

Shape-It-Up: Hand Gesture Based Creative Expression of 3D Shapes Using Intelligent Generalized Cylinders

Vinayak^{a,*}, Sundar Murugappan^a, HaiRong Liu^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

We present a novel interaction system, “Shape-It-Up”, for creative expression of 3D shapes through the naturalistic integration of human hand gestures with a modeling scheme dubbed *intelligent generalized cylinders* (IGC). To achieve this naturalistic integration, we propose a novel paradigm of *shape-gesture-context interplay* (SGCI) wherein the interpretation of gestures in the spatial context of a 3D shape directly deduces the designers intent and the subsequent modeling operations. Our key contributions towards SGCI are three-fold. Firstly, we introduce a novel representation (IGC) of generalized cylinders as a function of the spatial hand gestures (postures and motion) during the creation process. This representation allows for fast creation of shapes while retaining their aesthetic features like symmetry and smoothness. Secondly, we define the spatial contexts of IGCs as proximity functions of their representational components, namely cross-sections and skeleton with respect to the hands. Finally, we define a natural association of modification and manipulation of the IGCs by combining the hand gestures with the spatial context. Using SGCI, we implement intuitive hand-driven shape modifications through skeletal bending, sectional deformation and sectional scaling schemes. The implemented prototype involves human skeletal tracking and hand-posture classification using the depth data provided by a low-cost depth sensing camera (KinectTM). With Shape-It-Up, our goal is to make the designer an integral part of the shape modeling process during early design, in contrast to current CAD tools which segregate 3D sweep geometries into procedural 2D inputs in a non-intuitive and cumbersome process requiring extensive training. We conclusively demonstrate the modeling of a wide variety of 3D shapes within a few seconds.

Keywords: Generalized cylinders, Gesture Classification, NUI, KinectTM

1. Introduction

Ever since the invention of mouse in 1970 [1], computational research and advancements in technology have been motivated towards providing greater and better affordances for natural human-computer interaction (HCI). As a result, developments of novel algorithms for better virtual interfaces and innovations in hardware technology have become mutual partners, driving each other towards making HCI more intuitive and accessible. During the evolution of user interfaces, engineering design in the past has been supported by windows-icons-menu-pointer (WIMP) based computer-aided design (CAD) tools; when the main concerns were related

to the use of computation to support detailed design and manufacturing. Although CAD has been the backbone of industrial development for decades, the significance and demand of creative thinking in early design in the recent past has increased substantially. Over the last decade, Post-WIMP interfaces started to gain acceptance among the engineering design community [2]. Recently, the gaming industry saw a disruptive change with Microsoft[®] KinectTM (Kinect), where the human became the controller. The recent success of Kinect in the gaming industry is a direct example of the importance of using human motion as an expression, in a *non-intrusive* way, to create more involving, intuitive and interesting *virtual* experiences. Wigdor and Wixon [3] discuss that reality-based systems facilitate expert human-computer interaction with little or no prior instructions to the user. To this end, we believe that conceptualization of shapes by human beings is a natural

*corresponding author

Email addresses: fvinayak@purdue.edu (Vinayak), smurugap@purdue.edu (Sundar Murugappan), liu847@purdue.edu (HaiRong Liu), ramani@purdue.edu (Karthik Ramani)

process devoid of any specific tool. Thus, the instructions and training are primarily dedicated towards learning the usage of a modeling tool rather than learning how to think about shapes. This is what drives our research wherein, our intention is to bridge the gap between human expression and digital shape conceptualization during the early exploratory phases of design. To this end, we particularly consider shape modeling wherein the externalization of shape is more important than the numeric precision of the geometric model.

In this paper, we present a novel paradigm for shape interactions, *shape-gesture-context interplay* (SGCI), for creative expression of 3D shapes through the naturalistic integration of human hand gestures. To this end we develop a prototypical system, “Shape-It-Up” using a modeling scheme dubbed *intelligent generalized cylinders* (IGC). We describe a natural user interface (NUI) which facilitates cognitively simple interactions towards creative and exploratory design without the need for extensive training. With “Shape-It-Up”, our goal is to make the designer an integral part of the shape modeling process during early design.

1.1. Human Expression and Shape Conceptualization

Recently, Holz and Wilson [4] gave an interesting and comprehensive description of how natural gesticulation is critical in description of spatial objects. While 2D artifacts like sketches and drawings are better-off being created with 2D interfaces, the creation of 3D shapes using 2D interfaces may limit the capability of designers to experiment at conceptual and artistic levels [5]. Creation of organic and free-form shapes had been shown in literature using glove-based [6] and augmented reality interfaces [7]. Though these interfaces are promising in providing 3D interaction capabilities for creating a wide variety of 3D shapes, the physical interfaces are often either difficult to setup, involve wearable devices or are expensive to procure.

Theories pertaining to the natural use of hand gestures in a computer-aided design context have been developed in [8]. To this end, Horvath also investigated hand motion language for shape conceptualization [9]. Pavlovic et al. [10] illustrates the components of the human gesture interpretation process namely, video input, analysis and recognition supported by a mathematical representation of gestures and finally gesture description for further actions. Recent review of VR-based assembly and prototyping [11] states that “*the ultimate goal is to provide an invisible interface that allows the user to interact with the virtual environment as they would with the real world*”. However, VR-based technologies are typically more suitable for post-

design phases and are less affordable in terms of cost and setup-time. Although we inspire our approaches along the same lines, we do it primarily in the context of the early design phases where iteration of design prototypes necessitates the provision of an *affordable and non-intrusive environment* which can support exploratory thinking amongst designers. Our approach is based on a combination of one-handed and two-handed motions in 3D space towards the definition of the user’s intent in a shape exploration process. Based on the global and relative positions and orientation of each of the hands, our modeling system discerns the action which the user intends to perform. This enables our interaction strategies affordable to a first-time user with minimal training.

1.2. CAD and Shape Modeling

The representation of parametric shapes has been studied extensively in CAD literature. A comprehensive description of these representations can be found in [12]. Parametric surface modeling makes use of Bezier, B-Splines and NURBS for modeling complex shapes [13]. Generalized cylinders (GC) have been extensively studied in CAD and shape modeling literature [14, 15, 16] and several perspectives have been discussed towards their representation. Work [14] defines a GC as the sweep surface of the cross-sectional curve moving along the skeletal curve. The cross-sectional curve may change its shape along the skeleton. However, in most existing methods, the cross-sectional plane is typically defined by the normal and bi-normal on the skeletal curve. Work presented in [15] describes sweep surfaces as parallel, rotational, spined and synchronized based on the primitives and the sweeping rules. Special representations towards interactive deformation of GC’s were presented in [17, 14]. More recently, [16] presented a direction map based representation of GC’s which was particularly congenial for blending amongst the swept cross-sections. Deformation of generalized sweeps have also been investigated extensively in [18, 19], towards applications in human deformation.

1.3. Our Approach

We categorize the shape exploration process into three distinct components, namely, (a) shape creation, (b) shape modification and (c) shape manipulation (Figure 1). By shape creation, we mean the use of hand gestures to create a shape in an empty working volume. Shape modification refers to interactions with shapes with the intention of changing the geometric characteristics of the shape. Shape manipulation refers to the

