio-temporal model according to which a 'modifiable object' in space deforms when in close proximity to a 'modifying object'. Deformation occurs over a length of time, until the shape of the modifiable object matches the profile of the modifying object. In our case, the modifiable object is the surface of a virtual pot, the modifying operation is deformation, and the modifying object is a point-sampled surface of a hand or a hand-held tool. Thus, the idea is to define the modeling process of a 3D shape as a time-series of its deformed states converging, so as to conform to the shape of the hand or the tool (Figure 1).

To provide a realistic and enjoyable experience of pottery, there are four considerations which drive our modeling technique. These are: (a) smooth appearance, (b) behavioral realism, (c) intuitive interactions, and (d) the possibility of using real-world and virtual objects as tools for pot creation. We will first define the required components of the technique and describe a general strategy for the deformation of pots. Our strategy is based on Laplacian smoothing which has been extensively used in literature (see [41] for a detailed review).

2.1. Pot Definition

We describe a pot $\tilde{P}(h, r(h))$, as a simple homogeneous generalized cylinder (SHGC) containing a set of circular sections at heights $\tilde{h} \in [h_1, h_2]$ ($h_1, h_2 \in \mathbb{R}$) whose radii are defined by a smooth function $\tilde{r} : \mathbb{R} \to \mathbb{R}$

applied to the closed and connected interval $[h_1, h_2] \in \mathbb{R}$. Thus, the function $\tilde{r}(\tilde{h})$ can also be interpreted as the *profile curve* of the pot. In a discrete setting, a pot is given by:

$$P_{n,m} := \{(h_i, r_i) \mid r_i = r(h_i) , \ h_1 < \ldots < h_n \in \mathbb{R}\} \quad (1)$$

The surface $S(P_{n,m})$ of the pot is defined as a vertical sequence of n circular sections C_i of radii r_i at heights h_i. In a discrete setting, a circular section is approximated by a closed regular polygon with m sides (2) thus allowing for a simple quad-mesh representation.

$$C_i = [\mathbf{v}_{i,1}, \ldots, \mathbf{v}_{i,m}]^T \quad (2)$$

$$\mathbf{v}_{i,j} = \left[-r_i \sin(\frac{2j\pi}{n}), h_i, -r_i \cos(\frac{2i\pi}{n})\right]^T$$

A *manipulator* is defined as a point denoted by $\mu(t, \mathbf{p})$, where t is the time stamp and $\mathbf{p} \in \mathbb{R}^3$ is the position of the manipulator. The manipulating object (user's hand or a hand-held tool)can thus be represented as a point cloud (PCL) given by $M(t) = \{\mu(t, \mathbf{p})\}$ where each point is a manipulator. A *handle*, $\mathbf{q}(t) = \mathbf{v}_{i,j} \in V$ ($1 \le i \le n$ and $1 \le j \le m$), is defined as the vertex on the pot which is closest to the manipulator in euclidean space. We define a *convergence threshold* ε_C as the distance between a handle and its manipulator at which the handle is considered to be coincident with the manipulator. Finally, we define a *proximity threshold* as the maximum distance, ε_P, at which a manipulator should be from a handle in order for a deformation to occur. Making use of the structured topology of the pot mesh, we compute the handle for a given hand by first computing the index of nearest section and subsequently the point closest to the hand on the nearest section. Thus, the handle $\mathbf{q}$ at an instance t, is given by:

4

$$\mathbf{q}(t) = \mathbf{v}_{i,j} \tag{3}$$
$$i = \underset{i}{\operatorname{argmin}}(h_i - p_y(t))$$
$$j = \underset{j}{\operatorname{argmin}}(\|\mathbf{v}_{i,j} - \mathbf{p}(t)\|)$$

Here, $\|.\|$ is the Euclidian distance defined in $\mathbb{R}^3$. Further, i gives the index of the cross-section closest to the manipulator. It is clear that the relationship between a manipulator and its corresponding handle is that of *proximity*, the first component of *proximal persistence*.

2.2. Deformation Strategy

In real world pottery, deformations due to the potter's hands are either local or global based on the extent of the profile region in contact with the potter's hands or the tools held in the hands. The smoothness of deformation is an essential constraint as well. Consider the hand as a set of points in 3D space. Each point in the PCL deformed a small local region on the pot using the proximal persistence approach. On the whole this amounts to a gradual and progressive convergence of the pot-profile to the shape of the user's hand. We achieve this convergence by using the idea of selective Laplacian smoothing. This method performs local or global deformation of the profile based on the distribution of the manipulators on the profile. For manipulators within a small region of the profile, the deformation is local. If the manipulators cover a broader span of the profile, the deformation is global.

Given the profile function $r(h)$, the problem of deforming the surface of a pot is essentially transformed to that of deforming the profile curve. For a curve $\tilde{\gamma}(u) \in \mathbb{R}^3$ ($u \in [0, 1]$), deformation on $\tilde{\gamma}$ can be defined as a curve $\tilde{\delta}(u)$ such that $\tilde{\delta} : \tilde{\gamma} \to (\tilde{\gamma} + \tilde{\delta})(u)$. Our goal is to determine a deformation $\tilde{\delta}(u)$ which satisfies Laplace equation ($\nabla^2 \tilde{\delta} = 0$).

With seed displacements defined on selected vertices of the profile, we compute a smooth deformation and apply it to the profile (see figure 2). Let $\gamma := \{(h_i, r_i)\}$ be the discrete profile curve as given in (1) and $K \subset [1, n]$ define a set of selected vertices in γ. Given a set of seed displacements on the vertices in K, our goal is to determine the displacement of all other vertices so as to preserve the smoothness of γ. To achieve this, we first initialize a seed displacement curve δ (such that $\delta_i = 0 \forall i \notin K$) and iteratively apply discrete Laplacian smoothing, $\mathcal{L}(\delta) = 0$, (4) with Neumann boundary conditions. The deformed profile is then obtained by setting $r_i \leftarrow r_i + \delta_i, \quad \forall i$. The derivatives at the boundary are
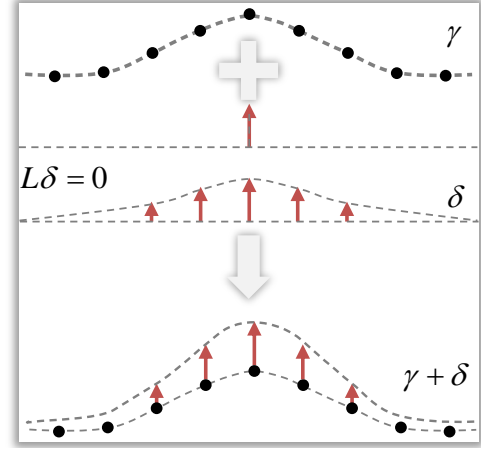

Figure 2: Deformation strategy of a curve γ (row 1) involves initializing a seed displacement (row 2), application of the Laplace operator to obtain a smooth deformation curve δ (row 3), and finally the computation of deformed curve $\gamma + \delta$.
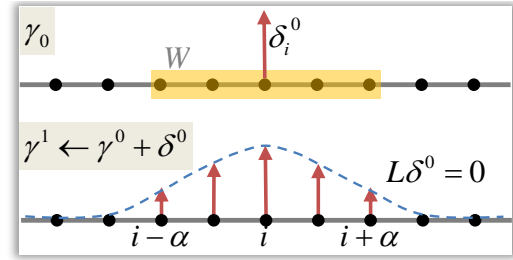

Figure 3: The selective application of Laplace operator is based on the window operator α which defines the neighborhood around some given vertex i in a profile.
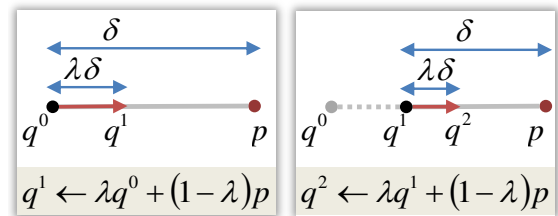

Figure 4: Recursive attraction of a handle q towards a manipulator p is shown. The parameter $\lambda \in [0, 1]$ defines the rate of convergence and is akin to the *smoothing constant* in single exponential smoothing.

set to zero, i.e, the radii at the boundary sections of the pot are the same as the radii at their neighboring sections. We use inverse edge weights to determine the non-diagonal elements of $\mathcal{L}$ (5).

