A Rapid Shape Acquisition Method by Integrating User Touching Input

Yong Chen*, Jinho Jung, Yongqiang Li
Epstein Department of Industrial and Systems Engineering
University of Southern California, Los Angeles, CA 90089

*Corresponding author: yongchen@usc.edu, (213) 740-7829

ABSTRACT
The easiness of creating three-dimensional (3D) models from physical objects is one of the core challenges that remain to be addressed in reverse engineering. In this paper, a touch-based 3D shape acquisition method is presented that is easy and intuitive to use. Based on the method, a user can easily interact with both real and virtual objects and directly generate feature-based CAD models. The key technical challenges on developing the related hardware and software systems are discussed. By using widely available consumer electronic devices, a low-cost prototype system is designed and built. Based on the designed system, a novel 3D coordinate computation method is developed to obtain the touching point positions. Related challenges on using such a system in generating 3D models are also discussed. Multiple examples are presented to illustrate the effectiveness and efficiency of the developed method.

Keywords: Reverse engineering, touch-based system, shape acquisition, augmented reality.

1. INTRODUCTION
Reverse engineering (RE) to convert existing physical objects into digital models has been well developed and widely used [1]. Various three-dimensional (3D) scanning technologies have been developed before including coordinate measuring machines, laser scanner, structured light digitizers, and computed tomography. Figure 1 shows typical data flow based on such reverse engineering technologies. First, a physical object is scanned into point cloud data; polygonal surfaces are then constructed from the densely scanned points. For the purpose of part repair or model re-design, further processing of the constructed polygonal surfaces is required. Additional software modules such as NURBS reconstruction and feature extraction are typically required to generate feature-based computer-aided design (CAD) models, which can then be used in various applications.

![Figure 1: Typical data flow of point-based reverse engineering](image)

While sophisticated software products exist for processing densely sampled data acquired by 3D scanning technologies, the current use of reverse engineering is lengthy and laborious. Users are typically required to have significant knowledge on the scanning technologies and related geometric computation techniques. In this paper, an alternative shape acquisition method based on given physical objects is
presented. Motivated by the popularity of touch-based systems such as *iPhone* and *iPad* from Apple Inc., a touch-based Reverse Engineering system (called *Touch-RE*) has been developed to simplify the shape acquisition process. The data flow of the developed method is shown in Figure 2. By integrating user touching input in the shape acquisition process, feature-based CAD models can be directly reconstructed. In addition, the developed Touch-RE system is easy-to-use and inexpensive. Such a system, if fully developed, may enable the general public including non-technical persons to quickly reconstruct digital models from physical objects that are used in their daily lives.

![Figure 2: Data flow of Touch-based reverse engineering method](image)

2. OUR APPROACH – TOUCH-BASED REVERSE ENGINEERING

In our daily lives, touch and vision are co-located during the exploration of objects. The perception of object shapes is guided by vision. Accordingly it is intuitive for a user to touch the surfaces of a recognized shape. For example, as shown in Figure 3, the object in the real world (left side of the figure) is easily to be recognized as a $Z$-axis aligned cylinder. Hence the user can touch a minimum of three points ($P_1$, $P_2$, $P_3$) on the cylindrical surface to uniquely define such a cylinder in the virtual world. If the touching points ($x_i$, $y_i$, $z_i$) can be quickly and accurately identified, a parameterized CAD model of the cylinder can be easily constructed. The generated CAD model can then be displayed to the user as visual feedback. In addition, since the cylinders in the real and virtual worlds have known correspondence, the visual feedbacks of the real and virtual worlds can be synchronized. For example, if the physical object is rotated along $Z$ axis by an angle $\alpha$, the virtual model can be updated based on the same rotation, and vice versa. Such an integration of both virtual and real environments is called mixed reality [2] or augmented reality [3].

![Figure 3: Illustration of the principle of Touch-RE](image)
Hence the core idea of the touch-RE is to allow the user to see the real and virtual objects in a co-located way, and accordingly create virtual models by touching the real object. Traditional CAD systems require the user to visualize the shape of a virtual object and use standard input devices such as a mouse and a keyboard to define it. In comparison, the touch-RE system allows the user to look at the real 3D object instead of 2D screen in defining its CAD model. Hence, 3D shape indication and visualization become trivially natural and intuitive, which is critical for consumers who have no technical knowledge or previous training on CAD. In addition, the created virtual object are feature-based. They can be used in various applications such as visualization, duplicating, modifying, and rapid prototyping.

As shown in Figure 3, the middle region between the two ends of a continuum, the real world and a totally virtual environment, is called **mixed reality** [3]. Traditional CAD systems are based on totally virtual environment. Extensive researches [4-7] have shown that augmented reality (AR) and augmented virtuality (AV) can be beneficial on the easiness of usage. Our method has adopted the principle of AR by integrating real and virtual objects in the shape acquisition process. Accordingly a reverse engineering system has been developed to assist users in rapidly acquiring 3D models of given physical objects. A simple example to illustrate the developed Touch-RE system is shown in Figure 4. A user first pick a Z-axis aligned cylinder function. He/she then uses a pen to selectively touch the surface of given physical object. The indicated three points on the object (refer to Figure 4.a-c) will consequently enable the Touch-RE system to automatically compute a CAD model related to the Z-axis aligned cylinder including its size and position as shown in Figure 4.(d).