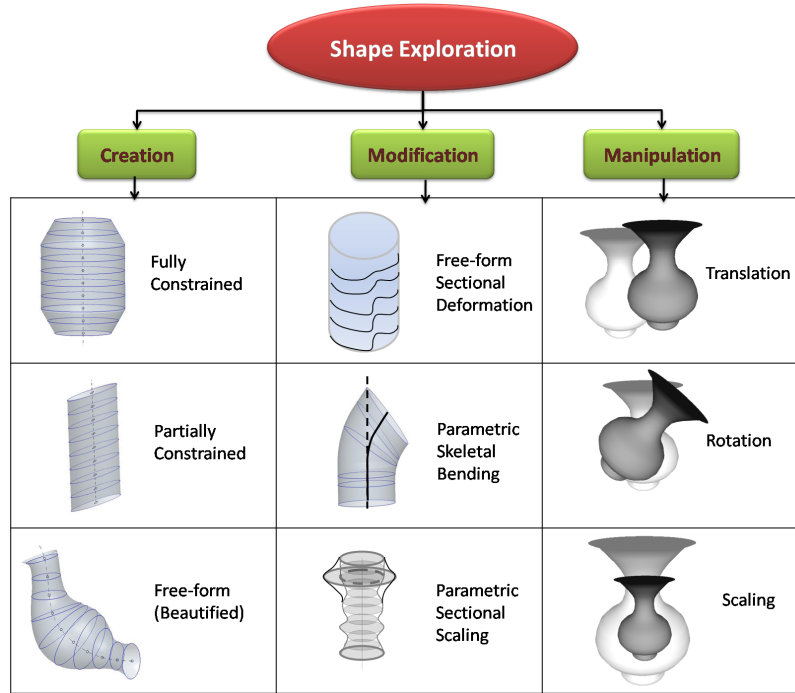


Figure 1: Components of the proposed shape exploration paradigm

rigid-body transformations for translating, rotating and scaling 3D shapes. The following sub-sections give a detailed description of our technical approach towards the shape exploration process. Work in [9] presented the conceptual and technical descriptions of hand-based interaction systems. In our work, we define hand gestures as a combination of the posture of the hands and the 3D motion of the arms. Most importantly, we encapsulate the three shape exploration components in a unified mathematical framework, SGCI and demonstrate its strength for NUI-based applications.

2. Contributions

The main contributions of our work are as follows:

- **Natural Shape Interaction** - We demonstrate a framework to support natural interactions with 3D shapes in 3D space by allowing users to directly create, modify and manipulate 3D shapes without the need for extensive training. We develop this framework using low-cost depth sensing commodity camera (Kinect™) which requires minimal setup time.
- **Shape-Gesture-Context Interplay** - We introduce a novel generic concept, namely *shape-*

gesture-context interplay (SGCI) which unifies the shape exploration process with the gestural intent of the designer. This framework enables the automatic deduction of the nature and extent of geometric constraints by interpreting human gestures and motions. Through this we develop a fundamental theoretical framework wherein the representation of a shape can be tied up seamlessly to the interactions possible on the shape.

- **Intelligent Generalized Cylinders** - We develop the idea of *intelligent generalized cylinders* (IGC) as a unified shape representation which inherently integrates the contextual interactions induced by the postures and motion of the hands. We demonstrate the creation of IGCs in the context of natural human shape expression. We use the representations to enable extremely quick creation of a variety of constrained and free-form shapes while retaining the aesthetic characteristics of the shapes.
- **Shape Exploration Approaches** - We demonstrate free-form modifications of shapes with human gestures. In contrast to the common inverse kinematics based skeletal deformation methods, we present novel skeletal-bending method along

with sectional scaling and sectional deformation schemes for enabling users to bend generalized cylinders.

3. Mathematical Framework for SGCI

In this section, we present the mathematical representations for the components of the SGCI paradigm, namely *shape*, *gesture* and *context*. We represent shapes as generalized cylinders (GC) in a way such that naturalistic modifications can be enabled for hand-based interactions. The representation of gestures is sub-divided into postures and motion.

3.1. Gesture

Given a finite set of k postures, we define a hand posture as an positive integer $\sigma \in \{0, 1, \dots, k-1\}$. Additionally, we represent the motions of a hand as the instantaneous locations $\vec{\omega}$ of the center of the hand. We define a gesture Γ as a function of time t (equation 1) which is represented by an ordered pair of the postures $\sigma(t)$ (equation 2) and locations $\omega(t)$ (equation 3) of the left and right hands respectively.

$$\Gamma(t) = (\sigma(t), \vec{\omega}(t)) \quad (1)$$

$$\begin{aligned} \sigma(t) &= (\sigma_l(t), \sigma_r(t)) \\ \sigma_l(t), \sigma_r(t) &\in \{0, 1, \dots, k-1\} \end{aligned} \quad (2)$$

$$\begin{aligned} \omega(t) &= (\vec{\omega}_l(t), \vec{\omega}_r(t)) \\ \vec{\omega}_l(t), \vec{\omega}_r(t) &\in \mathbb{R}^3 \end{aligned} \quad (3)$$

In the given set of gestures, we declare $\sigma = 0$ as the *NULL* posture. In terms of interactions, this gesture plays the role of discontinuing an ongoing interaction in progress. In the future sections we will describe the representation advantage of this imposition.

3.2. Context

Before describing the formalization of contexts, we introduce two parameters, $\varepsilon_1 \in \{0, 1, \dots, k-1\}$ and $\varepsilon_2 \in \mathbb{R}$, which we call the *operation parameters*. These parameters aid in the extract of designer's intent through the interpretation of gestures. The parameter ε_1 represents the posture classifier and ε_2 represents a threshold whose meaning changes according to the context in which it is used. We define the context Υ_Γ through the

boolean association (equation 4) of a set of modeling operations $\{\Sigma_1, \dots, \Sigma_r\}$, with a given a set of gestures $\{\Gamma_1, \dots, \Gamma_r\}$ by using a combination of the Kronecker delta ($\delta(a, b)$) and heaviside functions ($u(x)$). Here, d is a distance function defined in \mathbb{R}^3 .

$$\Upsilon_\Gamma(\varepsilon_1, \varepsilon_2; \Sigma) = \begin{cases} \Sigma & \text{if } \delta(\sigma, \varepsilon_1) = 1 \ \& \ u(d - \varepsilon_2) = 1 \\ \neg\Sigma & \text{otherwise} \end{cases} \quad (4)$$

By appropriate selection of the operation parameters, we will be able to select amongst modeling operations like creation, modification and manipulation according the postures and locations of the hands.

3.3. Generalized Cylinder (GC)

While many representations have been discussed for GC's, these representations have been mainly envisaged in the typical interaction scenarios wherein the user-input is primarily two-dimensional. Our definition of the GC is inspired from the physical action of holding a sectional curve (say a wire-loop) and sweeping it in 3D space along a path.

A 3D curve approximated by a discrete poly-line of *resolution* (number of edges) " n " can be represented as a $3 \times n$ real matrix $C = [\vec{v}_1 \ \vec{v}_2 \ \dots \ \vec{v}_n]$ wherein each column $\vec{v}_i = [x_i \ y_i \ z_i]^T$ represents a point in \mathbb{R}^3 and the columns are *ordered* to define the edges of C . Any two curves defined in the manner described can be added together and multiplied with a scalar. In a geometric sense, the addition can be interpreted as a commutative deformation of either of the two curves with respect to the other. Interestingly, the addition of a curve C_2 with identical columns to another general curve C_1 would result in the translation of C_1 . Thus, in this representation, translation is a special case of deformation of a curve. Similarly, scalar multiplication would correspond to the scaling of the curve. For the purpose of generality, we do not distinguish between closed and open curves in this representation. In our implementations, we explicitly specify the closure as per requirements without vertex duplication.

Given a set of 3D curves $\{C_1, C_2, \dots, C_m\}$ all with resolution n , the surface " S " of a GC is given by equation 5. Here, $s_i \in \mathbb{R}$, $R_i \in M_{3 \times 3}(\mathbb{R})$ and $T_i \in M_{3 \times n}(\mathbb{R})$ represent the scaling, rotation and translation of the i^{th} cross-section C_i respectively. Here, $\Delta_i^C, \Delta_i^T \in M_{3 \times n}(\mathbb{R})$ represent the deformation of cross-section C_i and the skeleton T_i respectively. The order of cross-sections defines the development of the surface in a discrete sense. In the current section we assume Δ_i^C and Δ_i^T to be a *zero* matrix

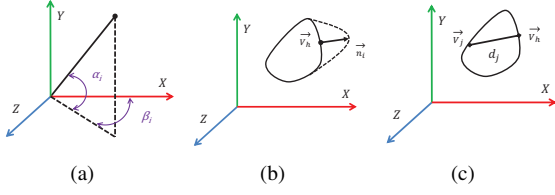


Figure 2: Section parameters (a) orientation (b) handle and deflection (c) handle-vertex distance

