SketchingWithHands: 3D Sketching Handheld Products with First-Person Hand Posture

Yongkwan Kim, Seok-Hyung Bae
Department of Industrial Design, KAIST
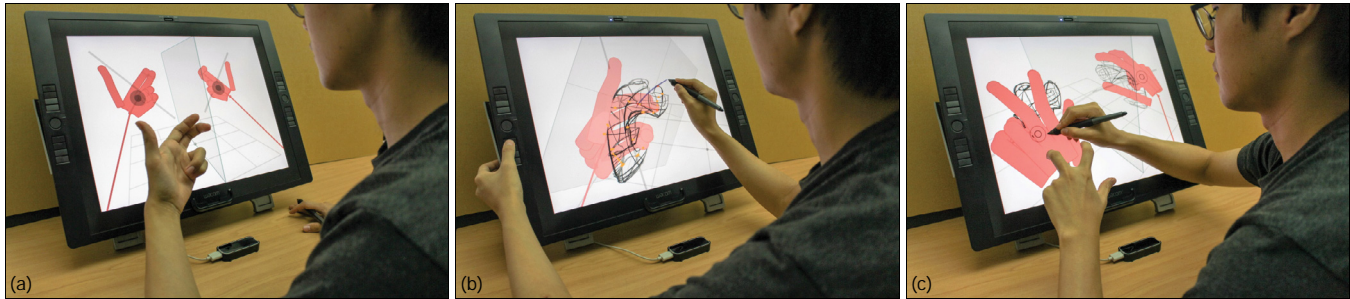yongkwan.kim | seokhyung.bae @ kaist.ac.kr

Figure 1. A designer working with SketchingWithHands: (a) capturing the posture of his hand with a hand-tracking sensor and placing it in the virtual 3D space, in which mirrors can be set up to show its reflected images, (b) creating 3D forms using freehand sketching on a sketch plane that he can freely create and manipulate in the 3D space with the captured hand information, (c) exploring and developing product concepts by modifying both the hand posture and the 3D sketches in an iterative manner.

ABSTRACT

We present SketchingWithHands, a 3D sketching system that incorporates a hand-tracking sensor. The system enables product designers to easily capture desired hand postures from a first-person point of view at any time and to use the captured hand information to explore handheld product concepts by 3D sketching while keeping the proper scale and usage of the products. Based on the analysis of design practices and drawing skills in the art and design literature, we suggest novel ideas for efficiently acquiring hand postures (palm-pinning widget, front and center mirrors, responsive spangles), for quickly creating and easily adjusting sketch planes (modified tick-triggered, orientable and shiftable sketch planes), for appropriately starting 3D sketching products with hand information (hand skeleton, grip axis), and for practically increasing user throughput (intensifier, rough and precise erasers)—all of which are coherently and consistently integrated in our system. A user test by ten industrial design students and an in-depth discussion show that our system is both useful and usable in designing handheld products.

Author Keywords
Product design; hand tracking; 3D sketching.

ACM Classification Keywords
H.5.2. Information interfaces and presentation (e.g., HCI): User Interfaces-Input devices and strategies, Interaction styles, User-centered design; I.3.8. Computer Graphics: Applications

INTRODUCTION
Many textbooks on product design emphasize the importance of drawing hands along with products while sketching to consider the scale and usage of the products [10, 11, 19, 41]. Otherwise, products manufactured later might be out of scale and not be comfortably graspable.

Nevertheless, many product designers are reluctant to draw hands in their sketches. Hands are difficult to draw due to their complex structure and shape [6, 17, 18, 19, 41], but it is easy to detect any misrepresentation because everybody knows what hands look like [41]. Exploring and finishing products takes priority for product designers, and they tend to consider drawing hands a waste of time [18].

In this paper, we thus propose a novel way that enables designers to explore handheld product ideas "with" hand information but "without" having to draw hands. By using a 3D hand-tracking sensor, designers strike natural hand poses and quickly find and capture desired poses as 3D geometrical shapes. By incorporating 3D sketching, they can effectively develop product concepts with the captured hand information as a "flexible" 3D reference that can continuously be modified as the design process goes on (Figure 1).

In the next section, we summarize related work focusing on hand tracking and 3D sketching. Based on our analysis of existing design practices and drawing skills described in the following section, we disclose our system design in terms of

functions and interaction styles, and we explain interaction and implementation details. We then show design results from our user study, discuss important findings and issues, and finally conclude this paper with future work.

RELATED WORK

In this section, we review the state-of-the-art hand tracking technologies and 3D sketching studies relevant to our work.

Hand Tracking

Hands are the best tools for object manipulation and gestural conversation due to their high levels of maneuverability and expressiveness [12, 45]. Therefore, many studies have attempted to track hands in order to develop natural and comfortable human–computer interfaces [40].

Wearing devices such as data gloves is the most accurate way of tracking hand motions [7, 14, 37]. But a noticeable weakness is that users cannot use their hands other than in the designated way, and thus their movements are restricted by devices and cables. From a practical point of view, this might not be suitable for product design sketching.

Some input devices are useful for capturing limited hand postures without wearing them. Kry et al. [23] proposed a method for obtaining a virtual hand model using a 3D mouse device that senses the pressure each finger applies, but the limited contact area and pressure range could not read certain hand gestures like a fist or open hand. Achibet et al. [1] suggested a way to control hand postures using a tablet PC. Similarly, multi-touch information cannot infer hand gestures faithfully, which is needed in our problem.

Image-based hand-tracking methods have an advantage in that users' hands are free. However, some of these methods require heavy installation and/or open space so that a camera(s) can see the user's hands [5, 7, 44]. With the prevalence of virtual reality devices such as head-mounted displays, small image-based sensors have been developed that can track hands at a short distance and in real time. These sensors are perfect for our system to place between the designer and the tablet to track the user's hands in the open space. We chose the Leap Motion controller [25] for our system due to its inexpensiveness, and simple installation and cabling.

3D Sketching

Sketching is simple but powerful. Designers can easily show 3D ideas in their 2D perspective drawing, and we can naturally understand them as objects in 3D. For this reason, many studies of sketch-based modeling have been conducted [22, 31], and commercial 3D sketching software has recently appeared on the market [43].

Due to the origin of 3D sketching, it is common to transfer traditional drawing practices and related skills to interaction techniques available in 3D sketching systems. Dorsey et al. [9] looked at architectural sketching. Bae et al. [3, 4] studied design sketching. Schmidt et al. [38] investigated analytic

drawing. Some research focused on using implicit information in design sketches to generate approximate shading [39] and to infer 3D shapes accurately [47]. Some studies tried to use existing context to create new results such as using orthographic sketches to reconstruct 3D shapes [42], using spot information to sketch 3D buildings [32], using 3D organic character models to create clothes [8], and using one's forearm to design arm braces [15]. In a similar manner, we looked into using hands to design graspable products in our study.

From a technical viewpoint regarding how to convert sketched 2D curves into unique 3D curves, 3D sketching algorithms can be categorized into two types: sketching from multiple views [3, 21], and first defining a sketch plane (or surface) and then sketching on it [4, 9, 20, 46]. The former is good for space curves that are twisted in 3D, but this requires users to follow specific steps to draw multiple curves, whereas the latter produces planar curves in 3D but gives users a lot of freedom to draw once a sketch plane has been chosen. We use a sketch plane method because our interest in this study resides in the early design exploration.