$$\mathcal{L}_{i,j} = \begin{cases} -1, & |i-j| = 0 \\ 1, & |i-j| = 0, \quad i \in \{1, n\} \\ w_{i,j}, & |i-j| = 1, \quad i \notin \{1, n\} \\ 0, & otherwise \end{cases} \quad (4)$$

$$w_{i,j} = \frac{l_j}{\displaystyle\sum_{k \in K} l_k} \quad (5)$$

$$K = \{k : |i-k| = 1\}$$

In order to deform the profile in a smooth manner, we begin by defining W to be a set of all *contiguous* subsets in $[1, n]$. Note that a single section $i \in [1, n]$ and the whole set $[1, n]$ trivially belong to W. Subsequently, we define a mapping $\alpha : W \to W$ as given in (6). Given a section i, it is clear that $\alpha(i)$ determines the window of sections above and below the section i, which are subject to Laplacian smoothing (Figure 3).

$$\alpha : \{i \dots j\} \mapsto \{max(1, i-\alpha) \dots min(n, j+\alpha)\} \quad (6)$$

$$\alpha \in \mathbb{N}$$

$$i, j \in [1, n]$$

Thus, for a given seed displacement $\delta(t)$, the deformation of the profile can be controlled using parameters, namely the window operator α and a parameter $\beta \in \mathbb{N}$ specifying the number of iterations for Laplacian smoothing. Note that a single point manipulator will instantiate a smooth but local deformation as should be expected in a typical real-world scenario - *a thin tool creates a thin impression*. However, we aim to allow for both global and local deformations. In our case, note that a manipulating object is a point-sampled surface. Thus, global deformations naturally occur due to the combination of window operators for spatially proximal manipulators.

2.3. Persistence

We define persistence as a continuous temporal response of the pot profile to the proximity of a given manipulator. The main step towards the definition of persistence is one involving initializing the seed displacement. For a discrete pot profile curve γ and a handle i, the idea is to *attract* a handle towards its corresponding manipulator by a fraction of distance between them
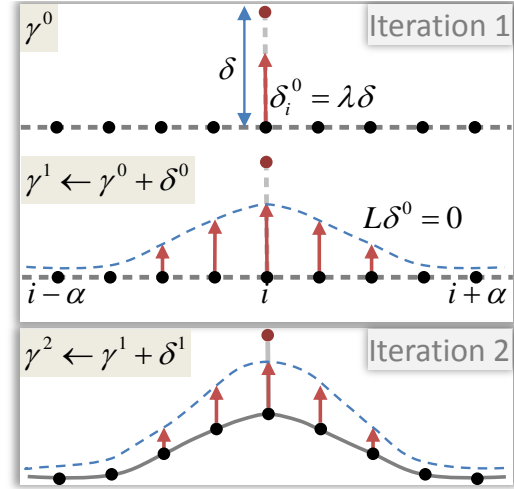
Figure 5: Algorithm for the application of proximal persistence to a discrete profile curve of the pot for one manipulator.

(Figure 4). Our idea is inspired by exponential smoothing [42]. Given a parameter $\lambda \in [0, 1]$, we converge the handle towards the manipulator over time in a smooth manner (Figure 5). Note that this parameter, when constant, is analogous to the *smoothing constant* in single exponential smoothing. Similarly, a temporally varying parameter $\lambda = \lambda(t) \in [0, 1]$ is analogous to adaptive exponential smoothing. We call λ the *persistence parameter*.

2.4. Intent Recognition:

Generally, a deformation to any given shape can either be inward (push) or outward (pull). The exact characterization of the deformation (shape, size, and location) is subject to the type of contact that is maintained on the deforming object. Along similar lines, our modeling technique has three general responses to a user input, namely push, pull, and no response. The goal is to recognize which of these represents the response to the actual intent of the user without asking the user to remember any prescribed rule (for instance, a button press or a static hand gesture). Note that the recognition of intent for "no response" is important, since it represents the robustness to accidental user input in cases when the user wishes to take rest or explore other features in a given modeling interface.

A push is characterized by an inward displacement i.e. when $\hat{\delta}_i^H < 0$. This is the simplest case wherein a user would typically approach the pot and subsequently recede away once the desired deformation has occurred. A pull is characterized by an outward displacement i.e.

when $\hat{\delta}_i^H > 0$. This is a non-trivial intent to recognize since a user would invariably approach the surface first and then recede to pull. The overall motion of the hand is similar to that of a push. However, unlike a push, a pulling action typically involves the articulation of hands so as to grasp the surface that is being deformed. In order to account for grasping actions in pulling tasks, we defined the persistence parameter as a smooth function given by $\lambda^P e^{-A(\hat{\delta}_i^N)^2}$ (Figure 7. Here, A defines the rate of decrease of the function with respect to $\hat{\delta}_i^H$ and $\hat{\delta}_i^N$ is the normalized horizontal distance of a manipulator from a given section of the pot, given by:

$$\hat{\delta}_i^N = \frac{\hat{\delta}_i^H}{\hat{\delta}_{max}^H - \hat{\delta}_{min}^H} \quad (7)$$

$$\forall i \text{ such that: } \hat{\delta}_i^H > 0$$

Our initial experiments showed that the normalization of $\hat{\delta}_i^H$ allowed us to standardize the design of bandwidth A. Also, λ^P is the maximum rate of pulling (for $\hat{\delta}_i^H = 0$).

2.5. Accidental Deformation:

Periods of rest, reflection, and accidental hand movements can sometimes lead to unintended deformations. We propose a temporal approach for avoiding accidental or unintended deformation of the pot by allowing for the pot to deform only when contact with the pot is maintained for a sufficient amount of time. We employ adaptive exponential smoothing to achieve this wherein we vary the persistence parameter λ according to a monotonically increasing function of time with a bounded range, i.e. a fixed duration of time T. Note that the same approach is also applicable to λ_P. In this work, we implemented a linear function to determine λ as follows:

$$\lambda(t) = max\left(\frac{t - T_0}{T}\tilde{\lambda}, \tilde{\lambda}\right) \quad (8)$$

Here, $\tilde{\lambda}$ and $\tilde{\lambda}_P$ represent pre-defined maximum values of the persistence parameters for push and pull respectively; T represents a pre-defined duration of time taken for λ to vary from 0 to $\tilde{\lambda}$ from the starting time T_0. The idea is to reset λ for every initial contact made with the pot and linearly increase it to a maximum prescribed value within a stipulated duration of time.

Given a handle $v_{i,j} \in C_i$ and its corresponding manipulator $\mu(t, \mathbf{p})$, the application of persistence translates to defining the increase or decrease in the radius of C_i such that $v_{i,j}$ converges to the $\mu(t, \mathbf{p})$ (Figure 6). The rate at
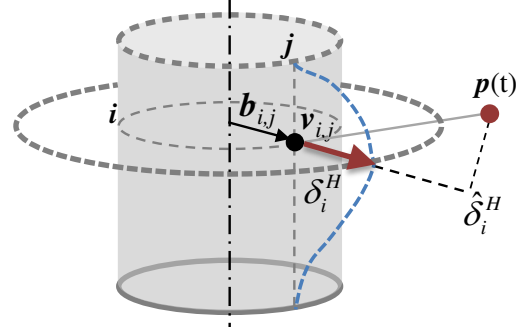

Figure 6: An illustration of proximal persistence for hand PCL-based deformation is shown.

which this convergence takes place is decided by the parameter λ as represented by the following equation:

$$v_{i,j} \leftarrow \lambda v_{i,j} + (1 - \lambda)(\hat{\delta}_i^H \mathbf{b}_{i,j}) \quad (9)$$

$$\hat{\delta}_i^H = |\langle \mathbf{b}_{i,j}, \mathbf{p} - v_{i,j}\rangle|$$

$$\mathbf{b}_{i,j} = \left[-\sin(\frac{2j\pi}{n}), 0, -\cos(\frac{2i\pi}{n})\right]^T$$

Using 9, the seed displacement and the corresponding section transformation are given by equation 10. Note that $\hat{\delta}_i^H$ is a signed displacement and can take both positive and negative values for outward (pull) and inward (push) displacements respectively.

$$C_i \leftarrow \left(\frac{r_i + \delta_i^H}{r_i}\right)C_i \quad (10)$$

$$\delta_i^H = \lambda\hat{\delta}_i^H$$

3. Guidelines for Controllability

The controllability of our deformation method is affected by two factors: (a) the disparity between what a user intends for the shape to be and what the shape actually becomes after the deformation, and (b) the responsiveness of the deformation. The goal is to minimize the disparity and *optimize* the responsiveness. Here we investigate parameters which affect the deformation behavior and responsiveness of the interaction as described below.

We can define proximal persistence by the set of parameters as $\mathcal{P}(\varepsilon_C, \varepsilon_P, \tilde{\lambda}, \tilde{\lambda}^P, A, \lambda(t), T)$ wherein the first two parameters characterize proximity and the last three parameters characterize persistence. Combined with α and β described in section 2.2, we define deformation parameters denoted as $\mathcal{D}(\alpha, \beta)$.