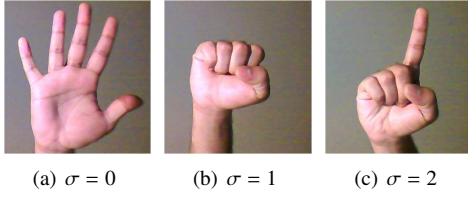


Figure 3: Selected hand postures (a)Release (b)Grab (c)Point

signifying that the cross-sectional curve is transformed as is and the trajectory is static.

$$S = \{K_i | K_i = s_i R_i (C_i + \Delta_i^C) + (T_i + \Delta_i^T), 1 \leq i \leq m\} \quad (5)$$

In this paper, we define a right-handed global coordinate frame such that the y -axis is vertical and the $z-x$ plane is horizontal. If α_i and β_i be the angles of elevation (rotation about z -axis) and azimuth (rotation about y -axis) (figure 2(a)) then the rotation $R_i = R(\alpha_i, \beta_i)$ be the rotation matrix. Similarly, the translation of the curve from the origin along a vector $\vec{o}_i = [o_{ix} \ o_{iy} \ o_{iz}]^T \in \mathbb{R}^3$ is given by equations 6. As mentioned earlier, it is easily observable that the translation can be considered as a special deformation in $M_{3 \times 3}(\mathbb{R})$ given by the product of the diagonal matrix $D(\vec{o}_i) \in M_{3 \times 3}(\mathbb{R})$ and the *unit* matrix $U_{3 \times n} \in M_{3 \times n}(\mathbb{R})$.

$$T_i = D(\vec{o}_i) U_{3 \times n} = \begin{bmatrix} o_{ix} & 0 & 0 \\ 0 & o_{iy} & 0 \\ 0 & 0 & o_{iz} \end{bmatrix} \begin{bmatrix} 1 & 1 & \dots & \dots & 1 \\ 1 & 1 & \dots & \dots & 1 \\ 1 & 1 & \dots & \dots & 1 \end{bmatrix} \quad (6)$$

3.4. Sectional and Skeletal Deformations

Geometrically, the deformation of a curve can be seen as a set of translations on each point on the curve. Also, this interpretation holds irrespective whether a curve is closed or open. Thus, for a discrete curve of resolution

n , the deformation can be described by a set of n translations, one for each point on the curve. We represent the deformation of a curve C using a simple matrix addition of the Δ curve. We present Δ as a *curve* simply because it has the same representational properties as the curve C , i.e. it has the same resolution as C and the columns of Δ are ordered. More precisely, there is a unique one-to-one correspondence between C and Δ . From interaction point-view, deformation requires a point on the curve which is being deflected (pulled or pushed) in a certain direction. In case of sectional deformation, we call this point the *handle* $\vec{v}_h \in C_i$ and the deflection is defined as a vector $\vec{\eta}_i = [\eta_{ix}, \eta_{iy}, \eta_{iz}]^T \in \mathbb{R}^3$ (figure 2(b)). The remaining task is to transfer the deflection to the other points on the curve. For sectional deformations, we achieve this by defining a *deformation-transfer* function $f(d_j)$ which scales the deflection on the handle and applies to the j^{th} point \vec{v}_j on the curve based on the euclidean distance d_j between the handle and j^{th} point (figure 2(c)). Thus, Δ_i can be formulated as follows:

$$\Delta_i^C = D(\vec{\eta}_i) U_{3 \times n} D(f(\vec{d})) \quad (7)$$

$$= \begin{bmatrix} \eta_{ix} & 0 & 0 \\ 0 & \eta_{iy} & 0 \\ 0 & 0 & \eta_{iz} \end{bmatrix} U_{3 \times n} \begin{bmatrix} f(d_1) & \dots & \dots & 0 \\ 0 & f(d_2) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \dots & f(d_n) \end{bmatrix}$$

$$\vec{d} = [d_1, d_1, \dots, d_n]^T \quad (8)$$

$$d_j = \|\vec{v}_j - \vec{v}_h\|_2$$

In equation 7, $D(\vec{\eta}) \in M_{3 \times 3}(\mathbb{R})$ is the diagonal matrix with components of deflection in the diagonal and $D(f(\vec{d})) \in M_{n \times n}(\mathbb{R})$ is a diagonal matrix with the deformation-transfer function evaluations on the diagonal.

Skeletal deformations could be treated in the same way as sectional ones. However, we present a novel technique which is more suitable for bending of skeletal curves in this paper which is not similar to the sectional deformation described above. Since our method is specific in its construct and application, we will describe it in one of the following sections. In the current section we will maintain the general description of deformation as the addition of a curve Δ_i^T to the curve T_i .

3.5. Intelligent Generalized Cylinder

The currently known representations of GC's would typically look similar to the one described in this paper till this stage. Now we introduce the idea of the *intelligent generalized cylinder* (IGC). Shape modeling with sweep representations typically involves constraining the primitives defining the sweep surface. For instance, the representations given in [15] involve constraints of parallelism of section planes for parallel sweeps, intersection of section planes for rotational sweeps and tangency of sections with respect to the trajectory for spined sweeps. The section and skeleton were two primitives in these representations. In our scheme, we have four distinct primitives using which a GC can be developed. These are (a) sectional scale (s_i^*), (b) sectional rotation parameters (α_i^*, β_i^*) (c) sectional translation parameter (\vec{o}_i^*, T_i^*) and (d) sectional deformation (Δ_i^*). The challenge in our scenario is to model the IGC in such a way that the behavior of the GC's during creation, modification and manipulation is naturally associated with the gestures of the designer. Another important issue under consideration is that the jitter in the observed locations of the hands either due to noise in the input data or due to the inherent properties of the control of hand movements should not affect the shape modeling process adversely. The noises from the depth camera can be reduced by smoothing or computer-vision based post-processing of the human skeletal data. We currently smooth the coordinates of each of the joint locations using exponential smoothing. On the other hand, the control of hand movements and their effects on modeling could be a subject of research in its own right. Although studies in neuroscience [20] on human movement sciences can be augmented into modeling interactions, they are not within the scope of this paper. With this in consideration, we develop the IGC representation in a manner which is amenable to future developments involving refined behavioral patterns of human movement.

Before describing the IGC representation, we introduce a third parameter, $\varepsilon_3 \in \mathbb{R}$, which we call the *intelligence parameter*. This parameter, like ε_2 introduced earlier, represents a threshold whose meaning changes according to the context in which it is used. The definition of IGC involves two main modifications in the representation described in equation 5. Firstly, we redefine the four primitives stated above as functions of the hand locations (equation 9). In the context of SGCI, the functions F , G_1 , G_2 and H are intended to be based on the locations of the hands and will be defined in later sections. Secondly, we replace the unit matrix $U_{3 \times n}$ with a

heaviside matrix $\Omega_{3 \times n}$ given by equation 10. The function $u(d - \varepsilon_3)$ is the *heaviside* function.

$$\begin{aligned} s_i^* &= F(\omega(t_i), s_{i-1}^*) \\ \alpha_i^* &= G_1(\omega(t_i), \omega(t_{i-1}), \alpha_{i-1}^*) \\ \beta_i^* &= G_2(\omega(t_i), \omega(t_{i-1}), \beta_{i-1}^*) \\ R_i^* &= R(\alpha_i^*, \beta_i^*) \\ \vec{o}_i^* &= H(\omega(t_i)) \\ T_i^* &= D(\vec{o}_i^*)\Omega_{3 \times n} \\ \Delta_i^{C*} &= D(\vec{\eta}_i)\Omega_{3 \times n}D(f(\vec{d})) \\ \Delta_i^{T*} &= J(\omega(t_i)) \end{aligned} \quad (9)$$

$$\Omega_{3 \times n} = D(u(d - \varepsilon_3))U_{3 \times n} \quad (10a)$$

$$D(u(d - \varepsilon_3)) = \begin{bmatrix} u(d_x - \varepsilon_{3x}) & 0 & 0 \\ 0 & u(d_y - \varepsilon_{3y}) & 0 \\ 0 & 0 & u(d_z - \varepsilon_{3z}) \end{bmatrix} \quad (10b)$$

The introduction of the heaviside function is the key factor which converts a GC to an IGC. By choosing appropriate values for the intelligence parameter we will be able to define the geometric constraints during the creation and modification operations using the hand locations. Thus, the surface S of the IGC is represented in terms of the hands as:

$$S^* = \{K_i^* | K_i^* = s_i^* R_i^* (C_i + \Delta_i^{C*}) + (T_i^* + \Delta_i^{T*}), 1 \leq i \leq m\} \quad (11)$$

Manipulation, being a global transformation, is trivially described as $s_i^G R^G(\alpha_i, \beta_i) S^* + T_i^G$, where s_i^G , $R^G(\alpha_i, \beta_i)$ and T_i^G are the global scale, rotation and translation respectively.