From a practical point of view, the success of a 3D sketching system using sketch planes highly depends on whether its user interface lets users quickly and accurately choose a desired sketch plane [4, 46]. 3D6B editor [20] uses icons on the screen to change the angle and position of a sketch plane, resulting in tedious user effort. Mental Canvas [9] uses predefined plane sets and traditional object controls used in most 3D modeling software. Napkin Sketch [46] provides revolving sketch planes, but the axis of revolution can only be set on the ground or on existing planes. Rhonda [33] uses a sketch plane that is always orthographic. EverbodyLovesSketch [4] uses tick gestures to set horizontal, vertical, and oblique sketch planes, but it does not allow users to adjust the angle and position of the sketch planes further. We developed tick-based sophisticated sketch plane adjusting methods that are more appropriate for product design. Some studies showed the possibility of using physical proxies of sketch planes [24, 36]. Such proxies can help users' eye–hand coordination, but they seem impractical mostly because users become fatigued.

LESSONS FROM THE LITERATURE

Before designing a new system to sketch products with hands, we studied related literature in art and design and summarized important lessons to determine the core functions and the look and feel of our system that will be described in the next section.

Drawing Hands by Observation

Traditional techniques of drawing hands are based on *drawing by observation* and thus focus on how to get desired hand models, understand their underlying structures, and reproduce the hand images on paper. We looked for lessons from these techniques to obtain and represent virtual hand models effectively in our system.

Non-dominant hands can be used as hand models (L1).
For those who want to draw hands by observation, their non-dominant hand can be a hand model [10, 17, 41]. They can easily and quickly get desired hand poses that way. Sometimes, however, they cannot see their hand from a desired viewpoint because of how it is connected to their body [6]. They cannot pose the dominant hand (or both hands) because it is already occupied with drawing [6].

Mirrors can help in hand-drawing exercises (L2).
When practicing drawing hands, it is recommended that designers use mirrors in order easily to see their hand poses from various viewpoints [6, 17]. We are very familiar with mirrors and can control our body by matching the reflected images with our kinesthetic sense [27, 28]. For this reason, many HCI studies have used mirrors [2, 5, 30].

Foreshortened cylinders are used to represent hands (L3).
In figure drawing, it is commonly taught that a hand can be simplified as series of cylinders. People can easily understand its structure and shape from the foreshortened cylinders [6, 17].

Designing Handheld Products with Hands
In product design, hands are assisting elements that provide context for products. We looked for lessons from the product design literature to develop a practical design workflow in our system that involves both hands and products.

Photos of hands are used as underlays (L4).
Due to the difficulty of drawing by observation, many designers take photos of hands and use them as underlay to copy them [10, 11, 18, 41].

3D hand models can be used instead of real hands (L5).
Some designers use 3D hand models with articulated finger joints [1]. They load one model in a digital model viewer, change its joints, find a desired camera view, and capture the screen image to make an underlay. However, due to the hand's high degree of freedom, it takes a long time to obtain desired results [1, 12, 45, 48]. For the same reason, 3D hand-viewing applications only provide hand models with preset postures for drawing exercises [13, 29].

Rendering hands should be minimal (L6).
In product design, hands should be rendered minimally in order to focus on products rather than hands. Silhouette lines, bone lines, and little shading are commonly used [11, 18, 19].

Cylindrical axes and scaffolds are good for hand grips (L7).
Most hand tools have strong directionalities in their form, resulting from their grips and/or barrels. To design such products, it is helpful to start sketching from cylindrical axes and scaffolds with foreshortened circles [10, 11, 19].

Hand pose determines product shape, and vice versa (L8).
In designing handheld products, the shapes of the products and the hands should be gradually developed while continuously affecting each other. Rather than completing one before the other, it is advisable to start with a rough volume of a product and a simplified hand posture, adding details to both in order to finish [19].

Representing 3D Shapes for Product Design
With or without hands, it is important to quickly and accurately represent design intentions in terms of product shape. We looked for lessons from general drawing skills to depict 3D shapes to improve our system's 3D sketching capability enough for product design.

Mirrors can complete symmetrical forms (L9).
Many products are symmetrical, but it is not easy to make a form exactly symmetrical. Mirrors are used in many design practices such as clay modeling and 3D modeling.

Dominant-view profiles describe the whole shape (L10).
Most products have certain views that best describe their whole shape. Designers are taught to find such a view and sketch important profiles from that view [10, 11, 19, 41].

A series of section sketches adds detail to 3D forms (L11).
Section sketches, most of which are perpendicular to or go along with dominant-view profiles, help easily construct and better describe the shape of products [10, 11, 19].

Ticks are used to mark points on planes in 3D (L12).
In design sketching, applying ticks is a simple but versatile technique. A typical example is to mark intersection points of a 3D plane and 3D curves and later to connect the points to make a cross contour [35].

Freehand sketching is effective for exploring new ideas (L13).
In the early stages of design, designers normally use very loose freehand sketching [41], which helps them to be more creative [20]. Rather than completing a perfect shape with clean strokes, they usually draw strokes over and over to modify a shape [19, 20]. It is also often required to erase part or the whole of a line(s) to get clean results as design concepts are being developed [11].

SYSTEM DESIGN
Based on the lessons that we extracted from the literature in the previous section, we designed an integrated system to be both useful and usable in real design practice.

Functions
We devised a list of functions to allow designers to quickly capture desired hand postures and effectively sketch products with the hand information. Note that lesson codes (e.g., L1) indicate which lesson was used for which function.

Capturing a desired hand posture.
For users to get a desired hand posture, they need to be able to have a free hand to pose (L1), strike hand postures as they want (L1), check postures from multiple viewpoints (L4, L5), and take 3D photo of them (L5). We thus devised a way to acquire, process, and visualize hand information digitally using a Leap Motion sensor [25]. Users can attach and detach a 3D hand model in the virtual space using the analogies of *freezing* and *melting* [49]. Instead of using the pen and button, we chose a time threshold for those actions so that users can concentrate on posing the hand to the end without moving or shaking it inadvertently. To help users freely make hand poses in their first-person view, we provide two mirrors in the virtual space: the *front mirror* to see the reflected image of the opposite side of the hand (L2), and the *center mirror* to see symmetrical hand poses and sketch symmetrical products (L9).

Starting 3D sketching with the captured hand information.
According to lessons from figure drawing and product design, we visualize virtual hand models simply with *cylindrical volumes* (L3) that are flat in color and enclosed by silhouette lines (L6). Using observations about designing products with strong directionalities, we provide users with the *axis of hand grip* and *hand skeleton* so that they can start with the captured hand information (L7).

Creating and manipulating sketch planes freely.
Quickly situating sketch surfaces where users want them is fundamental to the success of 3D sketching. We thus provide users with effective means. Users can easily create sketch planes by applying *ticks on any linear elements* (L12). One- and two-tick triggered sketch planes are *perpendicular* to the linear elements, which helps users quickly go back and forth between dominant-view (L10) and section (L11) sketches. To help users draw a series of section sketches without repeatedly defining planes, we provide functions for quickly adjusting the planes by *rotating* them and by *shifting* them along sketched curves (L11).