![Illustration of the shape acquisition process for a given cylinder](image)

**Figure 4**: An illustration of the shape acquisition process for a given cylinder

A critical issue to be addressed is how to integrate user’s touching input in a reverse engineering system such that general 3D shapes can be defined easily, quickly, and accurately. In addition, the Touch-RE system should be extremely low cost for a wide adoption by the general public. To achieve such goals, both hardware and software issues have been investigated in the paper. Details of the developed method are organized as follows. Section 3 describes the selection of hardware components and their setup in the Touch-RE system. Section 4 presents a 3D coordinate computation method of a touching point based on multiple 2D coordinates captured from different views. Section 5 describes the data flow of the system and related software modules. Section 6 explains the model generation and alignment methods. Section 7 presents four examples to illustrate the capability of the Touch-RE system. Finally Section 8 concludes the paper.

3. HARDWARE OF THE TOUCH-RE SYSTEM

Our design decisions on hardware selection and system layout for integrating touching input in 3D shape acquisition are introduced in this section.

3.1. Technology Selection

Various touching devices based on different technologies have been developed before.

1. 3D measuring arm such as FAROARM by *FARO Technologies Inc.* are widely used in reverse engineering. It is a technology based on inverse kinematics. A set of joints are used to link a touching probe to achieve desired freedom. By measuring current angles of joints, the 3D
coordinates of the contact point can be accurately computed (e.g. 0.02mm accuracy). However 3D measuring arm is expensive with a price range of $10K ~ $100K.

(2) Haptic devices such as PHANTOM Omni device from Sensable Technologies can provide 6 degree of freedom tracking and force feedback. They have been widely used in interactive 3D carving and sketching. However, the motion region is usually limited.

(3) Based on triangulation method, two cameras can capture the intersection point of an object surface and a ray shot by a laser. This approach has also been widely used in reverse engineering. However, controlling a laser from a distance to accurately indicate a touching point is rather difficult.

(4) Infra Red (IR) LED and related IR-sensor is a method that has recently becoming popular in human-computer interaction. One of the cheapest IR signal receiver is a Wii controller from Nintendo as a game input device. Lee [8] suggested a novel way of hooking Wii controller’s Bluetooth signal with personal computer (PC). Based on the work, Lee provided a 2D drawing application based on one Wii controller and one IR-LED pen. Accordingly Hay et al. [9] showed that two Wii controllers can be used for getting a 3D point’s coordinate with good accuracy (an average error of 2.46mm with a standard deviation of 2.23mm has been shown).

Table 1: Comparison of different technologies (sections in blue indicate dominate characteristics)

<table>
<thead>
<tr>
<th>Properties</th>
<th>3D arm</th>
<th>Haptic</th>
<th>Laser &amp; camera</th>
<th>IR LED &amp; sensor</th>
<th>Decision priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total cost</td>
<td>High</td>
<td>Medium</td>
<td>Low</td>
<td>Low</td>
<td>1</td>
</tr>
<tr>
<td>Accuracy</td>
<td>High</td>
<td>High</td>
<td>Low</td>
<td>Moderate</td>
<td>2</td>
</tr>
<tr>
<td>Hand freedom</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
<td>High</td>
<td>3</td>
</tr>
<tr>
<td>Device size</td>
<td>Big</td>
<td>Medium</td>
<td>Small</td>
<td>Small</td>
<td>4</td>
</tr>
<tr>
<td>Operation region</td>
<td>Large</td>
<td>Small</td>
<td>Medium</td>
<td>Medium</td>
<td>5</td>
</tr>
<tr>
<td># of working points</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>Calibration needs</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>7</td>
</tr>
</tbody>
</table>

A desired Touch-RE system would enable users to provide touching input easily, quickly, and accurately. In addition, the touching device needs to be low cost, light weight, and easy to use. A comparison of the discussed technologies with our decision priority is shown in Table 1. Based on the comparison, IR LED and related IR-sensor are selected. The technology has good properties for the Touch-RE system such as low cost, easy for hand to move, small device size, and reasonable accuracy.

3.2. Hardware Development

Based on the well-known triangulation method, a 3D point coordinate can be computed based on the input of two or more Wii controllers. Since a user may block some of the views when he/she interacts with a real object, four Wii controllers are mounted from various angles to provide sufficient redundancy such that user’s interaction will not be constrained. Further efforts are made in the layout design to ensure at least two Wii controllers will provide input for a touching point. Due to the unavoidable blocking by an object itself, all the four Wii controllers were mounted in one side of the object. A rotating table is integrated in the system to provide an all-around view of the object.