This section described the mathematical framework for SGCI with IGC as the representation of the shapes that we have considered. In the following sections, we will explicitly provide functions for concretely describing the gestures and contexts for IGC's with examples and results.

4. Implementation Approach for SGCI

In this section, we describe the gestures, operations and contexts which we have implemented in this work

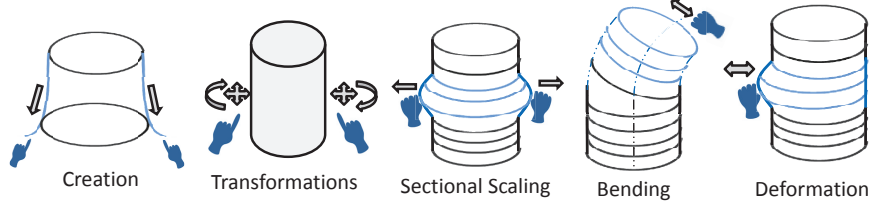


Figure 4: Physical interpretation of the SGCI framework for IGC modeling

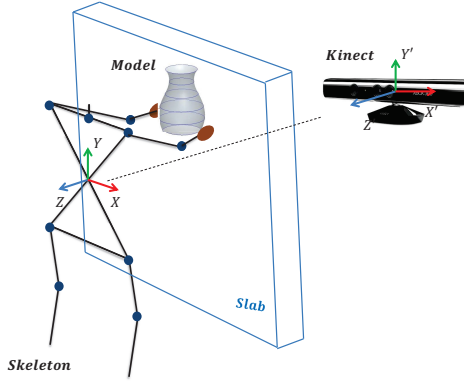


Figure 5: Virtual slab for IGC creation

towards the creative modeling of IGC's. We consider three hand postures, namely *release* ($\sigma = 0$), *grab* ($\sigma = 1$) and *point* ($\sigma = 2$) as shown in figure 3. As mentioned earlier, the *release* posture is the *NULL* posture in this set since it naturally creates an affordance for leaving contact with objects. Further, we combine the selected gestures with hand locations to define a set of five contexts leading to five modeling operations $\{\Sigma_1, \dots, \Sigma_5\}$. A physical interpretation of these contexts and the corresponding operations are shown in figure 4. The definition of functions leading to these operations can be prescribed by simply choosing appropriate operation parameters for the right (suffix *r*) and left (suffix *l*) hands (table 1). We assume that the IGC creation process (Σ_1) takes place along the global *z*-axis and the distance function is given by the euclidean distance of the hands with respect to the torso (\vec{p}_{tor}). Thus, we associate a virtual slab which moves with the designer parallel to the vertical ($x - y$) plane (figure 5).

In a naturalistic setup, bending typically takes place by holding a sufficiently slender shape with both hands. We simplify this process to a one hand operation by declaring the object stationary. Thus, we impose a spe-

Operation:	Create (Σ_1)	Deform (Σ_2)	Scale (Σ_3)	Bend (Σ_4)	Manipulate (Σ_5)
ε_{1r}	2	1(0)	1	1(0)	2
ε_{1l}	2	0(1)	1	0(1)	2

Table 1: Operation parameter definition for modeling operations

cial constraint on the bending operation that a skeleton can be bent only by grabbing the center of the top or the bottom section of an IGC. The distance functions in all other operations are defined between a general point the surface S^* of the IGC and the hand locations (equation 12). Here the symbols \wedge , \vee and $\underline{\vee}$ mean logical conjunction, logical disjunction and exclusive disjunction respectively.

$$d(\Sigma_i) = \begin{cases} \|\vec{\omega}_{r\wedge l}(t_i) - \vec{p}_{tor}\|_2 & \text{if } i = 1 \\ \arg \min_j (\|\vec{\omega}_{r\vee l}(t_i) - \vec{v}_j^{S^*}\|_2) & \text{if } i = 2 \\ \|\vec{\omega}_{r\vee l}(t_i) - \vec{o}_{1\vee m}^{S^*}\|_2 & \text{if } i = 4 \\ \arg \min_j (\|\vec{\omega}_{r\wedge l}(t_i) - \vec{v}_j^{S^*}\|_2) & \text{otherwise} \end{cases} \quad (12)$$

$$\vec{v}_j^{S^*} = j^{th} \text{ vertex of } S^* \\ m = \text{number of sections}$$

5. Hand Posture Classification by Random Forest

This section describes our approach to classify hand postures observed by a depth camera, the KinectTM. The KinectTM camera provides digital images (640×480 resolution) of its observed environment in a four-channel RGB-D format at 30 Hz wherein, the “RGB” values specify the color and “D” signifies the *depth* value (11-bit resolution) at each pixel. For considerations of both accuracy and efficiency, we use the depth-maps in conjunction with the random forest approach for hand posture classification. The random forest has proven to be a fast and effective multi-class classifier for many tasks

[21], and can be implemented efficiently on the GPU [22] as well.

The strategy for hand posture classification involves the training of the random forest with a set of hand data obtained from the KinectTM. Suppose the training dataset Ψ contains n data points, that is, $\Psi = \{p_1, \dots, p_n\}$. Each data point $p_i = \{x_i, y_i\}$, where x_i is a 80×80 patch and y_i is the label of this patch. A random forest is an ensemble of T random decision trees, each consisting of split and leaf nodes. Each split node consists of a feature f_ψ . To classify each patch x , one starts at the root of each tree, and repeatedly branches left or right according to the result of the comparison $f_\psi < 0$. At the leaf node of tree t , a learning distribution $P_t(c|x)$ is obtained. The distributions are averaged over all random decision trees in the random forest to give the final classification, that is,

$$P(c|x) = \frac{1}{T} \sum_{t=1}^T P_t(c|x) \quad (13)$$

We employ simple depth comparison features, that is, we compute

$$f_\psi(x) = d_x(u) - d_x(v) \quad (14)$$

where $\psi = (u, v)$ and u and v are 2D integer coordinates within the range $[0, 79]^2$. $d_x(u)$ and $d_x(v)$ represent the depth value of the patch x at position u and v , respectively.

Training. 20 random decision trees are trained, with each tree is trained independently on all training data, using the following algorithm:

1. Randomly propose a set of splitting candidates $\psi = (u, v)$.
2. Partition the set of examples $Q = \{p\}$ into left and right subsets by each ψ :

$$Q_l(\psi) = \{p | f_\psi(p) < 0\} \quad (15)$$

$$Q_r(\psi) = \{p | f_\psi(p) \geq 0\} \quad (16)$$

3. Compute the ψ giving the largest gain in information:

$$\psi^* = \arg \max_{\psi} G(\psi) \quad (17)$$

$$G(\psi) = H(Q) - \sum_{s \in \{l, r\}} \frac{|Q_s(\psi)|}{|Q|} H(Q_s(\psi)) \quad (18)$$

where $H(Q)$ represents Shannon entropy of the set Q .