Drawing and erasing lines freely.
To encourage users to explore ideas in creative ways, we allow them to use *freehand sketching* without forcing them to use specific pen marks to invoke commands (L13). Because directly registering and rendering what designers draw preserves the creative process and the sketchy visuals [20], each stroke is stored and displayed in a simple *point curve* form (L13). This inevitably results in messy sketches as the design progresses. Thus, we provide two types of erasers—one is a *rough eraser* to remove whole curves, and the other is a *precise eraser* to trim part of curves (L13).

Modifying hand posture at any time.
In the course of sketching, the shapes of the products and hand postures are developed together with mutual influence (L8). Thus, users should be able to change the hand posture *any time* they want. However, posing and re-posing a virtual object in the air without tactile feedback is not an easy task [12, 48]. We thus provide a novel visualization technique called *responsive spangles* to help users check whether the hand touches the sketched product and a special widget called the *palm-pinning widget* to help them control the orientation and position of the hand with less fatigue.

Interaction
To be practical, systems should not only be intuitive, but they should also guarantee high throughput. To meet both goals, we set our interaction style to apply consistently.

Assigning roles to the two hands separately.
Users are supposed to capture hand postures and sketch products repeatedly, and their workflow might be cut off accordingly without special care. Thus, we give the two hands separate roles. The dominant hand always holds a stylus pen to sketch products and give commands, while the non-dominant hand plays the role of a hand model to be captured or a mode changer in bimanual interaction based on the kinematic chain model [16].

Avoiding round trips to menu and gestures.
Menus force users make frequent round trips. Thus, some systems use pen gestures [3, 4, 42]. But pen gestures require users to remember, take time to be applied, conflict with similar-looking geometric contents, and go with recognition problems. Introducing air gestures with a hand-tracking sensor might be one possible solution for some commands such as 3D navigation, but we expected additional problems mainly due to fatigue. Our priority was to bring our system to real designers, so we chose practical solutions—no menus and no gestures except ticking and tapping with the pen.

Providing quasi modes to frequent functions.
Li et al. [26] showed that activating quasi modes with the non-dominant hand is the best mode-changing strategy in terms of speed. We placed all frequently used functions into three categories and assigned them to three buttons on the graphics tablet so that users can immediately activate each of them when they need (Figure 2a).

Modifying modes to intensify functions semantically.
Quasi modes are a good way to access frequently used functions quickly, but the number of quasi modes is limited due to both physical constraints (e.g., the number of buttons) and mental constraints (e.g., memory capacity for immediate retrieval). To overcome those constraints to some degree, we propose a new concept called an *intensifier*, which uses an additional button on the back of the tablet (Figure 2b). We first collected similar functions in groups of two, and then assigned the general/weak function to one of the front buttons (first-level quasi mode) and the other specific/strong function to the button with the intensifier (second-level quasi mode) (Figure 2c–e). We consistently use this concept in the interaction design of the viewing control, the sketch plane control, and the eraser activation.

Tapping areas and crossing lines.
Areas are easy to tap and lines are easy to cross. We consistently applied this rule when selecting and applying tick gestures to relevant graphic elements (Figure 2f–g).
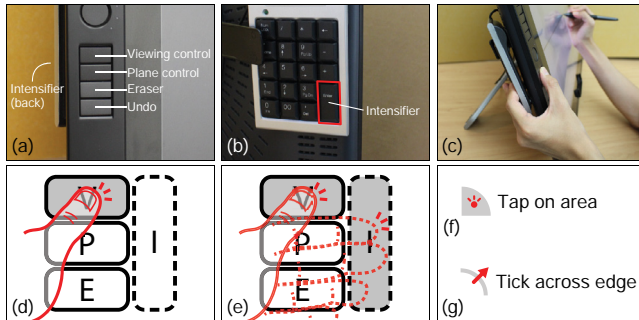

Figure 2. Buttons: (a) built-in buttons on a tablet for the three quasi modes and undo, (b) the intensifier button attached on the back of the tablet, (c) pressing on front and back buttons. Graphical notations: (d) a first-level quasi mode (front button only), (e) a second-level quasi mode (together with intensifier), (f) a tap on an area element, (g) a tick crossing a linear element.

Allowing the eraser to remove anything.
Removing all or part of a geometric entity using an eraser (or eraser cursor) is commonly accepted in many graphic software programs. We extend the usage of the eraser to remove geometric constraints as well.

Offering timely feedback and feedforward.
To help designers focus more on their job and be more aware of possible actions and expected results, we provide timely and minimal feedback and feedforward.

SKETCHINGWITHHANDS
We now describe in detail how to pose and re-pose hands and sketch products with the hand information in our system.

Freezing and Melting Hands
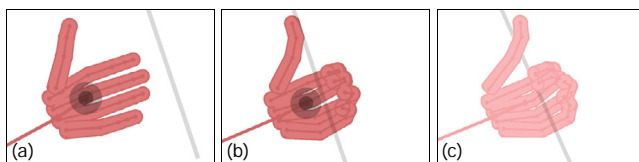The user freezes the hand by keeping the posture constant (Figure 3).

Figure 3. Freezing the hand: (a) a hand tracking sensor continuously captures the user's hand and visualizes it with a linear skeleton and a series of cylindrical volumes; (b) when the user stops moving the hand for a while (we use 3 seconds), (c) the hand gradually fades out and is eventually frozen with the axis of an imaginary grip held by the fingers.

The user melts the frozen hand when to modify or delete it (Figure 4). Both the translation and orientation of the palm are initially prevented when it is melted. At this point, the user can modify its posture by affecting only its fingers. The user can further modify its moving constraints by using the palm-pinning widget, which will be explained in the next.
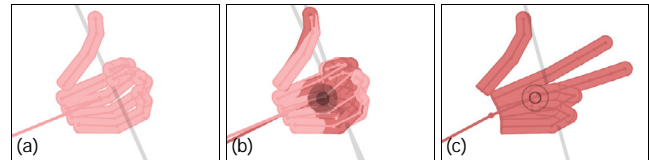

Figure 4. Melting the hand: (a) & (b) when the user makes a hand pose similar to the frozen hand pose (for 3 seconds), (c) the frozen hand gradually fades in and is eventually melted.

Constraining Hands
Controlling hand posture is difficult because of the hand's high degree of freedom. We thus provide a tool called the *palm-pinning widget*, which is always on the palm of the melted hand. This constrains the global movement of the palm so that the user can concentrate on controlling part of the hand with less fatigue from repetitive posing (Figure 5).
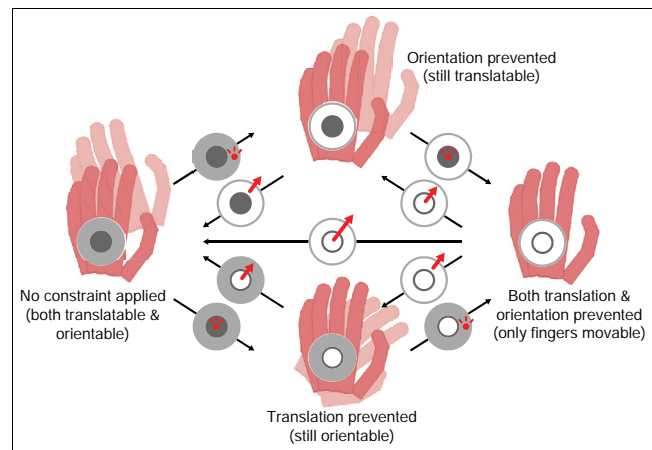

Figure 5. Palm-pinning widget: tapping to the inner shaded area constrains the position of the non-dominant hand's palm; tapping to the outer shaded area constrains the orientation; when the tapped area changes to a circle, applying a tick gesture across the circle removes the constraint.