A commercial Wii controller can sense a maximum of 745mm × 580mm rectangle area when the target is 1000mm away. This means that a Wii controller can sense horizontally 40.88° and vertically 32.36°. Since a Wii controller returns more accurate coordinate information from a closer distance, the Wii controllers in our system are put as close as possible to the working volume as long as the sufficient coverage of the volume is ensured. For a working volume of 100mm × 100mm × 100mm, the Wii controllers are put ~310mm away from the center of the working volume, which can provide a margin of ~35mm around the volume.

The hardware setup of the Touch-RE system is shown in Figure 5. The three major hardware components of the system are explained as follows.
Wii controllers: Wii controllers have been developed for game market. They are widely available at rather low price (~$25). The device has a built-in infra-red sensor that can detect up to 4 different IR-LED signals. Every time a Wii controller senses IR signals, it detects their positions and radiates Bluetooth signal. If a PC has a Bluetooth receiver, the signal from the Wii controller can be recorded and further decoded by using a Wiimote library. Currently C, C++, and C# version of Wiimote libraries are widely available. The decoded 2D position has coordinate ranges of 0 ~ 1023 in X-axis and 0 ~ 767 in Y-axis.

Rotating table with a servo motor and driver: One problem of the developed system is that it only allows touching objects from one side. By using a rotating table, the object can be rotated with a precise angle to enable touching input from another side. For example, to touch a portion of an object that is blocked by itself, the object can be rotated along the Z-axis by 180 degrees. Since the rotation information has been known, the Touch-RE system can ensure that the next input, compared to the previous one, has been rotated 180 degrees along Z-axis. To ensure the repeatability of the rotation, a HS-985MG servo motor and a Phidget’s AdvancedServo driver were used. A rotating table is mounted on top of the servo motor with a set of gears to achieve sufficient torque.

IR-LED pen: An IR-LED pen has an infrared LED on its tip (wavelength of 940nm). The LED has a viewing angle of ±25°. The pen has a switch on its body; hence it can be easily turned on/off as needed. The location of the IR LED that will be identified by a Wii controller is the ‘hot-spot’, the place with the strongest light intensity. Due to the viewing angle, such a position is the touching point of the IR-LED on the physical object. Up to four IR-LED pens can be simultaneously used to enable two-hand action.

3.3. Hardware Verification

The prototype system has been examined to ensure the operation region and the coverage of the Wii controllers. As shown in Figure 6, Wii #1 and #4 are used for covering the left and right side of the object respectively; while Wii #2 and #3 are used to track touching on the top and bottom side of the object. Figure 6 shows different views of the system based on a cup. When touching the cup in the front-middle view, all the 4 Wii controllers can sense the location of the IR-LED. For the rest cases, IR-LED points can be sensed by at least 3 Wii controllers. To ensure 100mm cubic space is fully covered, a cube with the same size is used in verifying that all the eight corners can be detected by at least two Wii controllers.
4. ACQUIRING 3D COORDINATE OF A TOUCHING POINT

A Wii controller can sense an IR light and send signal to a Bluetooth adapter at a refresh rate of 100Hz. As shown in Figure 7, an integrated Wiimote library can decode the signal and generate 2D coordinate value \((x, y)\) for each Wii controller. No signal will be sent if there is no IR light in the viewing field of the Wii controller. Hence, when a user switches on an IR-LED pen to indicate a touching point \(P\), all or some of 2D coordinates from the 4 Wii controllers will be captured depending on their viewing angles. Accordingly, a key problem in the Touch-RE system is how to accurately compute the coordinate \((x, y, z)\) of \(P\) from a set of 2D inputs \((x_i, y_i)\), \(2 \leq i \leq 4\). In addition, the 3D point computation needs to be fast (less than 10 millisecond) in order to capture the real-time movement of the IR-LED pen.

4.1. Principle on Computing 3D Coordinates from 2D Inputs

Triangulation in computer vision has been well developed for determining a point in 3D space given its projections onto two or more images. As a Wii controller is essentially an IR camera, the focal point of each camera needs to be known in order to perform triangulation. There have been extensive researches on camera calibration. For example, Zhang [10] introduced a camera calibration method for calibrating a camera easily. To use Zhang’s method, the relationship of two cameras (extrinsic matrix) and each camera’s characteristics (intrinsic matrix) are required. Hay et al. [9] implemented a Wii-based 3D pointing application by using Zhang’s method and related camera calibration toolbox. Based on the calibrated intrinsic and extrinsic matrices, the triangulation method can then be used for converting 2D input into 3D coordinates.

Even though the triangulation method is well established and general, it does not consider the unique properties of the Touch-RE system including:

1. Each Wii controller in the system can send its 2D input to the application. Hence it is usual to have inputs from 3 or 4 Wii controllers. However, in the aforementioned method, only two calibrated cameras are considered. Hence, for the input of three or more cameras, the results related to each pair of the cameras need to be computed. Accordingly a strategy such as computing the average value needs to be determined to generate the final result. In comparison, it is more desired to have a method that can directly incorporate the inputs from multiple cameras to achieve a better accuracy.
Since the positions and orientations of all the four Wii controllers in our system are pre-defined and fixed, their calibration only needs to be done once. Hence a computing method with better accuracy is preferred even if the calibration procedure will take longer time and require more efforts.