4. If the largest gain $G(\psi^*)$ is sufficient, and the depth in the tree is below a pre-specified threshold (15 in our experiments), then recurse for left and right subsets $Q_l(\psi^*)$ and $Q_r(\psi^*)$. As mentioned earlier, we consider three hand postures, namely, release, grab and point. For each posture, we collected 10,000 frames of data

Training Size:	50%	60%	70%	80%	90%
Point	86.48%	86.45%	88.73%	88.40%	89.80%
Release	92.98%	93.40%	94.27%	94.55%	95.00%
Grab	94.00%	94.80%	95.33%	95.55%	97.50%
Average	91.15%	91.55%	92.78%	92.83%	94.10%

Table 2: Results for hand posture classification

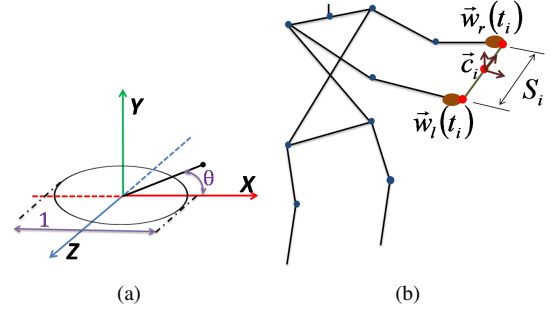


Figure 6: IGC creation (a)template section (b)IGC-hand association

in several orientations and distances from the camera, resulting in a total of 30,000 frames. For each posture, we randomly sample a portion of frames for training, and the rest are for testing. The results are reported in Table 2.

As Table 2 shows, the classification precision is very high, especially, the more training data is, the higher the precision is. Besides, it is very efficient in classification, since only hundreds of comparisons are needed for each classification. In fact, according to our experiment, the average time to classify a hand posture is only 0.039 milliseconds.

6. IGC Creation

For simplicity and symmetry, we initialize a template cross-section, C , as a circle of radius 0.5 centered at the origin (equation 19) and lying on the horizontal ($z - x$) plane (figure 6(a)). In a discrete setting, this translates to a closed regular polygon with sufficiently large resolution. The advantage of this definition is its inherent capability to represent sharp polygonal cross-sections only by setting a low resolution.

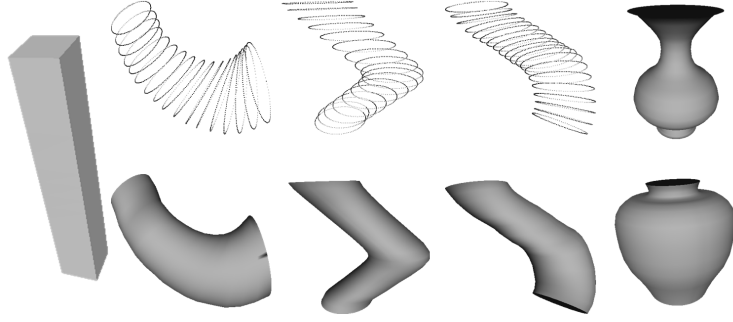


Figure 7: Fully constrained, partially constrained and free-form IGC, orientational and translational snapping in pots

$$C_i = \begin{bmatrix} 0.5 \cos(\theta_1) & \cdots & 0.5 \cos(\theta_m) \\ 0 & \cdots & 0 \\ 0.5 \sin(\theta_1) & \cdots & 0.5 \sin(\theta_m) \end{bmatrix} U_{3 \times n} \quad (19)$$

$$\forall 1 \leq i \leq m$$

$$\theta_j = \frac{2j\pi}{n}, 0 \leq j \leq n-1$$

Given a cross-section, the distance between the hands, the orientation of the line joining the two hands and the mid-point of the line joining the two hands specify the size, orientation and position of the cross-section respectively (figure 6(b)). Thus, the temporal variations of the locations of the two hands in 3D space completely define the evolution of the IGC. We use the locations of the hands, $\vec{\omega}_l(t_i)$ (left) and $\vec{\omega}_r(t_i)$ (right) at a given i^{th} instance t_i to evaluate the scaling, rotation and translation (equation 20). During the creation of an IGC, our goal is not only to allow the designer to create shape freely, but also to implicitly retain the aesthetics of the shape being created on-the-fly. This is where our intelligence parameters and heaviside function based representation play a key role. Using ε_{3s} we control the amount of variation in scale in order for it to take effect (equation 20a). If the hands are almost at the same distance from one frame to another, we use the previous scale value. Similarly, the choice of $\varepsilon_{3\alpha}$ and $\varepsilon_{3\beta}$ (equations 20c and 20d) decides about the orientation of the current section to be either equal to the previous one or orthogonal to the trajectory.

$$s_i^* = (1 - u(s - \varepsilon_{3s}))s_{i-1}^* + u(s - \varepsilon_{3s})s \quad (20a)$$

$$s = \|\vec{\omega}_r(t_i) - \vec{\omega}_l(t_i)\|_2$$

$$\vec{o}_i^* = \frac{\vec{\omega}_r(t_i) + \vec{\omega}_l(t_i)}{2} \quad (20b)$$

$$\alpha_i^* = (1 - e)\alpha_{i-1}^* + e\alpha \quad (20c)$$

$$e = \max\{u(A_\alpha - \varepsilon_{3\angle}), u(d_\alpha - \varepsilon_{3\alpha})\}$$

$$A_\alpha = |\alpha - \text{elevation}(\vec{o}_i^* \vec{o}_{i-1}^*)|$$

$$d_\alpha = |\alpha - \alpha_{i-1}^*|$$

$$\alpha = \arctan \left(\frac{\omega_{ry} - \omega_{ly}}{\sqrt{\omega_{rx}^2 - \omega_{lx}^2 + \omega_{rz}^2 - \omega_{lz}^2}} \right)$$

$$\beta_i^* = (1 - e)\beta_{i-1}^* + e\beta \quad (20d)$$

$$e = \max\{u(A_\beta - \varepsilon_{3\angle}), u(d_\beta - \varepsilon_{3\beta})\}$$

$$A_\beta = |\beta - \text{azimuth}(\vec{o}_i^* \vec{o}_{i-1}^*)|$$

$$d_\beta = |\beta - \beta_{i-1}^*|$$

$$\beta = \arctan 2 \left(\omega_{rx} - \omega_{lx}, \sqrt{\omega_{rz}^2 - \omega_{lz}^2} \right)$$

Similar to the strategy given above, we use the parameters ε_{3x} , ε_{3y} and ε_{3z} in equation 10 to *snap* the skeletal trajectory of the IGC based on the motion of the hands. In our current implementation, we have used a common threshold of $0.5mm$ for the scale and translation parameter values for IGC creation based on the size of objects modeled which is typically of the order of $50cm$ to $100cm$. Figure 7 shows shapes for a variety of intelligent constraints for the creation of IGC's. We show three varieties of IGC's wherein, (a) the scale of the GC changes with the hand motions while the skeleton is kept vertical and orientability of the sections is not allowed, (b) the skeleton can take a free-form on 3D space while the orientation is still constrained and (c) a completely free-form mode wherein the skeleton, orientation and the section scales can be varied as per the wish of the user.

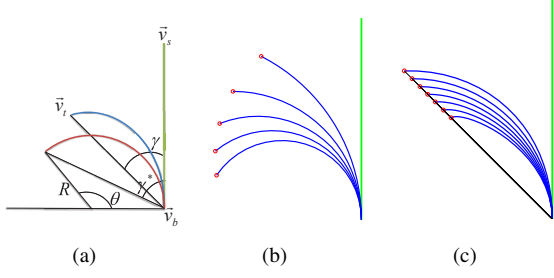


Figure 8: Skeleton deformation (a) method (b) circular bending and (c) rectification for $\theta = \pi/3$ and $0.75 \leq d_{bt} \leq 0.95$

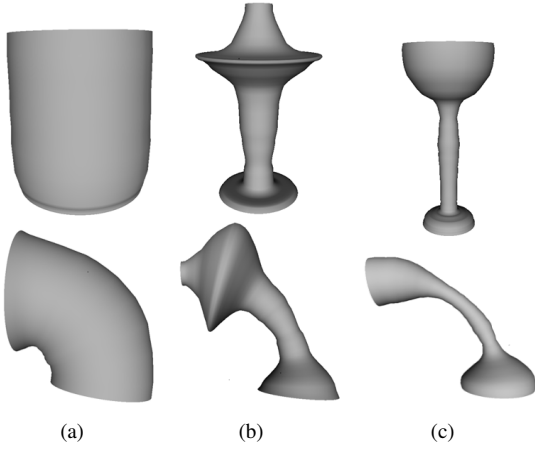


Figure 9: Recursive bending of IGC concepts

7. IGC Modification

In this paper, three operations are described for modifying IGC shapes, namely (a) *skeletal deformation* (bending), (b) *sectional deformation* and (c) *sectional scaling*. With respect to the IGC representation presented earlier, the first two modeling operations translate to the addition of a deformation curve as given in equation 11. The third operation is similar to the description given in equation 20a.