Setting Mirrors
The user sets up the *front mirror* to check the posture from the opposite side (Figure 6a). The user can adjust the hand posture without disorientation by maintaining the first-person view with the mirror. The user sets up the *center mirror* to transfer the hand symmetrically to the opposite side. The center mirror can be utilized to check the entire form of the half-drawn product that is symmetrical (Figure 6b). The user can move the mirrors up and down at any time and in any combination (Figure 6c).
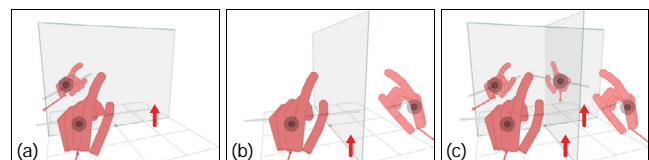

Figure 6. Mirrors: (a) setting (pulling) up the front mirror by crossing the front mirror line on the floor (cf. removing [pulling down] the mirror by crossing the line), (b) setting up the center mirror similarly, (c) using both mirrors.

Viewing

Without efficient navigation, 3D sketching does not work. The user can tumble around the point of interest in the first-level viewing mode as well as pan and dolly in the second-level viewing mode with the intensifier button pressed.

Creating Sketch Planes

The user can start 3D sketching from captured hand information and continue also with sketched curves (Figure 7). The sketch plane has a tinted transparent color so that it visualizes intersecting contours with the hand. A sketch plane is represented as a finite square with two orthogonal centerlines, the size of which is automatically adjusted to accommodate all of the strokes on it.
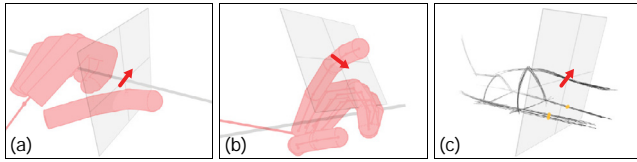

Figure 7. A sketch plane can be attached to any linear element with a tick across it on: (a) the grip axis, (b) each bone segment of the hand, (c) an existing sketched curve(s).

Considering the process of progressive product design, which requires going back and forth between dominant-view and section sketches that are relatively orthogonal, we provide a modified tick-triggered sketch plane creation method (Figure 8) that is different from the original one that defines horizontal and vertical planes [4].
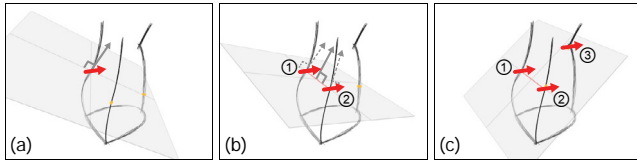

Figure 8. Tick-triggered sketch plane creation: (a) with a single tick applied to a curve(s), a sketch plane is created at the (average) intersection point on the curve(s), with the orientation perpendicular to the curve(s); (b) with two ticks, a sketch plane is created that passes through the two intersection points, with the orientation as perpendicular as possible to both the curves; (c) with three ticks, a unique sketch plane is created that passes through the three intersection points.

Manipulating and Erasing Sketch Planes

Accurately adjusting a sketch plane is critical in product design for both dominant-view and section sketches, we thus provide quick and easy methods to rotate about an axis and shift along a curve(s) in the *plane control mode* (Figure 9). While rotating, the user can easily recognize the angular changes of the sketch plane by the angle of the handle and adjust the resolution of control by distancing the pen from the rotating axis. While shifting, the user can recognize the moving direction of the sketch plane by the visualization of its surface normal vector as a feedforward. The intersection points between the sketch plane and existing curves are

updated and visualized in yellow in real time to help users manipulate the sketch plane.
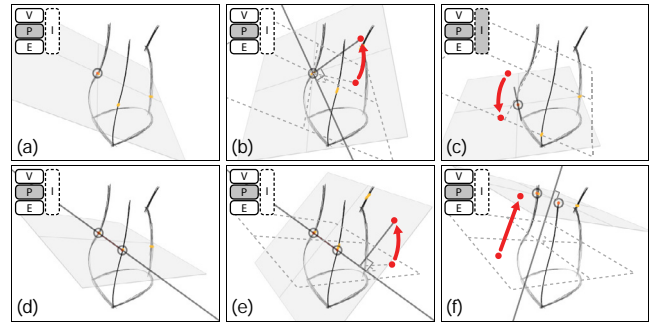

Figure 9. Manipulating sketch planes: (a) for a sketch plane with a single constraint, (b) in the first-level plane control mode, the pen-down point on the plane determines the axis of rotation and subsequent pen movement spins the plane about the axis; (c) in the second-level plane control mode, the pen movement shifts the sketch plane tangentially along the curve; (d) for a sketch plane with two constraints, (e) in the first-level plane control mode, pen movement spins the plane about the already determined axis; (f) in the second-level plane control mode, pen movement translates the plane in its normal vector.

The user can erase constraints in order quickly to redefine a sketch plane or remove it in the *eraser mode* (Figure 10a-b). The user can also erase it by crossing its boundary (Figure 10c).
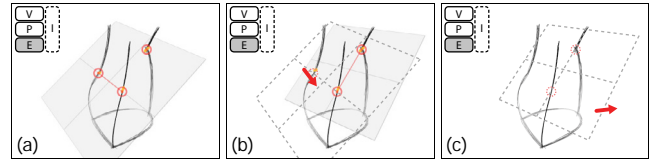

Figure 10. Removing (erasing) constraints of a sketch plane: (a) removable constraints (red circles) when the eraser mode is on, (b) a new sketch plane with the remaining constraints when one constraint is erased, (c) removing (erasing) a sketch plane by crossing its orthogonal centerlines or boundary lines.

Drawing and Erasing Sketch Strokes

The user can create sketches by simply putting pen strokes on a sketch plane (Figure 1b). Strokes generated in our system are varying in length and in the number of element points, and they are not organized as layers or groups. Therefore, selecting and modifying each stroke are not realistic. We thus offer a mixed-style eraser, which can be both *rough* and *precise*, as a powerful solution with which the user can "sculpt" entire sketches based on the user's desired silhouette in mind (Figure 11). The rough one can effectively remove entire strokes in a complex scene. The precise one is good at trimming extended strokes. The user can use both types of erasers in combination by pressing/releasing the intensifier button in the *eraser mode*. This strategy is similar to conventional 3D sculpting systems such as ZBrush [34]. In comparison, our method requires less effort and time because the user of our system only needs to deal with lines and not 3D volumes.
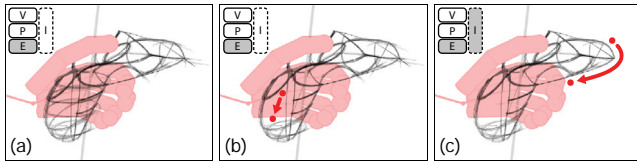
Figure 11. Erasing 3D curves: (a) a given 3D sketch, (b) the rough eraser that removes the whole of each 3D curve when it touches any part of the curve in the first-level eraser mode, (c) the precise eraser that removes only part of each 3D curve that it sweeps in the second-level eraser mode.