Based on the considerations, a 3D coordinate computation method based on machine learning has been developed in the Touch-RE system. The test results indicate that, compared to the aforementioned camera calibration method [9], the approach is intuitive and extremely fast (< 1 millisecond). It also achieves a reasonable accuracy (< 1 mm) within the working volume of the system.

### 4.2. 3D Coordinate Computing Method based on Machine Learning

The basic idea of the machine learning approach is straightforward including two main steps:

1. **Calibration**: Build an extensive database on the relations between pre-defined sampling points \((x, y, z)\) and related 2D inputs \((x_1, y_1)\), \((x_2, y_2)\), \((x_3, y_3)\), and \((x_4, y_4)\) from the four Wii controllers;
2. **Computation**: Based on the built database and identified patterns, compute an unknown 3D coordinate \((x, y, z)\) from any input of Wii controllers \((x_i, y_i)\), \(i = 1 - 4\).

The two steps are discussed in more details as follows.

1. **Building a database by uniformly sampling the working volume**

   As discussed in Section 3, the working volume of the prototype system is a cubic of 100mm \(\times\) 100mm \(\times\) 100mm. It is assumed that all the touching input is inside this volume. Due to the constraint, a 3-axis linear slide (accuracy 0.076mm from Velmax Inc.) was used in moving an IR LED inside the working volume. The hardware setup is shown in Figure 8. Note 3D coordinates \((x, y, z)\) are known based on the motion of the XYZ linear slide. When the IR-LED is turned on by a microcontroller, a set of 2D points \((x_1, y_1)\), \((x_2, y_2)\), \((x_3, y_3)\), and \((x_4, y_4)\) can be recorded from the 4 Wii controllers.

   In the test the working volume was sampled by an increment of 5mm in \(X\), \(Y\), and \(Z\) axis. Hence a total 8000 \((20 \times 20 \times 20)\) samples has been recorded. For the validation purpose, another 1000 \((10 \times 10 \times 10)\) samples were taken by shifting the IR-LED 2.5mm in \(X\), \(Y\), and \(Z\) axis and sampling the volume by an increment of 10mm. Table 2 shows examples of the collected calibration data. The above calibration process takes ~8 hours to perform, which is a drawback of the machine learning approach. In addition, the same procedure needs to be repeated if any settings of the system have been changed.

2. **Computing 3D coordinate by using a regression method**

### Table 2: Example of collected samples

<table>
<thead>
<tr>
<th>Coordinate index</th>
<th>Real coordinate ((X, Y, Z)) (mm)</th>
<th>Collected data from Wii ((x_1,y_1,x_2,y_2,x_3,y_3,x_4,y_4))</th>
</tr>
</thead>
</table>

(2) Computing 3D coordinate by using a regression method
Based on the collected data, various machine learning techniques can be used in capturing characteristics of interest and establishing their relations with observed variables. Our data analysis indicates that the differences between two neighboring layers are small while the differences between the 1st and 20th layers are much larger. In addition, \((x_i, y_i)\) within the same layer are changing gradually. Accordingly a two-step fitting approach has been developed for computing a 3D coordinate from a set of 2D inputs. The approach includes:

(a) Rough estimation: An estimate coordinate \((X', Y', Z')\) is first computed based on all the sampling points in the working volume. Although many machine learning approaches can be used here, a simple approach based on a linear regression model is illustrated as follows. Suppose \(\Phi\) is a matrix with all the collected sample data:

\[
\Phi = \begin{bmatrix}
1 & x_1 & y_1 & z_1 & x_2 & y_2 & z_2 & \cdots & x_{20} & y_{20} & z_{20} \\
1 & x_1 & y_1 & z_1 & x_2 & y_2 & z_2 & \cdots & x_{20} & y_{20} & z_{20} \\
1 & x_1 & y_1 & z_1 & x_2 & y_2 & z_2 & \cdots & x_{20} & y_{20} & z_{20} \\
\end{bmatrix}
\]

And \(\Gamma\) matrix is the related 3D coordinate value of IR-LED in \(X\)-, \(Y\)-, or \(Z\)-axis:

\[
\Gamma = \begin{bmatrix}
X^1 \\
X^2 \\
\vdots \\
X^{100} \\
Y^1 \\
Y^2 \\
\vdots \\
Y^{100} \\
Z^1 \\
Z^2 \\
\vdots \\
Z^{100} \\
\end{bmatrix}
\]

Hence a set of weight values can be computed as:

\[
\beta_X = (\Phi^T \Phi)^{-1} \Phi^T \Gamma_X, \beta_Y = (\Phi^T \Phi)^{-1} \Phi^T \Gamma_Y, \beta_Z = (\Phi^T \Phi)^{-1} \Phi^T \Gamma_Z.
\]