Figure 8: The overall setup consists of a computer and a depth camera. We implemented our approach using two such cameras, SoftKinetic DS325 and the Leap Motion controller.
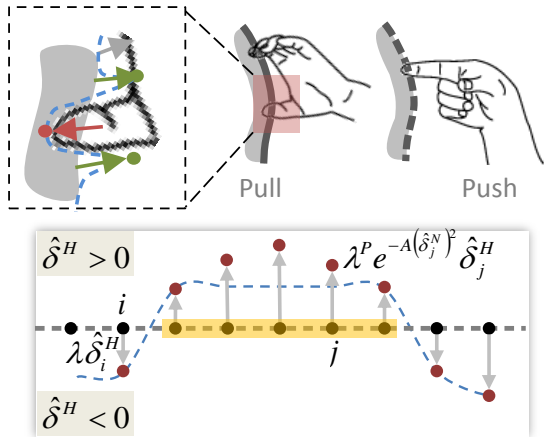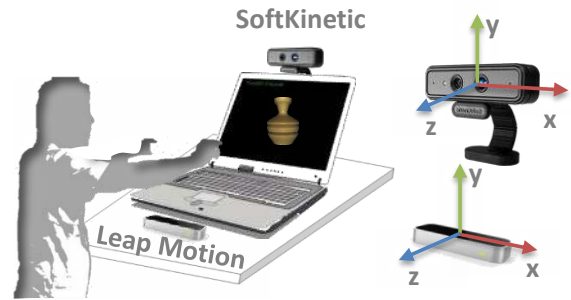
Figure 7: The intent for push and pull can be disambiguated by the identifying the shape of a manipulating object even when the overall motion is similar (top row). We apply outward attraction on the profile in terms of a smooth function decreasing with respect to the distance between that manipulators and the handle (bottom row).

3.1. Sampling and Resolution:

From our deformation strategy, it follows that our approach is affected by the density of points in the manipulating object and the resolution of the pot mesh (defined by m and n). Note that the maximum point density in the manipulating object is dependent on the resolution of the depth camera. Thus, we define the pot resolution such that the hand or tool PCL's are densely sampled in comparison to the pot. The rationale is that the sparsity of the hand or tool PCL will result in undesired effects when the user intends a global deformation.

3.2. Deformation Parameters ($\mathcal{D}(\alpha, \beta)$):

The window operator defined by α primarily decides how local the deformation will be for a *single point manipulating object*. Larger values of α result in global deformation. This parameter is also a factor in producing global deformations for a manipulating object with several points. The parameter β affects responsiveness. Recall that we apply Neumann boundary conditions for Laplacian smoothing of the deformation curve which is an *open 1-manifold*. Thus, applying the Laplace operator selectively near an internal section (i.e. a section not at the profile boundary) for a reasonably large amount of time (i.e. increasing β sufficiently) would make the deformation converge to zero making the profile non-responsive to any deformation.

3.3. Persistence Parameters ($\mathcal{P}(\varepsilon_C, \varepsilon_P, \tilde{\lambda}, \tilde{\lambda}^P, A, \lambda(t), T)$):

These parameters specifically affect the responsiveness of deformation. Assuming $\lambda(t)$ to be a constant function, the persistence parameters $\tilde{\lambda}$ and $\tilde{\lambda}^P$ are directly proportional to responsiveness. The parameter A defines the bandwidth of the exponential function for pulling. Lower values of A will distribute the rate of pull along the profile while higher values will lead to localized pulling at the points of contact on the pot. Low values of the proximity threshold ε_P require precision from the user while deforming the pot. Higher values require the user to recede from the pot at higher speed once a desired deformation has been achieved. The value for ε_P is decided based on the geometric properties of the modeling scene, the size of the pot and size of the hand. Finally, the parameter T signifies the amount of time required for the pot deformation to vary from being completely non-responsive to optimally responsive. Higher values of T would require the user to maintain contact for a longer time, thus making the deformation under-responsive.

4. Implementation

We explore our approach with two depth sensors to obtain such an input, as described below.

4.1. System Description

Our system setup consists of a ThinkPad T530 laptop computer with Dual Core CPU 2.5GHz and 8GB RAM, running 64 bit Windows 7 Professional with a NVIDIA NVS 5400M graphics card, a depth sensor (such as a SoftKinetic DS325 sensor or a Leap Motion Controller). The placement of the depth sensor varies according to needs (Figure 8). Our applications were
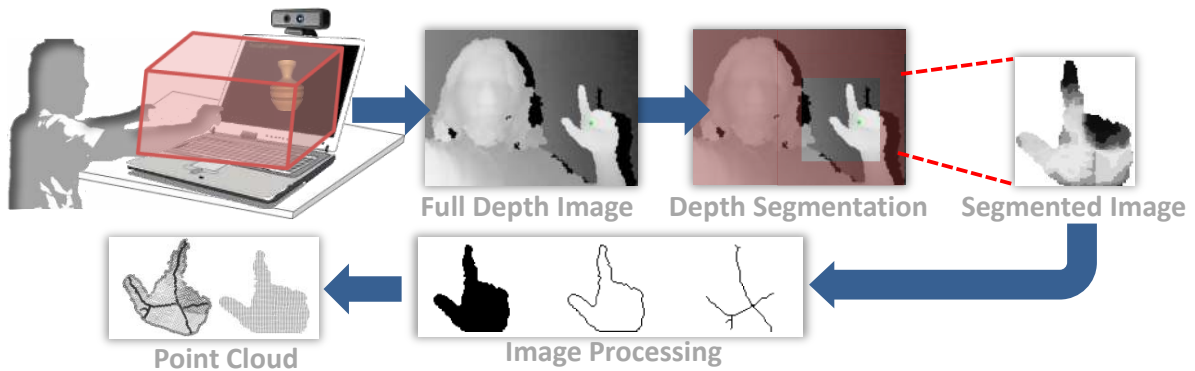
Figure 9: The depth sensing module for SoftKinetic involves the segmentation of hand's depth image from the full depth image using the Depth-Sense SDK. This is followed by optional image processing steps for skeletal and boundary images, and subsequently the segmented depth image is converted to a PCL using camera parameters.

developed in C++ and openGL with the openGL Shading Language (GLSL). We implemented two instances of our pottery application using the two different depth sensors, namely Leap Motion controller and SoftKinetic DS325 sensor.

SoftKinetic DS325 is a close range (0.1m-1.5m) time-of-flight depth sensor that provides a live video stream of the color and depth image of the scene. Every pixel on a given depth image can be converted to a 3D point using the camera parameters. We use the DepthSense^{TM}SDK to obtain the depth-map of the scene. Our first step was to segment the hand from the scene based on a depth threshold. We then convert the segmented depth map to the PCL of the hand using the camera parameters. This is akin to using a pre-defined a volumetric workspace as the active region in front of the computer screen.

Since the main data provided by the sensor is a depth map, several image processing algorithms can be applied to extract skeletal and boundary points as we will demonstrate in future sections (see Figure 9). Leap Motion provides a skeletal representation of the hand comprised of the palm and fingers. We used this device to implement virtual tools, with each tool defined by a pre-scribed set of manipulators. We used the palm position and orientation, provided by the Leap SDK, for tool manipulation.

4.2. Interface

Our interface comprises of an integrated 3D scene with a rotating pottery wheel, a set of shaping tools overlaid on natural outdoor background. In our pottery application, we combine three spatial interactions, namely, (a) 3D environment navigation though interaction space partitioning and 3D camera transitions , (b) robust tool selection using cylindrical zoning, and (c) gesture-free shaping interactions using proximal persistence. The interfaces for the two depth sensors, are similar in appearance. However, we designed additional interactions for tool selection for our Leap motion version.

4.2.1. Leap Motion Controller:

We partitioned the 3D scene into three distinct interaction spaces for unambiguous interactions (Figure 10). Each of these partitions, when active, map to the interaction space of the Leap Motion device. This allows for precise hand-motions in real space and constrains the user's focus to the active area. The user can freely transition between the spatial partitions by moving towards the right or left extremities of the Leap motion device. The *potter's wheel* partition is the main workspace wherein a user can shape the pot by modifying the lump of clay into a pot. The right and the left partitions represent the shaping and decoration tools respectively.

4.2.2. SoftKinetic:

The interface for SoftKinetic is simpler in that we did not need to use any tool selection interactions. Thus, this interface consists of the whole 3D scene without any spatial partitions. The user sees only the potter's wheel and a PCL representation of their hands, or the tools held in their hands.