Changing Rendering Styles of Hands

As a product sketch develops, the user might need different information on the hands, and could choose an appropriate style of hand visualization accordingly (Figure 12).
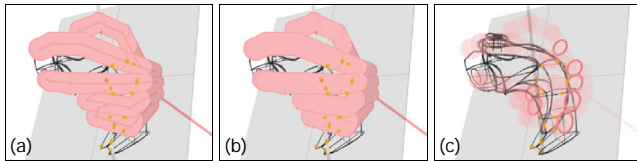

Figure 12. Various rendering styles for hands: (a) opaque hand with its skeleton for defining sketch planes, (b) opaque hand without the skeleton for checking the overall look of the hand and product, (c) transparent hand with intersection contours with a sketch plane for sketching shapes occluded by the hand.

Re-posing and Sketching with Responsive Spangles

To provide the user with visual information on where the hand is positioned with respect to the sketches, we developed *responsive spangles*. Responsive spangles are evenly distributed on each sketched 3D curve, and each spangle's size and color change based on the distance to the closest point on the hand skeleton, with its orientation always facing the point (see the Appendix for implementation details). Because responsive spangles

behave as if they are a "flexible" and "penetrable" net wrapping sketched models, the user can easily make contact while watching ripples on the imaginary net (Figure 13a-c) or revise sketched strokes while watching the colors of spangles (Figure 13d-f).
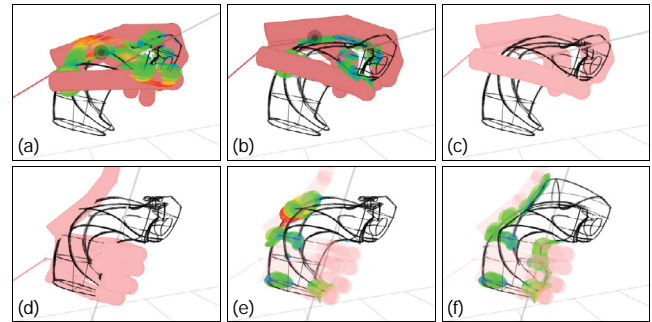

Figure 13. Responsive spangles: (a) spangles on the sketched strokes showing collision information by responsively changing their size and color (approaching: small and bluish, contact: large and greenish, collision: large and reddish), (b) the hand adjusting its pose with most of spangles in green, (c) a successfully posed and frozen hand fit to the sketch, (d) a different hand posture applied for idea exploration, (e) spangles showing collision under the thumb base and no contact with index and middle fingers, (f) redrawn strokes changing the spangles in green, resulting in new design variation fit in the hand.

Walking Through an Example Handheld Product Design

To show how the proposed functions and interaction techniques are holistically integrated to help product designers explore ideas and finalize quality outcomes, we present a step-by-step design example in Figure 14.

IMPLEMENTATION

We built our hardware system (Figure 1) using a Leap Motion controller, a Wacom Cintiq 21UX display tablet
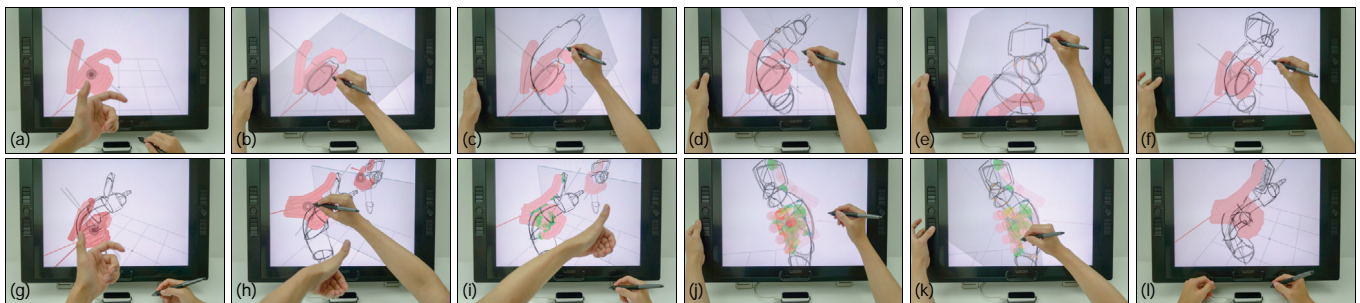

Figure 14. Designing a smart hand drill: (a) obtaining a 3D hand model with a posture to hold a new drill in mind (freezing), (b) sketching a profile curve on a section with a trigger for the index finger (a tick on the grip axis), (c) sketching a dominant-view profile (two ticks on the cross section), (d) sketching cross sections along the dominant-view profile (shifting the sketch plane), (e) sketching a touch screen on the upper body of the drill (zooming in and adjusting the position and orientation of the sketch plane), (f) removing unnecessary cross sections (rough eraser) and cleaning the sketch (precise eraser), (g) removing the current hand model for a new hand posture to operate the touch screen while holding the drill (melting), (h) trying to make various hand poses (front mirror, palm-pinning widget), (i) finding a desired hand pose (responsive spangles, freezing), (j) checking collision inside the palm (transparent hand, responsive spangles), (k) revising the sketch for the new hand posture (transparent hand, responsive spangles, precise eraser), (l) developing and finishing the design concept by changing both the drill shape and hand posture iteratively.
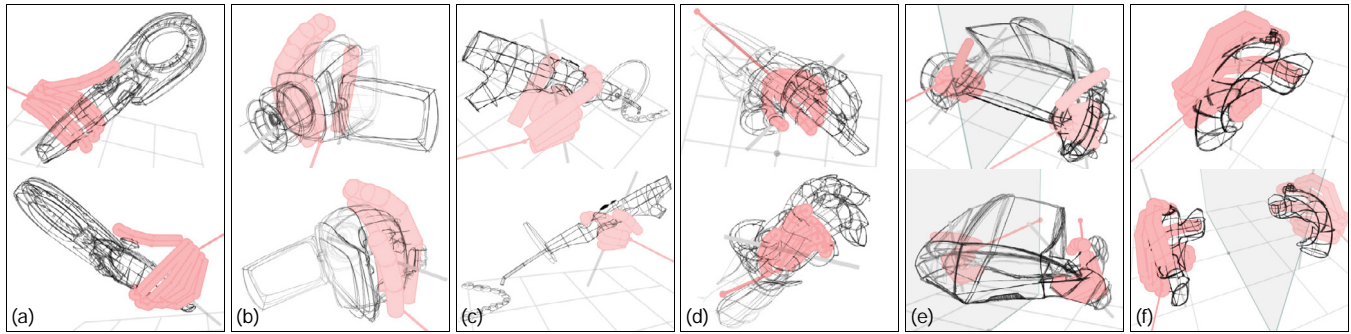
Figure 15. Examples of resulting product concepts: (a) handheld noise detector (P1), (b) camcorder (P2), (c) dental curing light (P7), (d) robotic exoskeleton brace (P2), (e) drone controller (P3), (f) VR controllers (P3).

with a pen stylus, and a laptop PC with Windows 7 64bit OS, Intel Core i7 2.70GHz CPU, and a Nvidia Quadro K2000M GPU. We wrote our program in Java using the Leap Motion SDK and OpenGL.