The weight values \(\beta\) is a \(9\times1\) vector \((\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7, \beta_8, \beta_9)^T\). For any given 2D point input \(\eta = (1, x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4)^T\),

\[
X' = \eta \times \beta_X, Y' = \eta \times \beta_Y, Z' = \eta \times \beta_Z.
\]

(b) Refined estimation: Based on the estimated coordinate \((X', Y', Z')\) of \(P'\), the sampling points in the neighboring region of \(P'\) can be used in refining the touching point estimation. Since the sampling points in the neighboring region have smaller variations, a better regression model can generally be achieved. Among many strategies in picking such neighboring regions, a straightforward approach is illustrated as follows. The approach is based on identifying the closest \(XY\), \(XZ\), and \(YZ\) layers to the estimated touching point. For example, in order to refine the \(Y\) value of \(P'\), the \(XZ\) layer whose \(Y\) value is the closest to \(Y'\) is first computed. Accordingly the 400 sampling points in the computed \(XZ\) layer can be used in fitting a better regression model. For example, in the approach based on a linear regression model, matrix \(\Phi\) and vector \(\Gamma_Y\) can be updated as:

\[
\Phi = \begin{bmatrix}
1 & x_{400}^1 & y_{400}^1 & x_{400}^2 & y_{400}^2 & \cdots & x_{400}^{400} & y_{400}^{400} \\
1 & x_{400}^1 & y_{400}^1 & x_{400}^2 & y_{400}^2 & \cdots & x_{400}^{400} & y_{400}^{400} \\
\end{bmatrix}
\quad \text{and} \quad
\Gamma_Y = \begin{bmatrix}
Y_1^{400} \\
Y_2^{400} \\
\vdots \\
Y_{400}^{400} \\
\end{bmatrix}
\]

Hence, \(\beta_Y = (\Phi^T \Phi)^{-1} \Phi^T \Gamma_Y\), and \(Y' = \eta \times \beta_Y\). Similarly, to refine the \(X\) and \(Z\) values of the estimated touching point, the closest \(YZ\) and \(XY\) layers can be computed respectively. A refined coordinate \((X, Y, Z)\) can then be returned as the final 3D coordinate of the touching point.

Discussion:

(1) The computing process based on inputs of four Wii controllers has been illustrated. The presented 3D coordinate computing approach can easily be extended to other cases with only two or three Wii controller inputs. For example, suppose only Wii controllers \#1, \#2, and \#4 have input. Hence,
Accordingly, \( \Phi \) matrix can be updated by deleting \( x_3 \) and \( y_3 \) columns. Hence a matrix \( \Phi \) is generated with a size of \( 8000 \times 7 \) in rough estimation and \( 400 \times 7 \) in refined estimation. Accordingly the computed weight values \( \beta \) would be a \( 7 \times 1 \) vector. Hence the 3D coordinate based on the three Wii controllers is

\[
(X \ Y \ Z) = (\eta \times \beta_x \ \eta \times \beta_y \ \eta \times \beta_z).
\]

(2) Based on the presented method, the 3D coordinate of a touching point can be computed real-time based on any given 2D coordinates \( \eta \). Note all the weight values \( \beta \) for different combinations of Wii controllers and different layers can be computed offline. The pre-calculated weight values can then be indexed and saved in a matrix. Hence the computation of a coordinate \((X, Y, Z)\) only requires two matrix multiplications, which is trivially fast.

(3) A method based on linear regression models is presented here as an example. Calibration of the built prototype system has been performed. The computed linear regression models based on the calibration results have been analyzed. Although R-Square of the computed models is reasonable (0.998, 0.997, and 0.999 in \( X \), \( Y \) and \( Z \) axes respectively), multiple interacted terms have been identified to be significant. Refined linear regression models by incorporating the identified interacted terms show \(~15\%\) improvements on average computing errors; however, the computation process and data representation become much more complicated. In addition to nonlinear regression models, methods based on more advanced machine learning techniques can also be developed and incorporated in the touch-RE system.

4.3. Test Result

The coordinate \((X, Y, Z)\) of the 8,000 collected samples and 1,000 verification samples have been computed. For each sample point, the distance between the computed and exact coordinates is calculated as its error. For different combination of the 4 Wii controllers, the errors at all the 9,000 sampling points have been computed. The resulted statistics of the computed errors for different input of Wii controllers are shown in Tables 3 and 4. The results illustrate that better accuracy can be achieved based on the input of more Wii controllers. Within the given working volume, an estimated 3D point based on the linear regression models has an error that is generally less than 1mm. As shown in the tables, the cases in which the average error or standard deviation exceed 1mm are highlighted in red. The result is shown to be more accurate than the one presented in [9] (an average error of 2.46mm with a standard deviation of 2.23mm) although our approach only considers a much smaller working volume. As discussed before, the accuracy of the computed 3D coordinates may be further improved if more complex regression models are used.