5. Results

In this section, we will discuss and demonstrate three use cases for our approach, namely (a) hand, (b) physical tools, and (c) virtual tools. For the first two cases, we further explore multiple modes of sampling such as
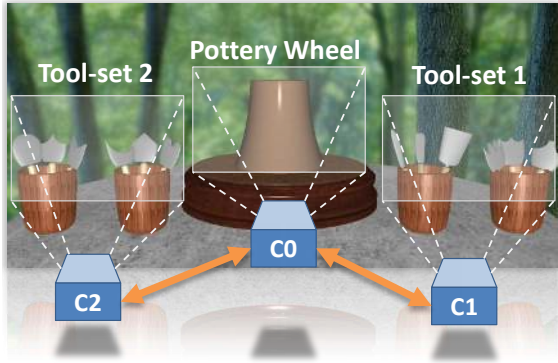
9

Figure 10: Our interface consists of active partitions of the total interaction space where each partition is associated with a virtual camera. Cameras C0, C1, and C2 correspond to the pottery wheel, tool-set 1, and tool-set 2 respectively. User's motion towards the left or right extremity results in camera transition and a subsequent re-mapping of the users physical interaction space to the active partition. The selection of tools is based on a dwell-time strategy.
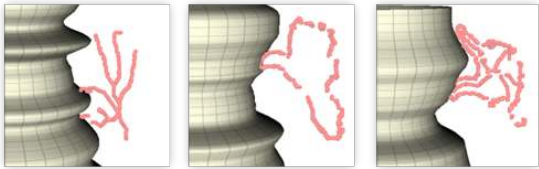

Figure 11: Persistence parameters identified with sufficiently-responsive deformation were applied for skeletal (left), boundary (middle), and hybrid (right) PCL's of the hand. While the boundary and hybrid PCL's are generally more controllable, the skeletal PCL shows better controllability for local-pulling tasks.

Figure 12: Three use cases are shown for using day-to-day physical objects (left - kitchen knife, middle - mobile phone, and right - CD-case) as shaping tools.

(i) full PCL, (ii) boundary PCL, and (iii) skeletal PCL. Based on the designed interaction space, we fixed the pot to be 0.6 units in length with radius in the range $[0.1875, 0.3125]$. Further, we fixed the resolution of the pot by assigning $m = 314$ and $n = 100$, i.e. the difference in height between each section is set to 6×10^{-4} units. After spatial mapping and PCL scaling, the average distance between two neighboring points in the PCL were observed to be 2×10^{-3}, 4×10^{-4}, and 1.5×10^{-3} units along x, y, and z directions respectively. The deformation thresholds were set as $\varepsilon_C = 10^{-8}$ $\varepsilon_P = 0.1$. Our deformation parameters $\mathcal{D}(\alpha, \beta)$ were set to $\mathcal{D}(5, 50)$. This is based on experiments we conducted for the chosen pot resolution.

5.1. Hand and Physical Tools

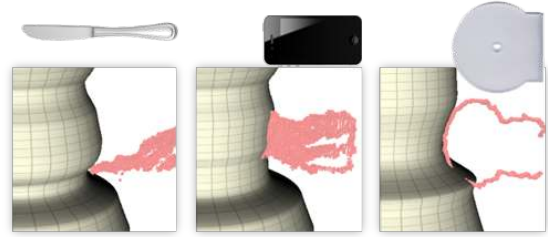The proximal persistence approach involves several parameters (section 3) which must be appropriately determined in order to design controllable interactions with our interface. Thus, our first step was to study the effect of persistence parameters ($\tilde{\lambda}$, $\tilde{\lambda}^P$, A, and T) on the intuitiveness and responsiveness of the pot deformation process. We note that effect of these parameters are not independent. Thus, an exhaustive study of all combinations is prohibitively difficult. We conducted an informal pilot study where participants used our system with six set of parameter combinations. Here, the ranges of parameters were:

- $0.1 \leq \tilde{\lambda} \leq 0.4$
- $0.1 \leq \tilde{\lambda}^P \leq 0.4$
- $A \in \{0.5, 1.0, 1.5, 2.0, 2.5\}$
- $T \in \{0.5, 1.0, 1.5, 2.0, 2.5\}$

The general trend we observed is that $\tilde{\lambda}_P > \tilde{\lambda}$ and $A < 2.0$ resulted in uncontrollable pulls. We also found that despite sufficient responsiveness, the deformations were unstable for $T < 1.5$ seconds, in that they alternated between inward and outward directions. We believe this to be a consequence of the Laplacian smoothing parameter β. Our final parameters provided by user's feedback were $\tilde{\lambda} = 0.3$, $\tilde{\lambda}^P = 0.1$, $A = 2.0$, and $T \in 1.5$. These values worked well for boundary and skeletal PCL of the hand (Figure 11). Use of physical tools proved to be more stable in comparison to the hand (Figure 12). This, we believe, can be attributed to the highly articulated nature of hand-based manipulations in comparison to the relatively rigid PCL obtained from the tools.

5.2. Virtual Tools:

Based on parameters determined from our experiments, we implemented virtual tools using the Leap Motion (Figure 13). In this case, we constrained the deformations to be purely inward. This was a design decision considering the typical use of tools in real world
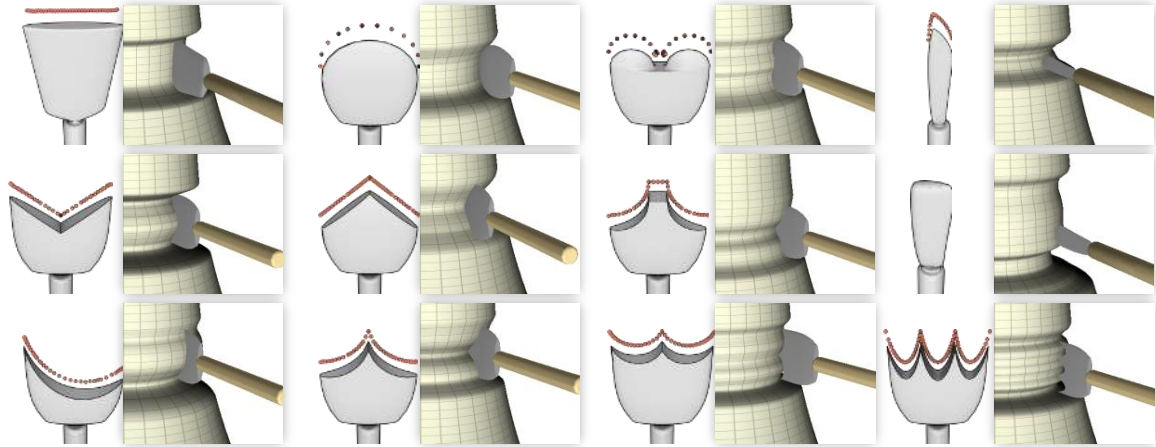
10

Figure 13: A variety of virtual tools are shown with their respective deformations on the pots.

pottery. We implemented a smoothing tool wherein we did not associate any PCL with the tool. This tool simply applies the Laplace operator on the profile of the pot itself. The use of virtual tools lends itself to rapid conceptualization of several shapes. To demonstrate this, we created a virtual chess set (figure 14) where each piece was created in less than one minute, taking a total time of approximately six minutes to create all the pieces.

5.3. Efficiency:

The main cause of computational bottleneck is the need to determine handles corresponding to each manipulator. This can be a major cause of disruption and dissatisfaction for users. For arbitrary meshes, the typical solutions in such a case would be to use spatial hashing or the GPU. In our case, however, the SHGC representation of the pot eliminates the need to use any such special method. For a pot mesh $S(P_{n,m})$, the worst case complexity to determine handles for k manipulators is $O(k(n + m))$. This is attributed to the fact that the computation of a handle $\mathbf{v}_{i,j}$ is sequentialized in first finding the closest section (i.e. i) based on only the heights of the manipulators and sections, and then finding the closest vertex per section (i.e. j). We experimented with different pot and hand PCL resolutions and found that the total time taken for a pot with 314×100 vertices deformed by a hand with 3524 manipulators was about 17 ms. Increasing the pot resolution to 470×150 vertices resulted in a total time of 29 ms for 3142 manipulators on the hand (see Table 5.2). Given that the average frame-rate for SoftKinetic is 60 Hz, our computational
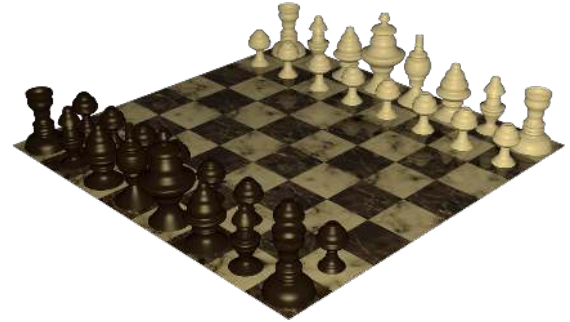

Figure 14: The results of the pot composition are shown Pots were created by participants using our full hand PCL-based interface.

efficiency is well within the required range for an interactive design application.