USER STUDY
In order to see the possibility of applying our system to the real product design practice, we invited ten students (P1-P10) majoring in industrial design and aged 22-29 (7 males and 3 females). All of them were educated with product design skills and were good at using various 3D modeling programs and 2D digital sketching on the display tablet.

Procedure
We brought each participant to our system on a one-on-one basis. One of us gave a 10-minute demonstration showing every function of our system (note: this user study was conducted before implementing responsive spangles and alternative rendering styles of hands) and sketch examples, and then we let participants use our system to create new product concepts with a one-page cheat sheet containing every function and interaction. During the session, participants were allowed to ask the assistant for help and also make suggestions. We asked participants to use our system for at least one hour and complete at least one sketch about hand-related product concepts. Since we considered our system as a tool intended to support free design ideation, we did not give the participants any restriction in terms of the quantity and quality of resulting designs or the time to use our system, except the minimum requirements.

We collected reported problems and suggestions during the test, video recordings of the test, and sketch outcomes. After the test, we interviewed each participant to gather feedback on overall experience, pros and cons, and ideas to improve our system.

Results
All participants could learn functions provided by our system without big problems, apply them in their own design processes, and create various handheld product concepts. Ten test users made 47 sketches in total, and some of them are shown in Figure 15.

ANALYSIS OF SKETCH OUTCOMES
Recalling our research goals of enabling designers to 1) obtain a virtual hand model of a desired posture, 2) sketch the shape of a handheld product along with the captured hand information, 3) develop both the product and hand posture, and 4) get a high-quality design outcome considering the scale and usage of the product, we thus analyzed the hand postures, product shapes, and hand-product relations in the sketches made by the test users, and we made a physical mockup from a concept to check if the design was satisfactory in the real world.

Hand postures well showing product concepts were used.
In all sketch results, we could find natural hand postures that reveal the scale and usage of products. With a hand posture, we could easily identify how to grasp the corresponding product as well as which hand and what each finger should do: a handheld noise detector with a thumb joystick (Figure 15a), a camcorder held by a right hand and controlled by a thumb (Figure 15b), a dental curing light gripped by a right hand for delicate operation (Figure 15c), an exoskeleton brace of which the robotic fingers are articulated by human fingers (Figure 15d), a drone controller with a display carried by both hands (Figure 15e), and a pair of VR controllers (Figure 15f) each of which has a thumb joystick and front buttons and is held by each hand separately.

Product shapes were sketched using hand information.
The skeletal segments and grip axis of a virtual hand were fully utilized for different product characteristics. Some results showed positive implications for using the grip axis to give directionality of products. In the camcorder concept (Figure 15b), the section sketches for the lens and the body are aligned perpendicular to the grip axis. The grips of the noise detector (Figure 15a), drone controller (Figure 15e), and VR controllers (Figure 15f) also show section sketches, along with grip axes. In the sketch of the exoskeleton brace (Figure 15d), the sketched sections from skeletal segments of each finger create a tight shape for its purpose of use.

Interaction between hands and products were well defined.
On many product sketches, we could see interface elements in consideration of both product shapes and finger positions

and/or moving ranges in their use situations. The joystick cap on the noise detector (Figure 15a) and VR controllers (Figure 15f) and recording button on the camcorder (Figure 15b) are placed by considering the position of the thumb. The interaction area on the drone controller (Figure 15e) is within the reachable range of the thumb. On the other hand, the power button of the dental light (Figure 15c) is intentionally placed at a distance from the fingertips to prevent the user from pushing the button by mistake.

Designed product fitted in hand as intended when made.
To make sure that the scale and usage of products in sketches are proper in the real world, we made a physical mockup from a VR controller (Figure 16). We exported 3D point curve data from the sketch model, created a 3D surface model in SolidWorks using the data (without any scaling operation), and rapid prototyped it. We could comfortably hold the mockup with real hands and make the same poses to those with virtual hands in our system. We could also verify that fingers were naturally placed on intended locations on the mockup, where the interface elements (joystick and buttons) were depicted in the sketch.
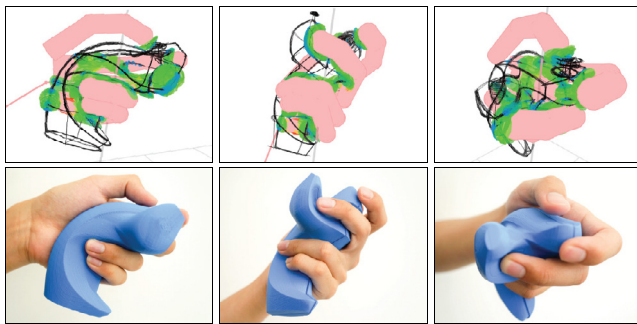

Figure 16. VR controller sketched in our system (Figure 15f) with responsive spangles (first row) and a physical mockup made using 3D modelling and rapid prototyping based on the 3D point curve data from the concept model (second row). Note: all the figures represent the same product held with the same hand posture but are seen from different viewpoints.

USER FEEDBACK AND DISCUSSION
Based on our observation and user feedback during and after the test session, we evaluated our system in a qualitative manner in terms of the characteristics of the new workflow with our system and the usage pattern and effectiveness of each function and/or interaction technique.

They Were Good.
We could find that many of our ideas implemented in our system worked well according to our expectations.

Hands served as a good scale reference in the virtual world.
A natural way of roughly measuring size by hand in the real world was still valid in our system. "*My hand guarantees correct measurement for the concept.*" (P4) "*I started sketching by roughly measuring its (the product concept's) size with my hands.*" (P8)

Hand pose and product concept went hand in hand.
Hand postures took the lead at first, but they handed over the lead to product sketches when sketches were matured to some degree, which often required revised hand postures. The same thing happened repeatedly. "*I was drawing a hand drill for one hand, and suddenly realized it could be good for the other hand, so I added another handle.*" (P2) "*I did not know it when I made the first pose, but I could see the grip was wrong. So I changed my pose.*" (P1) "*I could think of more details like buttons [that] better fit the hand.*" (P10) "*I considered one of shake-hand grip and pen grip. While sketching, I decided to use both.*" (P1)

Different mirrors were used well differently.
The front mirror was not used often, but it was helpful for the first pose. The center mirror was used often, but most users normally set it off to avoid visual clutter when sketching and set it on when checking the whole shape. "*I like the (center) mirror. I can feel the whole object quickly with it.*" (P2) "*I want to move the mirror to other positions to sketch symmetric forms.*" (P9)

Sketch planes created differently worked differently.
Test users were able to set desired planes by applying different numbers of ticks depending on situations. They readily accepted one-tick triggered sketch planes if they were satisfactory at a glance. If not, they did not adjust them. Rather, they applied one more tick to obtain sketch planes with two constraints, and then they rotated the planes. They also accepted three-tick triggered sketch planes if the planes were satisfactory. If not, they just abandoned the planes and did not adjust the planes. Two-tick triggered sketch planes were well accepted and the most popular.