7.1. Skeletal Deformation

Skeletal deformation entails the bending of the skeletal curve T_i^* of the IGC. In this paper, we propose a bending scheme which attempts to preserve the total length of the skeletal curve. All we need to achieve this bending is a bending curve $\Delta_i^{T^*}$ which can be added to T_i^* . For simplicity, we assume that the skeleton is vertical in its initial configuration and the bending happens only on the plane defined by the base, source and target

points. First, we rotate this plane using a transformation R_B such that all the three points lie on the vertical $(x-y)$ plane and the base point is at the origin. Assuming one end of the skeleton to be fixed at the *base point* ($\vec{v}_b = R_B \vec{o}_1^{S^*}$), the problem is to find a set of rotations of each of the line segments or *links* of the skeleton such that the location of the last point moves from the *source point* ($\vec{v}_s = R_B \vec{o}_m^{S^*}$) to the location of the user's hand which we call the *target point* ($\vec{v}_t = R_B \vec{v}_t^*$). This can be seen as an inverse kinematics (IK) problem for a serial manipulator and has been very well-studied in literature [23, 24]. However, we follow a different approach based on the circular bending of a skeleton followed by a rectification method for the final bending.

First, we compute the distance, $d_{bt} = \|\vec{v}_t - \vec{v}_b\|_2$, and the angle $\gamma = \angle(\vec{v}_t \vec{v}_b, \vec{v}_s \vec{v}_b)$. Then we bend the skeleton to a circular arc taking d_{bt} as a chord with its arc length is equal to the length of the skeleton (Figure 8(a)). Obtaining the circular bend requires the determination of the radius R of the circle and the angle corresponding to the chord-length satisfying the constraint preserving the length of the skeleton (equation 21).

$$R = \|\vec{v}_s \vec{v}_b\| / \theta \quad (21)$$

$$\theta = \arg \min_{\theta \in [0, 2\pi]} \left(\left| \frac{\sin(\theta/2)}{\theta} - \frac{d_{bt}}{2\|\vec{r}\vec{b}\|} \right| \right)$$

In figure 8(a) the *bending angle* γ^* is different than the desired angle γ . Thus, in the final step, we minimize the angular error ($|\gamma - \gamma^*|$) by piecewise compensation for this error using the hyperbolic tangent function. Equation 22 gives the final coordinates, \vec{q}_i^* , of the i^{th} point \vec{q}_i on the bent skeleton. Figures 8(c) and 8(c) show a set of skeletal deformations for a bending angle of $\gamma = \pi/3$ and varying target points.

$$\vec{q}_i^* = \begin{bmatrix} \cos(\phi_i) & \sin(\phi_i) \\ -\sin(\phi_i) & \cos(\phi_i) \end{bmatrix} \vec{q}_i \quad (22a)$$

$$\phi_i = \sum_{j=1}^{i-1} \{\phi_j + w_j(\gamma - \gamma^*)\} \quad (22b)$$

$$\phi_0 = 0$$

$$w_i = \frac{\tanh(1/i)}{\sum_{i=1}^m \tanh(1/i)} \quad (22c)$$

m = Number of sections in the surface

The deformation curve is then be described in terms of the planar deformation of the skeleton as:

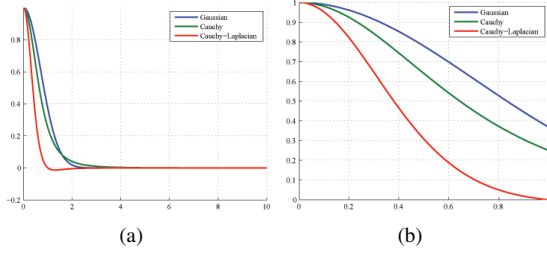


Figure 10: Deformation functions (a) long-range plot (b) close-up for range $[0, 1]$

$$\Delta_i^{T*} = R_B^{-1} \begin{bmatrix} q_{1x}^* - q_{1x} & \cdots & q_{mx}^* - q_{mx} \\ q_{1x}^* - q_{1y} & \cdots & q_{my}^* - q_{my} \\ 0 & \cdots & 0 \end{bmatrix} \quad (23)$$

Figure 9 shows the effect of our bending scheme of three different IGC's. The bending of lamp sections in figure 9(b) is undesirable since the shape of the important features have changed. In such case, a more localized method of bending based on the partial slenderness of a given IGC shape would improve the realism of these deformations.

7.2. Sectional Deformation

The general theoretical details of sectional deformation were described in the mathematical framework. The main detail of the sectional deformation method which was not discussed earlier was the *deformation-transfer* function $f(\vec{d})$ in equation 7. In this section we discuss three functions namely the gaussian function ($f_1(d) = e^{-d^2}$), cauchy's function ($f_2(d) = 1/(1 + r^2)$) and the laplacian of cauchy's function ($f_3(d) = (r^2 - 1)/(r^2 + 1)^4$). Figure 10 shows the plots of these functions. Figure 11 shows a comparison of the application of these functions on a circle of radius 0.5. We conducted experiments to observe the effect of these functions on the deformation of a circle with single as well as multiple handles. Based on the capacity of localized deformations and retainment of symmetry upon recursive deformation, we found the laplacian of the cauchy's function to be the best choice for our sectional deformation function.

The deformation of an IGC is split into two parts, first being the deformation of the *active* section C_a with center \vec{o}_a , containing the handle vertex and the second being the transfer of deformation to the remaining sections. This is achieved by scaling the deformation

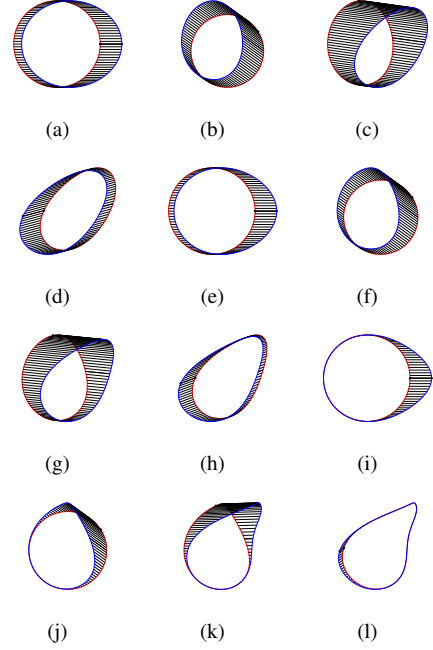


Figure 11: Recursive multi-handle deformation of a half-unit circle based on gaussian (a-d), cauchy (e-h) and laplacian of cauchy (i-l) functions

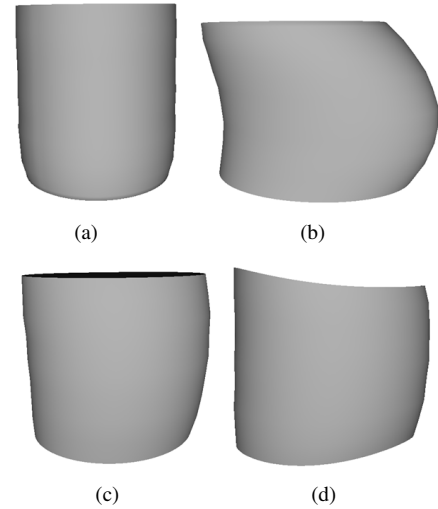


Figure 12: Deformation of a cylinder (a) original shape (b) horizontally constrained 2D deformation (c) and (d) recursive 3D deformations

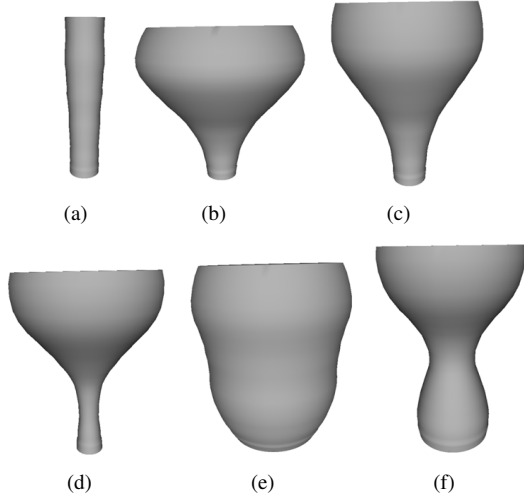


Figure 13: Recursive scaling of a cylinder

matrix $D(\vec{\eta}_i)$ with $f_3(d)$. In this case, d is the distance along the skeletal curve. Thus, the deformation matrix for the j^{th} section ($1 \leq j \leq m$) is given by $\|\vec{\partial}_j^{S^*} - \vec{\partial}_a^{S^*}\|_2 D(\vec{\eta}_i)$. Figure 12 shows the deformations of a cylinder with a variety of constraints using repeated application of $f_3(d)$ on an initial IGC.