6. Evaluation 1: Hand PCL

In this work, we intended to enable an interaction paradigm of: "what you do is what you get". Thus, the main motivation behind our user evaluation was to understand was the relationship between the **design outcome** ("what users want in the end") and **design process** ("how they want to get there"). In order to understand how users perceive and perform mid-air shape deformations, we conducted a lab experiment with our hand PCL implementation using the SoftKinetic camera (section 5.1). Our primary goals in this evaluation were: (**a**) to observe common and uncommon user patterns during the shaping process in terms of hand grasp and (**b**) to get user's feedback on the effectiveness of controlla-

11

Table 1: The times (millisec.) taken for the steps of our algorithm are shown for different combinations of pot resolution and PCL sampling resolution.

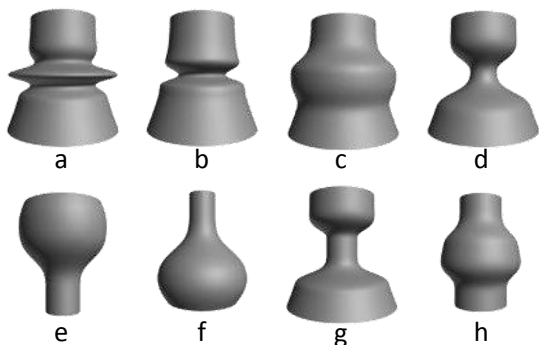| Pot Resolution	# Points in PCL	# Handles	Normal Computation	Pot-Pcl Map	Deformation Computation	Sectional Scaling	Total Time
314×100	3524	51	4.0	12.5	0.06	0.36	17
471×150	3142	69	9.7	17.3	0.11	1.62	29
314×100	816	49	6.6	5.2	0.10	0.71	13
471×150	805	66	6.6	3.1	0.07	1.30	11
628×200	821	71	26.9	10.3	0.21	4.37	42
628×200	352	73	26.5	5.1	0.20	4.33	36

Figure 15: Eight pre-defined pots were shown to participants in the quiz. These are: (a, b) thin convex and thin concave, (c,d) fat convex and concave, (e, f) round and flat, and (g, h) flat at center and ends.

bility. To cater to these goals, we performed an observational and user perception analysis. In order to support our observations and findings, we first wanted to characterize how well and how fast users could create basic shape features using our proposed algorithm. To assess the quality of shapes created by the users in relation to the time taken for creating the shapes, we utilized an approach based on curvature cross-correlation. Below, we describe our evaluation in detail.

6.1. Participants:

We recruited 15 (13 male, 2 female) science and engineering graduate students within the age range of $20-27$ years. 5 participants had familiarity with mid-air gestures and full body interactions through games (Kinect, Wii), and 12 participants reported familiarity with 3D modeling and computer-aided design. 3 participants had amateur experience with ceramics and pottery.

6.2. Procedure

The length of the study varied between 45 to 90 minutes. In the beginning of the study, each participant was given a verbal description of the setup, the purpose of the study, and functionality of the pottery application. We also recorded participants background regarding their familiarity with depth cameras, full-body games, and pottery. This was followed by a practical demonstration of how to use the application. The participants were then asked to perform the following tasks:

P *Practice:* Each participant used our application for three minutes to get an overall familiarity with the interaction of their hands with the pot surface. During this phase, the participants were allowed to ask questions and were provided guidance when required.

T1 *Quiz:* The participant was shown a pre-defined target shape and asked to shape a "blank" pot so as to roughly match the most noticeable feature of the pre-defined shape. A total of eight target shape were shown in a random order (Figure 15). Each target shape corresponds to a specific size (thin, fat) and location of a set of main geometric features (convex, concave, flat, round). The participants were allowed to undo and redo a particular deformation at any time using keyboard shortcuts. The participant could also reset the current shape to the blank pot. Once the participant was satisfied with the result, they would move to the next pre-defined pot shape.

Q1 *Questionnaire 1:* The participants were asked a series of questions regarding the intuitiveness, quality of intent recognition, responsiveness of the deformation and consistency of pushing and pulling during the quiz.

T2 *Composition:* The participants were given a duration of five minutes during which they were asked to think of certain specific pot shapes and shape them using their hands. They were asked to describe what they intended to make before begin-
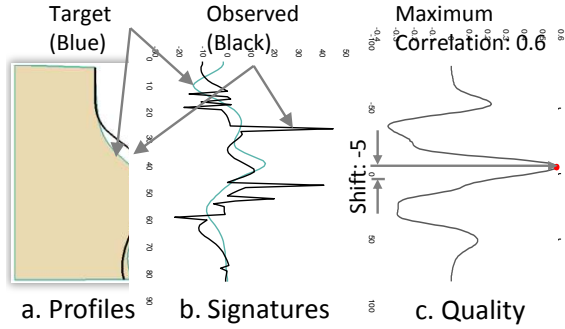
Figure 16: Curvature cross-correlation is used as measure of quality of user created pots. Given the observed and target profiles (a), we compute curvature signatures (b), and subsequently compute maximum correlation between the signatures (c). Notice the sensitivity of the curvature signature (b) for a seemingly high visual similarity. In the example shown, the profile of the observed pot are shifted upwards by 5 sections with respect to the target pot.

ning their creation. Although the duration of time was fixed, the users were allowed to complete their last composition that was started before the end of the specified duration.

Q2 *Questionnaire 2:* Finally, each participant was asked a series of questions regarding enjoyability, ease of use, and learning. We also recorded specific comments about what they liked and disliked about the application, interface, and interaction.

6.3. Metric for Quality of User Response:

The main aspect that we sought from the *Quiz* was the quality of the final outcome across participants for a given quiz problem. We also wanted to understand what features were difficult for the users to create, i.e. a comparison of the final outcomes across features. Thus, we sought to compare experimentally observed (user created) pot profiles with respect to the target shapes (quiz problems). The requirements for our metric were: (a) invariance to the shift between the features of the profiles along vertical and horizontal directions and (b) sensitivity to capture small local dissimilarities across users. Thus, we used curvature cross-correlation as a measure to compare the quality of user created profiles. We first compute the curvature signature of an observed profile wherein each point on the signature is the curvature of a point in the profile (Figure 16(a)). Subsequently, we compute the normalized cross-correlation [43] between the curvature signatures of the observation with that of the ground truth (Figure 16(b)). The quality is then defined as the maximum value of the correlation

(Figure 16(c)). Since the signatures are normalized before cross-correlation, the value of the measure of quality is in the range [0, 1]. Here, higher values represent better quality (1 corresponding to perfect match and 0 no match).

6.4. Quiz Results:

6.4.1. User Performance:

Each user perceived and approached a given target shape; there was no observable correlation between the time taken by each user and the quality (curvature cross-correlation) of the final pot created by the user for any of the target shapes. Hence, we chose to represent the user performance as a bag-plot [44] (Figure 17), where the time taken and the response quality are considered as independent variables. Rousseeuw et al. [44] state: *Like the univariate boxplot, the bagplot also visualizes several characteristics of the data: its location (the depth median), spread (the size of the bag), correlation (the orientation of the bag), skewness (the shape of the bag and the loop), and tails (the points near the boundary of the loop and the outliers).* In this section, we will identify notable aspects of user response by looking at the location, spread, and skewness of the bag-plots for each of the quiz questions. In the subsequent sections, our goal will be to report our visual observations of user behavior and usage patterns across the different quiz questions based on the bag-plot observations.