3D sketching could encourage creativity.
Test users unanimously said that they could find creative shapes using 3D sketching, which would not be possible to get otherwise. "*I was able to get new ideas from an infinite number of (unfamiliar) views.*" (P2) "*I had no idea at first, but I could find a way soon to use serendipitous patterns in there.*" (P1) "*I could get ideas from intersection points on a (sketch) plane.*" (P1)

Intensifier showed potential.
Test users easily understood the concept of the intensifier, learned it quickly, and showed a high efficiency. "*I like that similar functions are together.*" (P1) "*They were there where I expected.*" (P1) "*It was quick to go between this (tumble) and this (dolly).*" (P2) "*Visual changes on the screen, like between rough and precise erasers, perfectly follow how the function changes and what I can do in that mode.*" (P7)

They Needed Improvement.
Not all of our ideas were found to be perfect. We could find issues to revisit in the future.

Erasing anything worked but required care.
Users readily accepted the concept of erasing geometric entities and constraints in the same eraser mode. But they occasionally removed something they did not intend to because they only paid attention to what they wanted to remove. "*When I erased curves, I erased a sketch plane.*" (P2) "*I used to erase curves to erase a constraint.*" (P5) Most of them could recognize this problem and avoid it with special care. One possible solution will be understanding user intention with the first-applied element and keeping the intention until the pen is off.

Hands needed different rendering styles sometimes.
Our minimalistic rendering style of hands worked well to show necessary hand information while not taking much attention from products. But some users gave opinions on hand rendering. "*It's hard to draw near the palm because the hand hides my curves.*" (P9) "*Is it possible to remove the hand or make it transparent?*" (P2) "*The bones hinder [me] when I want to anticipate the final image.*" (P3) Using the feedback, we developed alternative rendering styles of hands in the later version of our system shown in Figure 12.

Re-posing hands on complicated sketches was challenging.
Some test users reported that it was not easy to place hands on existing models, especially when they were not of simple shape. "*I don't have a sense of distance to the model. No idea if I succeeded in grabbing it again.*" (P1) "*I would like to see my fingertip touches on the joystick.*" (P1) We took this problem seriously because it might lead designers to use hands as a static template and block our two-way product design process from hand to product, and vice versa. We thus developed the responsive spangles later, as shown in Figure 13. After implementing responsive spangles, we invited some of the test users (P1, P3, P7) to take part in another test session and verified that they could re-pose their hands much easier than before.

CONCLUSION AND FUTURE WORK
Many studies have shown that the combination of computer graphics and sensor technology can improve design. We also looked in the same direction for real design practice. In this work, we first pointed out a critical problem in the product design process and analyzed related design knowledge. We then used the lessons for our decision-making process to systematically develop a new digital design system while paying special attention to high-throughput interaction. Finally, we implemented a fully working system. Test users learned our system with little effort to capture their hand postures and sketch handheld products, thus generating high-quality design results. They all agreed with the importance of our targeted problem and the necessity of our system, and they were greatly pleased with what they created in our system, impressed by what was rapid prototyped from the concept, and eager to use our system for their work. We are sincerely looking forward to seeing our work help designers in the field and shed light on

innovating real design practices by applying cutting-edge technologies.

Future work includes: 1) global/local modification of product sketches by hand motion with a hand tracking sensor; 2) calculation of ergonomic factors when virtually grasping in a hand(s), such as contact area, pressure, the moving range of each finger, and joint torque and muscle load; 3) consideration of various hand sizes for designing universal or customized products as well as products for specific groups of people such as toys for kids; 4) empirical studies on each of the proposed functions and interaction techniques such as double quasi modes using the intensifier button; and 5) long-term user studies in the wild.

APPENDIX: COLOR AND SIZE OF THE SPANGLES
The radius $r(x)$ and color of a spangle are determined by a distance to the closest point of the hand skeleton x (Figure 17a), and its normal vector is set to the point so that the spangle always faces the hand (Figure 17b-c). At the farthest effective distance from the skeleton x_{far}, the spangle starts to appear in blue. In contact with the finger, the spangle reaches maximum radius r_{max} and turns green. From x_{far} to the contact, the spangle size is determined by a sigmoid curve, and its color is interpolated between blue and green. If the distance is smaller than the finger radius R, the spangle retains r_{max} and turns from green to yellow to red until it reaches the nearest effective distance x_{near}. Our function for the radius is

$$r(x) = \begin{cases} 0, & x > x_{far}, \ 0 \leq x < x_{near} \\ r_{max}\{1 - 1/(1 + Ce^{-k(x-R-D)})\}, & R \leq x \leq x_{far} \\ r_{max}, & x_{near} \leq x < R. \end{cases}$$

We set x_{far} to $2R$ and x_{near} to $R/3$. The maximum spangle radius r_{max} we set is $1.2R$ so that a spangle is visible even when its center is inside a finger. Other constants we set for the function were $C = 5.0$, $D = 3.0$, and $k = 1.2$.
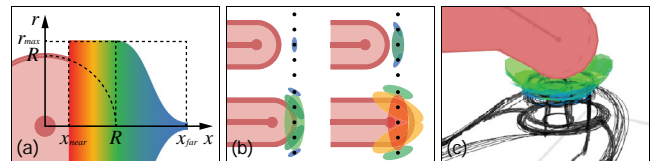

Figure 17. How responsive spangles work: (a) color and size (r) of a spangle by distance from a hand skeleton (x), (b) behaviors of spangles when the finger is approaching sketched strokes (represented as black dots), (c) resulting visualization in action.

REFERENCES

1. Achibet, M., Casiez, G., Lécuyer, A., and Marchal, M. THING: introducing a tablet-based interaction technique for controlling 3D hand models. In *Proc. CHI '15*, ACM (2015), 317-326.

2. Anderson, F., Grossman, T., Matejka, J., and Fitzmaurice, G. YouMove: enhancing movement training with an augmented reality mirror. In *Proc. UIST '13*, ACM (2013), 311-320.

3. Bae, S. H., Balakrishnan, R., and Singh, K. ILoveSketch: as-natural-as-possible sketching system for creating 3D curve models. In *Proc. UIST '08*, ACM (2008), 151-160.

4. Bae, S. H., Balakrishnan, R., and Singh, K. EverybodyLovesSketch: 3D sketching for a broader audience. In *Proc. UIST '09*, ACM (2009), 59-68.

5. Benko, H., Jota, R., and Wilson, A. MirageTable: freehand interaction on a projected augmented reality tabletop. In *Proc. CHI '12*, ACM (2012), 199-208.

6. Cheek, C. *Drawing Hands*. Courier Corporation. 2008.

7. De Araùjo, B. R., Casiez, G., and Jorge, J. A. Mockup Builder: direct 3D modeling on and above the surface in a continuous interaction space. In *Proc. GI '12*, ACM (2012), 173-180.