<table>
<thead>
<tr>
<th>Errors (Wii: #1, #2)</th>
<th>Errors (Wii: #2, #3)</th>
<th>Errors (Wii: #3, #4)</th>
<th>Errors (Wii: #1, #3)</th>
<th>Errors (Wii: #1, #4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg.</td>
<td>Std</td>
<td>Avg.</td>
<td>Std</td>
<td>Avg.</td>
</tr>
<tr>
<td>( X )</td>
<td>2.1420</td>
<td>1.9675</td>
<td>2.2086</td>
<td>1.8278</td>
</tr>
<tr>
<td>( Y )</td>
<td>0.8220</td>
<td>0.5204</td>
<td>0.4526</td>
<td>0.4982</td>
</tr>
<tr>
<td>( Z )</td>
<td>0.2896</td>
<td>0.4699</td>
<td>0.2896</td>
<td>0.4699</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Errors (Wii: #1, #2, #3)</th>
<th>Errors (Wii: #1, #2, #4)</th>
<th>Errors (Wii: #1, #3, #4)</th>
<th>Errors (Wii: #2, #3, #4)</th>
<th>Errors (Wii: #1, #2, #3, #4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg.</td>
<td>Std</td>
<td>Avg.</td>
<td>Std</td>
<td>Avg.</td>
</tr>
<tr>
<td>( X )</td>
<td>1.3471</td>
<td>1.1428</td>
<td>0.2202</td>
<td>0.2409</td>
</tr>
<tr>
<td>( Y )</td>
<td>0.7940</td>
<td>0.1061</td>
<td>0.8050</td>
<td>0.4778</td>
</tr>
<tr>
<td>( Z )</td>
<td>0.2975</td>
<td>0.2010</td>
<td>0.3995</td>
<td>0.3293</td>
</tr>
</tbody>
</table>

The histograms of the error distribution based on the four Wii controllers are also shown in Figure 9. It can be seen that, for the 9000 samples in the working volume, most errors are within 0 ~ 0.25mm range. The test results also indicate that, for the best accuracy, a user should ensure a touching point can be seen by three or four Wii controllers. In addition, the viewing angles of Wii controllers 2 and 3 are too close in our system; hence the results based on them are inferior to those based on Wii controllers 1 and 4.
The real-time computation of 3D coordinates based on 2D coordinates \( \eta \) has also been tested. In the test, the \( X\!\!Z \) linear slide as shown in Figure 8 was used in moving an IR LED in various directions. During the movement, a micro-controller is used in turning the IR LED on and off. The 3D coordinates based on the input of four Wii controllers are dynamically computed. The computed 3D points for some tested linear motions are shown in Figure 10. The error statistics of such randomly sampled points is given in Table 5.

### Table 5: Error distance of computed linear motions based on four Wii controllers

<table>
<thead>
<tr>
<th>Errors (mm)</th>
<th>( X_1 )</th>
<th>( X_2 )</th>
<th>( Y_1 )</th>
<th>( Y_2 )</th>
<th>( Z )</th>
<th>( XY )</th>
<th>( YZ )</th>
<th>( XZ )</th>
<th>( XYZ )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg.</td>
<td>0.22</td>
<td>0.17</td>
<td>0.31</td>
<td>0.19</td>
<td>0.20</td>
<td>0.23</td>
<td>0.28</td>
<td>0.23</td>
<td>0.66</td>
</tr>
<tr>
<td>Std.</td>
<td>0.16</td>
<td>0.11</td>
<td>0.20</td>
<td>0.10</td>
<td>0.16</td>
<td>0.13</td>
<td>0.18</td>
<td>0.12</td>
<td>0.57</td>
</tr>
</tbody>
</table>

### 5. SOFTWARE OF THE TOUCH-RE SYSTEM

Several software libraries and drivers have been integrated in the Touch-RE system. As shown in Figure 11, a CAD model can be created based on the selected geometric elements and related touching input. The graphical user interface (GUI) of the system is shown in Figure 11. A set of 2D and 3D geometric elements have been incorporated. In addition, two other software modules, rotating table controller and orientation aligner (refer to Section 6.2), can update the transformation matrix related to the constructed 3D model. Accordingly the display of the CAD model will be dynamically updated to ensure the real and virtual objects are synchronized with each other.
5.1. Wiimote Library

BlueSoleil software (www.bluesoleil.com) was used in receiving Bluetooth radio signal sent by Wii controllers. To decode the signal from a Wii controller, a C++ base Wiimote library called ‘Wiiyourself’ [11] is used. The library converts device’s signal into a pre-defined C++ class data structure. WDK (Windows Driver Kit) was adopted for supporting Wiiyourself library. Hence our application can dynamically receive the x and y coordinates of each Wii controller. The library’s capability on supporting multiple Wii controllers has been tested. Our experiment indicated that the library can support and accept Bluetooth signal from all the four Wii controllers simultaneously.

5.2. Open-Source CAD application: Solidgraph

In the developed prototyping system Solidgraph CAD software is selected as the shape modeling tool. Developed by Geometros geometrical systems, Solidgraph is an open-source system for 3D modeling of complex geometric objects [12]. The application supports primitives such as point, line, circle, and spline. It also supports geometric operations such as Booleans and other Kinematic operations. The open-source application is available from: http://www.codeproject.com/KB/applications/solidgraph.aspx. All the drawing functions have been modified such that a CAD model can be created directly based on 3D touching input.