Users performed best for thin-concave and fat-convex targets with Tukey median values of (3.80, 0.71) and (1.90, 0.69) respectively. In particular, the fat-convex case shows a nearly vertical orientation indicating a strong correlation between the time taken and the quality of response. The average time taken by users was highest for round-flat features with significant variation in the quality. As expected, the thin-convex target feature was difficult to shape. This is also shown by the large spread (1.223) and the Tukey median of (0.60, 3.69). Contrary to our expectation, the top-flat-bottom-round feature (Figure 15(f)) was most difficult for the users to create, as indicated by the maximum spread (1.573) and lowest Tukey median of (0.42, 5.17). Similar difficulty was observed in the central flat feature (Figure 15(h)). The spread was consistently higher for all flat features (bottom row of Figure 17) indicating that the key problem the users faced was due to the lack of an explicit method for *smoothing* the surface of the pot. We found that the quality of responses in relation to the completion time are closely linked to how the users perceived correspondingly approached the shaping process. Below, we provide a detailed description of how
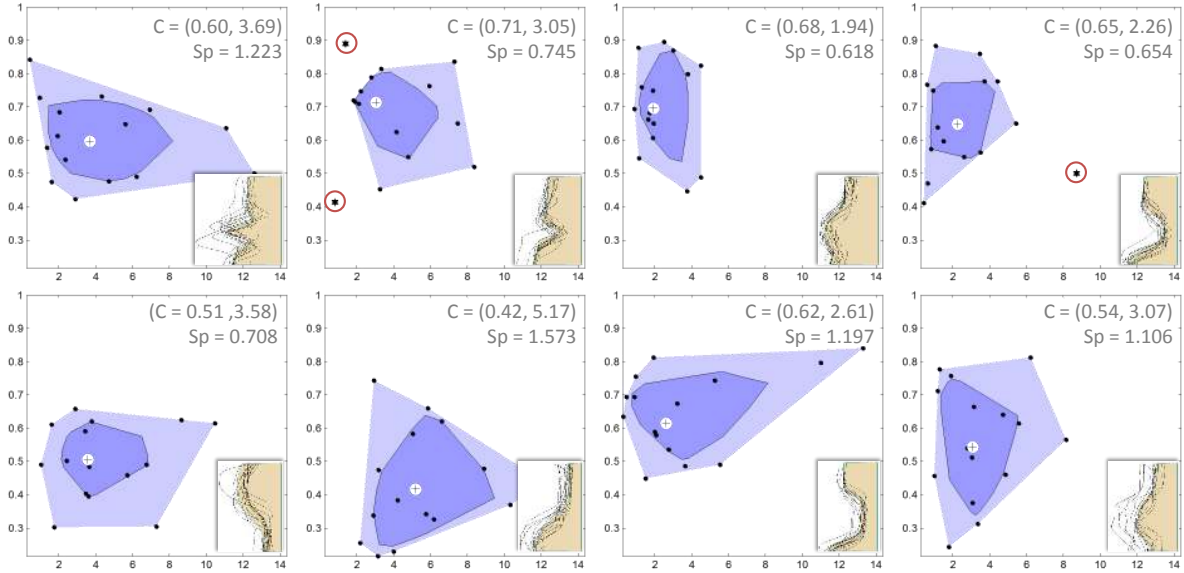
13

Figure 17: User performance is shown for the each quiz problem as a bag-plot. The x-axis is time in the range $[0, 14]$ minutes and the y-axis is the curvature cross-correlation in the range $[0, 1]$. The dark and light blue regions show the bag and fence regions, respectively. The white circle is the Tukey depth median and the points marked with red circles are the outliers. The insets show the actual pot profiles (black lines) created by the users in comparison to the target shapes (beige region) of the Quiz. The coordinates of the tukey median (C) and the spread (Sp) are provided for each target shape.
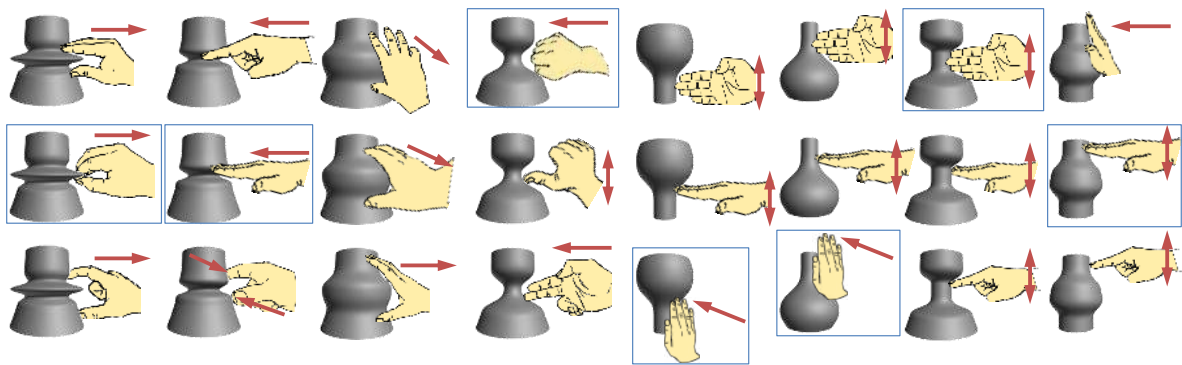

Figure 18: Common user patterns are shown in terms of grasp and motion performed by users for each target shape (in decreasing order of occurrence along columns). The hand images represent the grasp and the arrows (red) show the motion of the hand. The most successful strategies are indicated by blue boxes for each target shape.
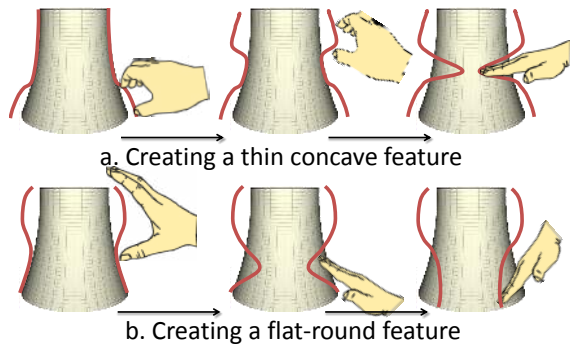
Figure 19: Two examples are shown of common deformation strategies are shown through which users created (a) thin concave and (b) flat-round features.
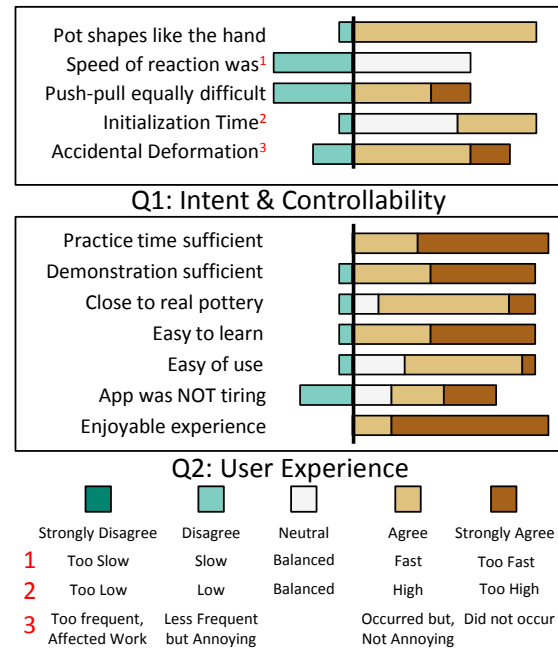

Figure 20: User response to are shown for proximal persistence. The main issue in terms of controllability (a) was the slow response and difficulty of pushing in comparison to pulling.

the measurements of the response quality and time are related to the behavior and strategies of the users for approaching, grasping, and deforming the pots.

6.4.2. User Behavior

Users generally preferred small finger level movements for thin features. For fat and flat features, we observed that the users first formed a grasp according to the amount of deformation required and then moved the whole hand to achieve the feature as expected [45]. Pushing generally required smaller finger movements in comparison to whole hand movements. Most users spent time smoothing and refining the surface of the pot after the general shape had been obtained. The motion of the hand was performed vertically along the surface of the pot (Figure 18). This led to frustration, for two reasons. First, the accidental unintended deformations caused due to the contact of the hands with regions of the pot other than the parts which the users were attempting to refine. Secondly, although our algorithm allowed for smooth deformations, there was no explicit way for the users to smooth or straighten a region of the pot. Another interesting observation was that most users avoided using the key-board commands for undo, redo, and reset. Instead, they preferred using their hands for reversing an accidental deformation. In some cases, users had to be reminded of the undo, redo, and reset functionalities.

6.4.3. Reaching, Grasping, & Deformation Strategies:

Generally, preferences towards grasping varied across users based on their expectation of the system and subsequent trial and error. However, we observed some common patterns for reaching, grasping, and deforming each target shape (Figures 18). Typically, while shaping thin-convex features, we observed that most users achieved a general convexity followed by pushing the top and bottom portions in the inward direction. We had assumed that users will create concave features in a single inward action. Interestingly, they first pulled the top and the bottom portions of the pot, and subsequently pushed the central region of the pot (Figures 19(a)). Similar was the case with flat-round features (Figures 19(b)). Many users first pulled out the round feature followed by straightening the flat regions of the pot. One of the most common hand pose that was observed for creating the thin concave feature, was the point pose. Due to the interference of the fingers other than the index finger, this pose limited the depth to which the users could push the surface inwards. Thus, most users resorted to using an open palm. For round features, the most common approach was cupping of the hands in conjunction with vertical movement of the hands.