8. De Paoli, C. and Singh, K. SecondSkin: sketch-based construction of layered 3D models. *ACM Transactions on Graphics*, *34*, 4 (2015).

9. Dorsey, J., Xu, S., Smedresman, G., Rushmeier, H., and McMillan, L. The Mental Canvas: a tool for conceptual architectural design and analysis. In *Proc. Pacific Graphics*, IEEE (2007), 201-210.

10. Eissen, K. and Steur, R. *Sketching: Drawing Techniques for Product Designers*. Bis. 2007.

11. Eissen, K. and Steur, R. *Sketching: The Basics*. Bis. 2011.

12. ElKoura, G. and Singh, K. Handrix: animating the human hand. In *Proc. 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association (2003), 110-119.

13. Ensomobile. Art Model 3 - Hands (Version 1.0). 2009. http://ensomobile.com/home/applications/art-model-3/.

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

15. Gannon, M., Grossman, T., and Fitzmaurice, G. Tactum: a skin-centric approach to digital design and fabrication. In *Proc. CHI '15*, ACM (2015), 1779-1788.

16. Guiard, Y. Asymmetric division of labor in human skilled bimanual action: the kinematic chain as a model. *Journal of motor behavior*, *19*, 4 (1987), 486-517.

17. Hamm, J. *Drawing the head and figure*. Penguin. 1983.

18. Hanks, K. and Belliston, L. *Rapid Viz, Third Edition: A New Method for the Rapid Visualization of Ideas*. Thomson. 2006.

19. Henry, K. *Drawing for Product Designers*. Laurence King Publishing. 2012.

20. Kallio, K. 3D6B editor: projective 3D sketching with line-based rendering. In *Proc. SBIM '05* (2005), 73-79.

21. Karpenko, O., Hughes, J. F., and Raskar, R. Epipolar methods for multi-view sketching. In *Proc. SBIM '04*. Eurographics Association (2004). 167-173.

22. Kazmi, I. K., You, L., and Zhang, J. J. A survey of sketch based modeling systems. In *Computer Graphics, Imaging and Visualization (CGIV)*, IEEE (2014), 27-36.

23. Kry, P. G., Pihuit, A., Bernhardt, A., and Cani, M. P. HandNavigator: hands-on interaction for desktop virtual reality. In *Proc. VRST '08*, ACM (2008), 53-60.

24. Lapides, P., Sharlin, E., Sousa, M. C., and Streit, L. The 3D Tractus: a three-dimensional drawing board. In *Proc. TABLETOP '06*, IEEE (2006), 169-176.

25. Leap Motion. Leap Motion controller. 2016. https://www.leapmotion.com/.

26. Li, Y., Hinckley, K., Guan, Z., and Landay, J. A. Experimental analysis of mode switching techniques in pen-based user interfaces. In *Proc. CHI '05*, ACM (2005), 461-470.

27. Maes, P., Darrell, T., Blumberg, B., and Pentland, A. The ALIVE system: wireless, full-body interaction with autonomous agents. *Multimedia Systems*, *5*, 2 (1997), 105-112.

28. Mitchell, R. W. Kinesthetic-visual matching and the self-concept as explanations of mirror-self-recognition. *Journal for the theory of social behaviour*, *27*, 1 (1997), 17-39.

29. Monroe, M. and McKie, S. Handy (Version 3.1). 2016. http://www.handyarttool.com/.

30. Morikawa, O. and Maesako, T. HyperMirror: toward pleasant-to-use video mediated communication system. In *Proc. CSCW '98*, ACM (1998), 149-158.

31. Olsen, L., Samavati, F. F., Sousa, M. C., and Jorge, J. A. Sketch-based modeling: a survey. *Computers & Graphics*, *33*, 1 (2009), 85-103.

32. Paczkowski, P., Kim, M. H., Morvan, Y., Dorsey, J., Rushmeier, H. E., and O'Sullivan, C. Insitu: sketching architectural designs in context. *ACM Transactions on Graphics*, *30*, 6 (2011).

33. Pitaru, A. Rhonda. 2003. http://rhondaforever.com/.

34. Pixologic. ZBrush (Version 4r7). 2016. https://pixologic.com/.

35. Robertson, S. *Lift Off: Air Vehicles Sketches & Renderings from the Drawthrough Collection*. Design Studio Press. 2006.

36. Sachs, E., Roberts, A., and Stoops, D. 3-Draw: a tool for designing 3D shapes. *IEEE Computer Graphics and Applications*, *11*, 6 (1991), 18-26.

37. Schkolne, S., Pruett, M., and Schröder, P. Surface Drawing: creating organic 3D shapes with the hand and tangible tools. In *Proc. CHI '01*, ACM (2001), 261-268.

38. Schmidt, R., Khan, A., Singh, K., and Kurtenbach, G. Analytic drawing of 3D scaffolds. *ACM Transactions on Graphics*, *28*, 5 (2009).

39. Shao, C., Bousseau, A., Sheffer, A., and Singh, K. CrossShade: shading concept sketches using cross-section curves. *ACM Transactions on Graphics*, *31*, 4 (2012).

40. Sharp, T., Keskin, C., Robertson, D., Taylor, J., Shotton, J., Kim, D., Rhemann, C., Leichter, I., Vinnikov, A., Wei, Y., Freedman, D., Kohli, P., Krupka, E., Fitzgibbon, A., and Izadi, S. Accurate, robust, and flexible real-time hand tracking. In *Proc. CHI '15*, ACM (2015), 3633-3642.

41. Sjölén, K. and MacDonald, A. *Learning Curves: An Inspiring Guide to Improve Your Design Sketch Skills*. KEEOS Design Books. 2011.

42. Tsang, S., Balakrishnan, R., Singh, K., and Ranjan, A. A suggestive interface for image guided 3D sketching. In *Proc. CHI '04*, ACM (2004), 591-598.

43. uMake. uMake (Version 1.0.6). 2016. https://www.umake.xyz/.

44. Wang, R., Paris, S., and Popović, J. 6D hands: markerless hand-tracking for computer aided design. In *Proc. UIST '11*, ACM (2011), 549-558.

45. Wu, Y. and Huang, T. S. Hand modeling, analysis and recognition. *Signal Processing Magazine*, *18*, 3, IEEE (2001), 51-60.

46. Xin, M., Sharlin, E., and Sousa, M. C. Napkin Sketch: handheld mixed reality 3D sketching. In *Proc. VRST '08*, ACM (2008), 223-226.

47. Xu, B., Chang, W., Sheffer, A., Bousseau, A., Mccrae, J., and Singh, K. True2Form: 3D curve networks from 2D sketches via selective regularization. *ACM Transactions on Graphics*, *33*, 4 (2014).

48. Ye, Y. and Liu, C. K. Synthesis of detailed hand manipulations using contact sampling. *ACM Transactions on Graphics*, *31*, 4 (2012).

49. Zhang, Y., Han, T., Ren, Z., Umetani, N., Tong, X., Liu, Y., Shiratori, T., and Cao, X. BodyAvatar: creating freeform 3D avatars using first-person body gestures. In *Proc. UIST '13*, ACM (2013), 387-396.