7.3. Sectional Scaling

The general idea of sectional scaling for deformation is akin to the scaling for creation, wherein the location of the hands define the deformation of scale on a certain cross-section based on proximity. This deformation of scale is transferred to the remaining sections using the same strategy of deformation transfer function described in the previous section. If i and j be the active sections due to proximity of left and right hands to the surface S^* then the scale deformation is given by $\delta(i, j)(\|\vec{\omega}_r(t_i) - \vec{\omega}_l(t_i)\|_2 - s_i)$. Note here that the Kronecker delta used here assures that i and j are the same sections, i.e. both the hands must share proximity to the same section which then becomes the active section. For practical purposes, it is difficult to get both the hands simultaneously close to the same section. To overcome this issue, we allow a range of sections (about 10% of the total number of sections) for the proximity of the left and right hands. We select the section in the middle of this range as the active section. Figure 13 shows a sequence of recursively scaled IGC's through sectional scaling using the deformation transfer function $f_3(d)$.

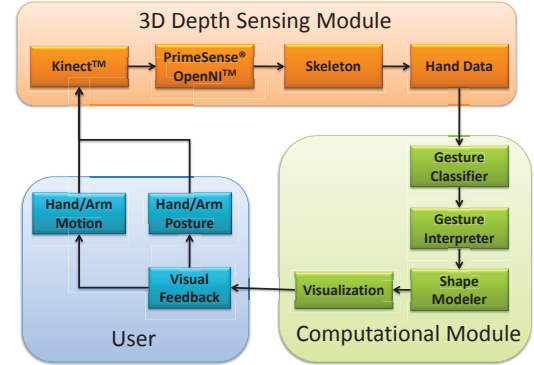


Figure 14: Pipeline showing the flow of information from the user to the proposed system

8. IGC Manipulation

The manipulation of an IGC is a global transformation activity which was described in section 3.5. In the context of hand based modeling, we use the *point* posture in conjunction with bimanual proximity (i.e. both hands near the surface) to define the context of manipulation. Here, the scaling, translation and rotation are given by equation 20, with different values of the intelligence parameter for the scaling. In the context of manipulation, i and $i - 1$ are instances of time. In this case, we intend to scale the object only with significant change in the inter-hand distance. Thus, we set the ε_3 to 2.5cm in the present work.

9. Prototype

To demonstrate natural shape exploration with the SGCI paradigm using IGC representation, we developed a prototype to support the creation of GC's. In the current implementation we assume the shape of the cross-section to be given apriori. In this case, we decided to take a circular cross-section with resolution of 100. The development of the tool was done using C++ and OpenGL was used for the online visualization of the results. We use an off-the-shelf Kinect™ camera in conjunction with the skeletal tracking capability provided by the openNI™ library to track the hand locations of the user. The pipeline for the implementation is shown in figure 14.

The interface developed in this tool is very simple, in the sense that the user only sees the global frame of reference and a working volume wherein the user can create the IGC's. Taking into account the spatial extent within which a user would typically feel comfortable

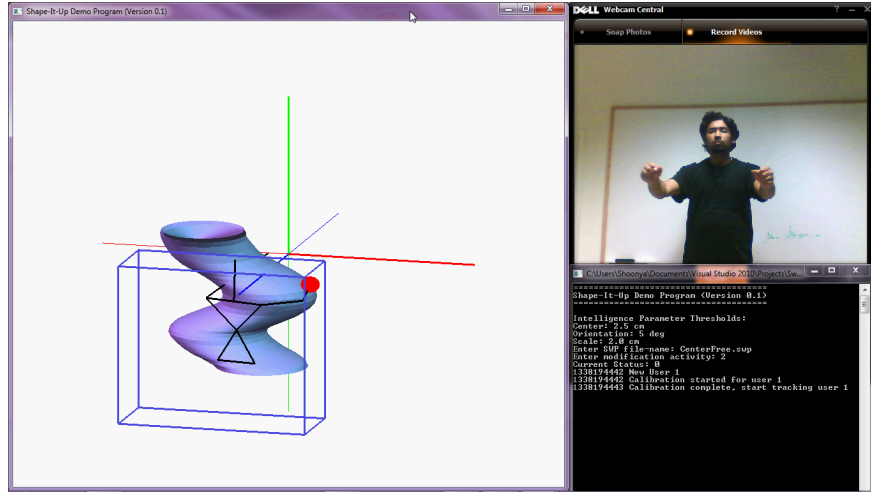


Figure 15: User interface of the prototype

working, we define the virtual slab at a distance of 75% of the total arm-length of the designer along the view direction of the Kinect™ camera. Figure 15 shows a typical interactive session using the interface developed.

10. Results and Discussion

Figure 16 shows a wide variety of shapes using the IGC representation within the SGCI paradigm. It can be appreciated that the description of shapes in the examples shown involves, to a large extent, the interpretation of what the designer wants to create. Since the actual dimensional details of a shape can always be specified as a post-process by standard parametric CAD techniques, the 3D shape creation process demonstrated in this paper is more intuitive and natural to use.

11. Conclusions and Future Work

We propose the idea of shape-gesture-context interplay (SGCI) towards natural 3D shape modeling, wherein the interpretation of gestures in the spatial context of a 3D shape directly deduces the designer’s intent and the subsequent modeling operations. Our primary focus is to propose SGCI as a generic framework which can be used to design not one but many novel gesture-driven interfaces for applications in shape modeling. In particular, we present “Shape-It-Up”, a hand gesture-driven 3D shape modeling tool for creative expression which uses a novel modeling scheme dubbed intelligent generalized cylinders (IGC). Our work contributes (a) the concept of SGCI for NUI based shape modeling and

(b) a generic formulation of generalized cylinders in the purview of SGCI, which can be used to seamlessly combine shape modeling with natural interactions. The developed framework is easy to setup and low-cost with only humans as the actors. We showcased the strength of our modeling framework through high-speed creation of complex shapes without the actual need for training which are otherwise difficult to model. “Shape-It-Up” offers only a glimpse at the variety of rich spatial interactions enabled by depth camera for 3D shape exploration. In the current implementation, the tool allows the designer to create 3D shapes. We can use the SGCI framework for testing a wide variety of contexts with different combinations of gestures and shapes. From a HCI perspective, this is important as a lot of user studies have to be done. A complete user study based on factors like fatigue, time of modeling, ease of use etc. is, in its own right, a research problem involving significant effort. Thus we plan to undertake this as a separate future research issue targeted solely towards interface design. Towards geometric modeling scheme, modeling on surfaces with holes and multiple cross-sections will be an important area which we will consider. Other future work includes deformations of the model with enforced symmetric constraints, constrained bending with natural limits and twists in cross-sections. Our work shows the potential to move the interactions from a conventional desktop based CAD environment to an NUI enabled spatial environment where the emphasis is on design and the computers become invisible.