Users frequently tried using two hands, particularly for round-flat combinations. Some users also changed their manipulating hand from dominant to non-dominant due to arm fatigue as typically expected in focused mid-air interactions. This, however, was problem due to: (a) the limited volume of the workspace, and (b) difficulty of avoiding unintended deformation due to

asynchronous motions of two hands. The tendency to approach the pot from the sides was common. Upon asking, the users typically answered" "*my own hand blocks the view of the pot*". Due to difficulty in depth perception, many users inadvertently reached behind the pot's surface. This caused confusion due to unintended deformations when the user did not expect one, or the lack of response when it was expected.

6.5. Composition:

6.5.1. User Behavior

On an average, users created 5 pots (max: 9, min: 3) within 5.75 minutes (std. dev.: 1.00 min). Most users felt that the control of the pot was significantly better during the composition phase. One user stated: "*I thought it was easier to learn the software when I was trying to make my own pot not a model one*".

This was expected because of the learning and practice that the users had during the quiz. However, according to the users' comments, the cause of difficulty in the quiz turned out to be split attention between the target shape and the user's pot. There were two observations that were common across users. First, almost all users tried making a pot that they perceived to be the most difficult ones to make. Surprisingly, these were the round-flat combinations (Figures 15(e),(f)), rather than the thin convex one (Figures 15(a)). Most of the users tried to make pots with large and straight stems, such as in a wine glass or chalice wherein most of their effort went in smoothing and straightening long vertical regions of the pot. This, in conjunction with their difficulty in the *Quiz*, strongly indicated the need for an explicit method for recognizing the intent for smoothing or fairing the surface of the pot.

6.5.2. User Feedback

Intent & Controllability: . In general, users agreed that the pot behaved according to the way the users shaped their hands (Figure 20(a)). About 50% of the users perceived the response to be slow while the remaining considered it balanced. In general, we also found a common agreement on the initialization time and robustness to accidental deformation. The most common and expected difficulty that users faced was that of: (a) pulling specific regions of the pot and (b) creating straight/flat features on the top portion of the pot. The key factor responsible was severe difficulty of depth perception. According to a user: "*Pushing seems easier than pulling. Part of the reason I suspect is the visual feedback. It is easier to determine if my hand starts to touch the pot, while it's not as easy to determine if my hand is still attached with the pottery or leaving it.*"

Figure 21: Pots that were created by young participants using our Leap-based interface.

Experience: . Despite the frustrations with pulling and smoothing, most users enjoyed the non-symbolic aspect of the interaction. According to a user: "*I enjoyed the lack of constraints in the design process; free-forming*". Another user commented: "*This is the first time I have seen something like this. The app is very sensitive so that way it gives the user a lot of freedom. i really loved that part*". The main aspects that the users liked were (a) perceived realism of pottery, (b) ease of learning, and (c) the freedom of choosing how to deform the pot (Figure 20(b)).

7. Evaluation 2: Virtual Tools

In this study, we wanted to evaluate how well our method allows young users to rapidly conceptualize shapes through direct spatial interactions. In order to do so, we wanted to provide the users with a simpler and finite set of deformation tools for pottery. Thus, we used the virtual tools implemented using the Leap Motion controller (section 5.2).

This study was conducted as an informal workshop wherein we invited twelve young participants studying in fourth, fifth, and sixth grades. We introduced our application and demonstrated the interactions. Following this, all participants were asked to create as many pots as possible in a duration of ten minutes as in the pot composition task. Figure 21 shows the pot concepts brainstormed by each participant during the pot composition task.

8. Discussion

8.1. Three-dimensionality:

There is no limitation in the method which restricts a user from manipulating the pot from any direction, as demonstrated in our results. However, we observed that the use of 2D displays is a factor, due to which users tended to use side configurations. We believe that 3D

visual feedback would encourage users to access the front and back faces. Further, hand skeleton tracking provided in the Leap API is not sufficiently robust to use in front configurations due to occlusions. Our PCL-based method works for hand configurations where gesture recognition and skeleton tracking will fail.

8.2. Limitations:

We see four limitations with our current approach. First, severe occlusion resulting from camera position and hand orientation is a problem particularly for skeleton-based gesture recognition. We partly addressed this challenge using our PCL-based approach which can make use of partial data even when the full hand skeleton is intractable. However, occlusion is an inherent problem in any camera-based method. Investigation of optimal camera position and the use of multiple cameras at strategic locations is important. Secondly, we provided a method for temporally adaptive persistence. However, this does not take the structure of the hand or the geometry of the physical tool into consideration. For instance, the optimal persistence parameters for the palm may be different from those of the fingers because of the nature of hand articulation. Hence, spatially adaptive estimation of persistence parameters needs further attention. Thirdly, in our current implementation, the definition of window operators is in terms of 2D profile topology, rather than actual distances in real space. Thus, our implementation is dependent on PCL sampling, relative to the mesh resolution of the pot. Independence from the sampling resolution may be achieved with an adaptive approach wherein new sections could be added according to manipulators, or old ones removed based on geometric properties of the pot profile such as curvature. Finally, we observed that our method gives plausible outcomes for different representations of manipulating objects such as full, boundary, and skeletal PCL. The performance and controllability in each of these modalities may differ according to the context and the user's intent. A method to determine the optimal modality for each modeling task requires further investigation.

8.3. Future Directions:

Intent disambiguation, user experience, and computational efficiency are affected by the representation of shapes, the modeling technique, and hand representation. Our approach allows for the accommodation of different representations of the hand, ranging from discrete gesture-classes to kinematic hand models. A comparative investigation of different hand representations

as well as traditional interactions would provide a comprehensive understanding of controllability of the shaping process. The augmentation of geometric information, such as normals to the PCL, may provide valuable insights on the articulation of hands and allow for determining parameter combinations adaptively based on the motion of the hands. Works by Rosman et al. [46, 47] provide frameworks for motion segmentation of point clouds and demonstrate specific examples related to hand segmentation were demonstrated. Augmentation of such frameworks with our technique could lead to several novel interaction mechanisms.

9. Conclusions

In this paper, we demonstrated that it is possible to tailor a geometric modeling method to suit the needs of controllable spatial interactions that use hands and finger motions for 3D shape modeling. This idea creates new pathways for further research, exploring hand-based shape modeling. It enables the augmentation of users' existing knowledge of real-world manipulations with new virtual modeling contexts in a "*what you do is what you get*" framework. At its core, proximal persistence is a general notion which can be instantiated in a variety of ways by combining different hand representations with different shape representations and modeling metaphors. It will be interesting to investigate methods which could automatically deduce appropriate hand representations for different modeling metaphors. With upcoming mid-air geometric design applications, achieving simultaneous efficiency, robustness, and controllability is a challenging problem towards enhancing the user's creative process and outcome. We believe that proximal persistence takes a fundamental step towards problem.

10. Acknowledgments

Ramani discloses he is a co-founder of ZeroUI that has licensed related technology.

References

[1] K. Hinckley, R. Pausch, J. C. Goble, N. F. Kassell, A survey of design issues in spatial input, in: Proceedings of the ACM Symposium on User Interface Software and Technology, 1994, pp. 213–222.

[2] C. Hand, A survey of 3d interaction techniques, Computer Graphics Forum 16 (5) (1997) 269–281.

[3] K. Hinckley, R. Pausch, D. Proffitt, N. F. Kassell, Two-handed virtual manipulation, ACM Transactions on Computer-Human Interaction 5 (3) (1998) 260–302.

[4] J. Jankowski, M. Hachet, A survey of interaction techniques for interactive 3d environments, in: Eurographics State of the Art Reports, 2013.

[5] K. Özacar, K. Takashima, Y. Kitamura, Direct 3d object manipulation on a collaborative stereoscopic display, in: Proceedings of the Symposium on Spatial User Interaction, 2013, pp. 69–72.

[6] P. Apostolellis, B. Laha, D. Bowman, A gaming interface using body gestures for collaborative navigation, in: IEEE Symposium on 3D User Interfaces, 2012, pp. 185–186.

[7] D. F. Keefe, D. A. Feliz, T. Moscovich, D. H. Laidlaw, J. J. LaViola, Jr., Cavepainting: a fully immersive 3d artistic medium and interactive experience, in: Proceedings of the ACM symposium on Interactive 3D graphics, 2001, pp. 85–93.

[8] D. F. Keefe, From gesture to form: The evolution of expressive freehand spatial interfaces, Leonardo 44 (5) (2011) 460–461.

[9] J. J. LaViola, D. F. Keefe, 3d spatial interaction: Applications for art, design, and science, in: ACM SIGGRAPH 2011 Courses, 2011, pp. 1:1–1:75.