5.3. Rotating Table Controller

An individual application, rotating table controller, has been developed based on Phidget libraries in C language. The controller has an intuitive interface to allow users to freely rotate the real object in both directions (clockwise or counterclockwise). The desired motion is converted into a set of commands defined by Phidget’s Application Program Interface (API) library. The motor movement commands are then sent via a USB cable, which are executed by a servo-motor controller board. Based on the recorded rotation angle, the transformation matrix of the CAD model can be dynamically updated. At the same time, the rotating table controller can also receive rotation request from the Solidgraph application.
Hence, when the CAD model in Solidgraph application has been rotated in Z axis by the user, the rotating table will also rotated the real object by the same angle.

6. 3D MODEL GENERATION BY THE TOUCH-RE SYSTEM

The use of the presented Touch-RE system in generating 3D models is discussed as follows. The alignment of the real and virtual objects is also discussed since the real object may be repositioned during the shape acquisition process.

6.1. 3D Shape Definition in Touch-RE

Five 2D primitives and eight 3D primitives have been implemented in Solidgraph application. Table 6 shows their definition and possible scenarios in creating digital models by using touching input. As shown in the table, almost all the primitives can be constructed by no more than three touches on the object. In the table, ‘Location of points’ means the minimum information to build models and ‘Optional point’ means more accurate shape can be built by providing more points. For example, two points at a minimum (i.e. left and right points of the circle) can be used in defining a circle in a Z plane. For more general cases, more point on the circle can be defined as touching input. Accordingly our application can fit the circle shape based on the input sampling points.

Table 6: Definition of primitives and their usage scenarios based on touching input

<table>
<thead>
<tr>
<th>Primitive name</th>
<th># of points</th>
<th>Dimension</th>
<th>Location of points</th>
<th>Optional point</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point</td>
<td>1</td>
<td>2D</td>
<td>Start</td>
<td>-</td>
</tr>
<tr>
<td>Line</td>
<td>2</td>
<td>2D</td>
<td>Start, end</td>
<td>-</td>
</tr>
<tr>
<td>Circle</td>
<td>2</td>
<td>2D</td>
<td>Left, Right</td>
<td>Yes</td>
</tr>
<tr>
<td>Arc</td>
<td>3</td>
<td>2D</td>
<td>Start, end, radius</td>
<td>-</td>
</tr>
<tr>
<td>Spline</td>
<td>Many</td>
<td>2D</td>
<td>Points as much user wants</td>
<td>-</td>
</tr>
<tr>
<td>Box</td>
<td>2</td>
<td>3D</td>
<td>Bottom left, top right</td>
<td>-</td>
</tr>
<tr>
<td>Sphere</td>
<td>2</td>
<td>3D</td>
<td>Middle left, middle right</td>
<td>Yes</td>
</tr>
<tr>
<td>Cylinder</td>
<td>3</td>
<td>3D</td>
<td>Bottom left, bottom right, top left</td>
<td>Yes</td>
</tr>
<tr>
<td>Cone</td>
<td>4</td>
<td>3D</td>
<td>Bottom left, bottom right, top left, top right</td>
<td>Yes</td>
</tr>
<tr>
<td>Torus</td>
<td>3</td>
<td>3D</td>
<td>Center, outside point, inside point</td>
<td>Yes</td>
</tr>
<tr>
<td>Ellipsoid</td>
<td>5</td>
<td>3D</td>
<td>Left, right, top, bottom, height</td>
<td>Yes</td>
</tr>
<tr>
<td>Spherical band</td>
<td>4</td>
<td>3D</td>
<td>Middle left, middle right, top, down</td>
<td>Yes</td>
</tr>
<tr>
<td>Extrude Op.</td>
<td>1</td>
<td>2D, 3D</td>
<td>End</td>
<td>-</td>
</tr>
</tbody>
</table>

6.2. Alignment between Various Orientations

As discussed in Section 2, both real and virtual objects co-existed in the Touch-RE system. Their correlated positions are known to us. Even though the rotating table provides limited motion, the user may still need to flip or re-orient the real object, e.g. to access a face in the bottom side of the object. Obviously such operation will break the established spatial alignment between the real and virtual objects. To re-align the virtual and real objects, a set of correlated positions need to be identified in both objects. Accordingly a transformation matrix can be computed based on the identified positions. In addition, such positions should be easy to identify and can be precisely touched by the user. Two types of alignment techniques have been developed as follows.