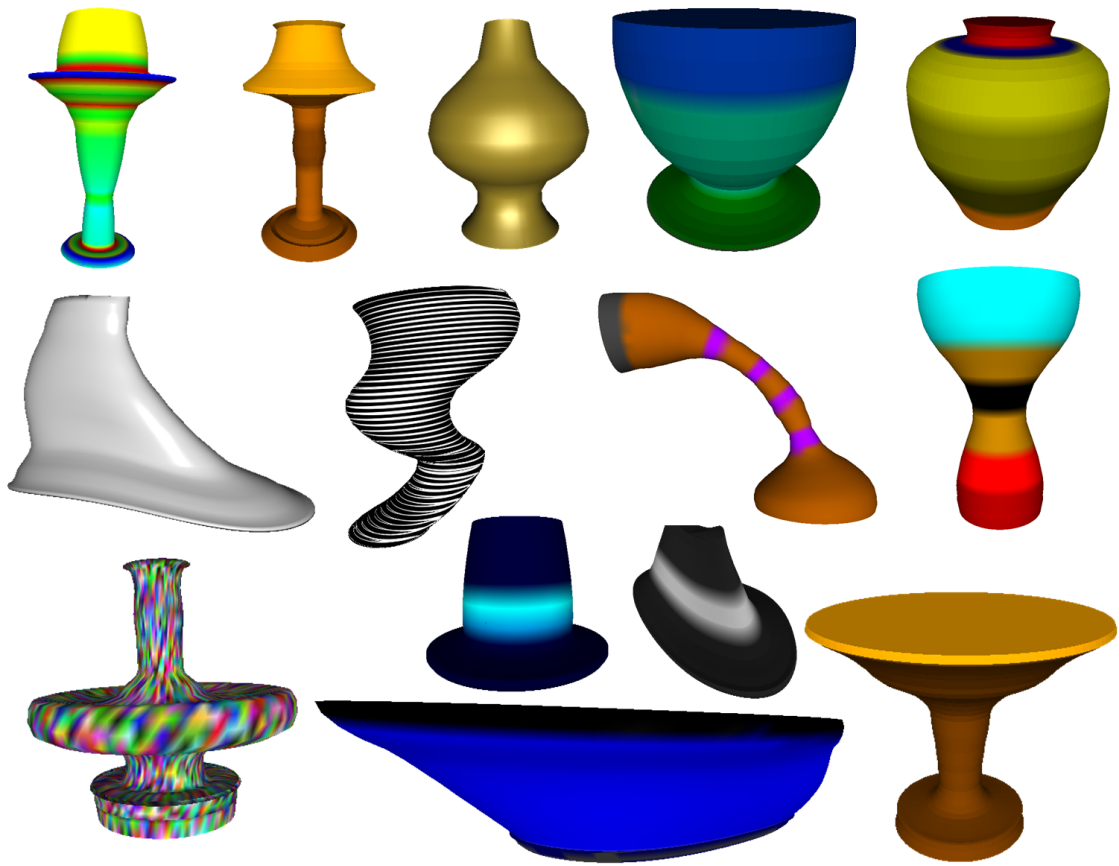


Figure 16: Shapes modeled using Shape-It-Up

Acknowledgements

This work is partly based upon work supported by the National Science Foundation Partnership for Innovation Grant# 0917959 (3D Hub), the Donald W. Feddersen Chair Professorship and the School of Mechanical Engineering. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

- [1] D. C. Engelbart, X-y position indicator for a display system (us patent# 3,541,541) (1970).
- [2] A. van Dam, Post-wimp user interfaces, *Commun. ACM* 40 (1997) 63–67. doi:10.1145/253671.253708.
- [3] D. Wigdor, D. Wixon, *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture*, 1st Edition, Morgan Kaufmann, 2011.
- [4] C. Holz, A. Wilson, Data miming: inferring spatial object descriptions from human gesture, in: *Proceedings of the 2011 annual conference on Human factors in computing systems, CHI '11*, ACM, New York, NY, USA, 2011, pp. 811–820. doi:10.1145/1978942.1979060.
- [5] E. Varga, Using hand motions in conceptual shape design: Theories, methods and tools, in: D. Talaba, A. Amditis (Eds.), *Product Engineering*, Springer Netherlands, 2008, pp. 367–382. doi:10.1007/978-1-4020-8200-9_18.
- [6] S. Schkolne, M. Pruett, P. Schröder, Surface drawing: creating organic 3d shapes with the hand and tangible tools, in: *Proceedings of the SIGCHI conference on Human factors in computing systems, CHI '01*, ACM, New York, NY, USA, 2001, pp. 261–268. doi:10.1145/365024.365114.
- [7] 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, *Comput. Aided Des.* 44 (10) (2012) 1020–1032. doi:10.1016/j.cad.2011.05.009.
- [8] I. Horváth, J. S. M. Vergeest, Theoretical fundamentals of natural representation of shapes generated with gestural devices, in: *Proceedings of TMCE 98*, 1998, pp. 393–409.
- [9] I. Horváth, On some crucial issues of computer support of conceptual design, *Product engineering ecodesign technologies and green energy* (2004) 123142doi:10.1007/1-4020-2933-0_9.
- [10] V. I. Pavlovic, R. Sharma, T. S. Huang, Visual interpretation of hand gestures for human-computer interaction: A review, *IEEE Trans. Pattern Anal. Mach. Intell.* 19 (7) (1997) 677–695. doi:10.1109/34.598226.
- [11] A. Seth, J. M. Vance, J. H. Oliver, Virtual reality for assembly methods prototyping: a review, *Virtual Reality* 15 (1) (2011) 5–20. doi:10.1007/s10055-009-0153-y.
- [12] M. Mäntylä, *An introduction to solid modeling*, Computer Science Press, Inc., New York, NY, USA, 1987.
- [13] D. F. Rogers, J. A. Adams, *Mathematical Elements for Computer Graphics*, 2nd Edition, McGraw-Hill Higher Education, 1989.
- [14] T.-I. Chang, J.-H. Lee, M.-S. Kim, S. J. Hong, Direct manipulation of generalized cylinders based on b-spline motion, *The Visual Computer* 14 (5/6) (1998) 228–239. doi:10.1007/s003710050137.
- [15] B. K. Choi, C. S. Lee, Sweep surfaces modelling via coordinate transformation and blending, *Computer-Aided Design* 22 (2) (1990) 87–96. doi:10.1016/0010-4485(90)90003-U.
- [16] J.-H. Lee, Modeling generalized cylinders using direction map representation, *Computer-Aided Design* 37 (8) (2005) 837 – 846, cAD '04 Special Issue: Modelling and Geometry Representations for CAD. doi:10.1016/j.cad.2004.09.012.
- [17] M.-S. Kim, E.-J. Park, H.-Y. Lee, Modeling and animation of generalized cylinders with variable radius offset space curves, *Teh Journal of Visualization and Computer Animation* 5 (1994) 189–207. doi:10.1002/vis.4340050402.
- [18] D.-E. Hyun, S.-H. Yoon, J.-W. Chang, J.-K. Seong, M.-S. Kim, B. Jüttler, Sweep-based human deformation, *The Visual Computer* 21 (8–10) (2005) 542–550. doi:10.1007/s00371-005-0343-x.
- [19] S.-H. Yoon, M.-S. Kim, Sweep-based freeform deformations, *Comput. Graph. Forum* 25 (3) (2006) 487–496. doi:10.1111/j.1467-8659.2006.00968.x.
- [20] G. Lee, L. Fradet, C. Ketcham, N. Dounskaia, Efficient control of arm movements in advanced age, *Experimental Brain Research* 177 (2007) 78–94. doi:10.1007/s00221-006-0648-7.
- [21] A. Liaw, M. Wiener, Classification and regression by random-forest, *R news* 2 (3) (2002) 18–22.
- [22] T. Sharp, Implementing decision trees and forests on a gpu, *ECCV 2008* (2008) 595–608doi:10.1007/978-3-540-88693-8_44.
- [23] L. Han, L. Rudolph, Inverse kinematics for a serial chain with joints under distance constraints, in: *Robotics: Science and Systems*, 2006, pp. 177–184.
- [24] L. Han, L. Rudolph, A unified geometric approach for inverse kinematics of a spatial chain with spherical joints, in: *ICRA*, 2007, pp. 4420–4427. doi:10.1109/ROBOT.2007.364160.