[10] K. Korida, H. Nishino, K. Utsumiya, An interactive 3d interface for a virtual ceramic art work environment, in: International Conference on Virtual Systems and MultiMedia, 1997, pp. 227–234.

[11] S. Schkolne, M. Pruett, P. Schröder, Surface drawing: creating organic 3d shapes with the hand and tangible tools, in: Proceedings of the ACM conference on Human factors in computing systems, 2001, pp. 261–268.

[12] M. Fuge, M. E. Yumer, G. Orbay, L. B. Kara, Conceptual design and modification of freeform surfaces using dual shape representations in augmented reality environments, Computer-Aided Design 44 (10) (2012) 1020 – 1032.

[13] M. Agrawala, A. C. Beers, M. Levoy, 3d painting on scanned surfaces, in: Proceedings of the ACM Symposium on Interactive 3D Graphics, 1995, pp. 145–ff.

[14] I. Llamas, B. Kim, J. Gargus, J. Rossignac, C. D. Shaw, Twister: a space-warp operator for the two-handed editing of 3d shapes, ACM Transactions on Graphics 22 (3) (2003) 663–668.

[15] K. T. McDonnell, H. Qin, R. A. Wlodarczyk, Virtual clay: A real-time sculpting system with haptic toolkits, in: Proceedings of the 2001 Symposium on Interactive 3D Graphics, I3D '01, ACM, New York, NY, USA, 2001, pp. 179–190.

[16] D. F. Keefe, R. C. Zeleznik, D. H. Laidlaw, Drawing on air: Input techniques for controlled 3d line illustration, IEEE Transactions on Visualization and Computer Graphics 13 (5) (2007) 1067–1081.

[17] J. Lee, G. Han, S. Choi, Haptic pottery modeling using circular sector element method, in: Proceedings of the ACM international conference on Haptics: Perception, Devices and Scenarios, 2008, pp. 668–674.

[18] T. A. Galyean, J. F. Hughes, Sculpting: An interactive volumetric modeling technique, Proceedings of the ACM Conference on Computer Graphics and Interactive Techniques 25 (4) (1991) 267–274.

[19] J. Sheng, R. Balakrishnan, K. Singh, An interface for virtual 3d sculpting via physical proxy, in: Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia, ACM, New York, NY, USA, 2006, pp. 213–220.

[20] P. G. Kry, A. Pihuit, A. Bernhardt, M.-P. Cani, Handnavigator: Hands-on interaction for desktop virtual reality, in: Proceedings of the 2008 ACM Symposium on Virtual Reality Software and Technology, ACM, New York, NY, USA, 2008, pp. 53–60.

[21] A. Pihuit, P. Kry, M.-P. Cani, Hands on virtual clay, in: Shape Modeling and Applications, 2008. SMI 2008. IEEE International Conference on, 2008, pp. 267–268. doi:10.1109/SMI.2008.4548000.

[22] S. Cho, Y. Heo, H. Bang, Turn: a virtual pottery by real spinning wheel, in: ACM SIGGRAPH Emerging Technologies, 2012, pp. 25:1–25:1.

[23] O. Hilliges, D. Kim, S. Izadi, M. Weiss, A. Wilson, Holodesk: Direct 3d interactions with a situated see-through display, in: Proceedings of the ACM Conference on Human Factors in Computing Systems, 2012, pp. 2421–2430.

[24] J. Segen, S. Kumar, Gesture vr: vision-based 3d hand interace for spatial interaction, in: Proceedings of the ACM international conference on Multimedia, 1998, pp. 455–464.

[25] G. Han, J. Hwang, S. Choi, G. J. Kim, Ar pottery: Experiencing pottery making in the augmented space, in: R. Shumaker (Ed.), Virtual Reality, Vol. 4563 of Lecture Notes in Computer Science, Springer, 2007, pp. 642–650.

[26] R. Wang, S. Paris, J. Popović, 6d hands: markerless hand-tracking for computer aided design, in: Proceedings of the ACM symposium on User interface software and technology, 2011, pp. 549–558.

[27] C. Holz, A. Wilson, Data miming: inferring spatial object descriptions from human gesture, in: Proceedings of the ACM conference on Human factors in computing systems, 2011, pp. 811–820.

[28] Y. Han, B.-j. Han, Virtual pottery: a virtual 3d audiovisual interface using natural hand motions, Multimedia Tools and Applications (2013) 1–17.

[29] Vinayak, S. Murugappan, C. Piya, K. Ramani, Handy-potter: Rapid exploration of rotationally symmetric shapes through natural hand motions, Journal of Computing and Information Science in Engineering 13 (2).

[30] Vinayak, S. Murugappan, H. Liu, K. Ramani, Shape-it-up: Hand gesture based creative expression of 3d shapes using intelligent generalized cylinders, Computer-Aided Design 45 (2) (2013) 277–287.

[31] J. Song, S. Cho, S.-Y. Baek, K. Lee, H. Bang, Gafinc: Gaze and finger control interface for 3d model manipulation in {CAD} application, Computer-Aided Design 46 (0) (2014) 239 – 245.

[32] Leap motion sculpting, https://airspace. leapmotion.com/apps/sculpting/windows.

[33] P. Suryanarayan, A. Subramanian, D. Mandalapu, Dynamic hand pose recognition using depth data, in: Proceedings of the International Conference on Pattern Recognition, 2010, pp. 3105–3108.

[34] N. K. Iason Oikonomidis, A. Argyros, Efficient model-based 3d tracking of hand articulations using kinect, in: Proceedings of the British Machine Vision Conference, 2011, pp. 101.1–101.11.

[35] Leap motion skeletal tracking, https://developer. leapmotion.com/features#hand-model.

[36] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, X. Twombly, A review on vision-based full dof hand motion estimation, in: Pro-

ceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition–Workshops, 2005, pp. 75–75.

[37] T. Baudel, M. Beaudouin-Lafon, Charade: Remote control of objects using free-hand gestures, Commun. ACM 36 (7) (1993) 28–35.

[38] H. Rateau, L. Grisoni, B. De Araujo, Sub-space gestures. Elements of design for mid-air interaction with distant displays, Research Report RR-8342, INRIA (August 2013).

[39] K.-i. Kameyama, Virtual clay modeling system, in: Proceedings of the ACM symposium on Virtual reality software and technology, 1997, pp. 197–200.

[40] R. Walter, G. Bailly, N. Valkanova, J. Müller, Cuenesics: Using mid-air gestures to select items on interactive public displays, in: Proceedings of the 16th International Conference on Human-computer Interaction with Mobile Devices & Services, Mobile-HCI '14, ACM, New York, NY, USA, 2014, pp. 299–308.

[41] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, B. Levy, Polygon Mesh Processing, AK Peters, 2010.

[42] C. C. Holt, Forecasting seasonals and trends by exponentially weighted moving averages, International Journal of Forecasting 20 (1) (2004) 5–10.

[43] J.-C. Yoo, T. Han, Fast normalized cross-correlation, Circuits, Systems and Signal Processing 28 (6) (2009) 819–843.

[44] P. J. Rousseeuw, I. Ruts, J. W. Tukey, The bagplot: A bivariate boxplot, The American Statistician 53 (4) (1999) 382–387.

[45] C. Armbruster, W. Spijkers, Movement planning in prehension: do intended actions influence the initial reach and grasp movement?, Motor Control 10 (4) (2006) 311.

[46] G. Rosman, A. M. Bronstein, M. M. Bronstein, R. Kimmel, Articulated motion segmentation of point clouds by group-valued regularization, in: Proceedings of the ACM Eurographics Conference on 3D Object Retrieval, 2012, pp. 77–84.

[47] G. Rosman, A. M. Bronstein, M. M. Bronstein, X.-C. Tai, R. Kimmel, Group-valued regularization for analysis of articulated motion, in: A. Fusiello, V. Murino, R. Cucchiara (Eds.), Computer Vision–ECCV. Workshops and Demonstrations, Vol. 7583 of Lecture Notes in Computer Science, Springer, 2012, pp. 52–62.

Technique: Proximal Persistence

j^{th} Meridian

Profile (h_i, r_i)

i^{th} Section C_i

Vertex $\boldsymbol{v}_{ij}$

Pot Definition   Proximity   Persistence   Deformation

Quiz ⟵ Evaluation ⟶ Brainstorming

Pot shapes like the hand

Speed of reaction

Push-pull equally difficult

Initialization Time[2]

Accidental Deformation[3]

Application: Pottery

Depth Sensor   Hand Point Cloud

Depth Sensor   Physical Tool

Leap Motion Controller   Virtual Tool