(1) Corner method

The most easily identified and touched positions are corners. As shown in Figure 12, the correspondence of four or more corners can be established by the user. Accordingly, a homogeneous matrix $H$ (4×4) with both translation and rotation information can be computed for such a rigid object:

Original object $\times H =$ Moved object

Assume the user indicates 4 points ($O_1, O_2, O_3, O_4$) in the original virtual model and related points ($N_1, N_2, N_3, N_4$) in the re-positioned real object. Accordingly,
Hence the virtual model can be realigned by applying the homogeneous matrix \( H \). Since the 3D coordinates \((N_1, N_2, N_3, N_4)\) may have input errors, the system also allows an input of more than four touching points. For example, if six points \((O_1, O_2, O_3, O_4, O_5, O_6)\) and related \((N_1, N_2, N_3, N_4, N_5, N_6)\) are given, a homogeneous matrix \( H \) can be computed based on the pseudo-inverse method as:

\[
H = \begin{bmatrix}
N_{i1} & N_{i2} & N_{i3} & N_{i4} \\
N_{i1} & N_{i2} & N_{i3} & N_{i4} \\
N_{i1} & N_{i2} & N_{i3} & N_{i4} \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1
\end{bmatrix}
\begin{bmatrix}
O_{i1} & O_{i2} & O_{i3} & O_{i4} \\
O_{i1} & O_{i2} & O_{i3} & O_{i4} \\
O_{i1} & O_{i2} & O_{i3} & O_{i4} \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1
\end{bmatrix}^{-1}
\]

\( (6.2) \)

Figure 12: Example of vector method

Figure 13: Example of vector alignment

(2) Vector method

Another type of correlated positions is edges, which can be easily identified and touched. The vector method is based on indicating two vectors related to two edges in the model. As shown in Figure 13, if the positions of two vectors \( \mathbf{ab} \) and \( \mathbf{ac} \) in the original virtual model and the re-positioned real object are given, a homogeneous matrix \( H \) can be computed. In many cases, it is found that defining two vectors is much faster than defining multiple points. One main reason is that defining four or more points compared to defining two vectors usually requires more rotation and interaction with the system.

7. TEST EXAMPLES

Four test examples are presented to demonstrate the capability of the Touch-RE system. The first two examples are based on common objects taken from our daily lives. The first example uses an eye drop bottle, which is relative simple, to illustrate the core idea of the shape acquisition method. The second example uses a toy car to illustrate the modeling process of multiple components. Such a process will require the alignment of an object between different orientations, as well as the positioning of different objects relative to each other. The last two examples are based on common engineering parts. The third example uses a car seat to illustrate the acquiring of multiple 3D curves. The fourth example uses a mechanical component to illustrate the construction of a curved surface.

7.1. Test 1: Acquiring the 3D Model of an Eye Drop Bottle

An eye drop bottle as shown in Figure 14 consists two cylinders and one cone. The model is relatively simple and can be built without table rotation. The building process is shown in Figure 14.(a)-(c). In Figure 14.(a), a cylinder can be constructed based on three touched points \((P_1 - P_3)\); in Figure 14.(b), another cylinder is constructed in a similar way; finally, a cone with a smaller size is constructed by touching 4 points \((P_7 – P_{10})\). Notice the relative position of the three features are determined by the touching input. The generated CAD models indicate that the prototype system can capture the 3D coordinates defined by the IR-LED pen in a satisfactory accuracy. The shape acquisition process also illustrates the easiness of the touch-RE system.
(a) Draw cylinder using three points (left-bottom, right-bottom, and left-top);

(b) Draw another cylinder using three points (left-bottom, right-bottom, and left-top);

(c) Draw cone using four points (left-bottom, right-bottom, left-top, and right-top).

---

**Figure 14:** The shape acquisition process of an eye drop bottle with multiple features

### 7.2. Test 2: Acquiring 3D Models of a Toy Car

The 3D models of a toy car with multiple components are reconstructed from the physical objects. The shape acquisition process requires the use of various geometric primitives, the Boolean and extrusion operations, the object rotation by using the rotating table, and the object alignment between various orientations. The building processes of all the components of the toy car are shown in Figure 15.(a) – (e). The use of the Touch-RE system for each geometric operation is similar to the ones as shown in Figure 14. In addition, the alignment of individually constructed CAD models is shown in Figure 15.(c).

(a) Draw wheels (spherical band) using four points (left-bottom, right-bottom, left-top, and right top);

(b) Draw two boxes using three points (left-bottom, right-bottom, and left-top);

(c) Draw round shape using contour and extrude operation. After this, the new shape is booleaned with the box. Then the wheel are aligned with the booleaned shape;
(d) Draw three cylinders using three points (left-bottom, right-bottom, and left-top);

(e) Draw half-cylinder using contouring and extruding operations;

(f) Generate two more boxes and put the half-cylinder to the right position. Finally a comparison of the acquired models (left) and the real objects (right).

**Figure 15:** The shape acquisition process of a toy car with multiple components

7.3. Test 3: Acquiring 3D Curves on a Seat Object

The test demonstrates the capability of the Touch-RE system on acquiring 3D curves. A scaled down version of a car seat as shown in Figure 16.(a) was used in the test. The object was built by the Stereolithography Apparatus (SLA) process. Due to the translucent material, its reconstruction based on the traditional laser scanning technology is problematic. To define the seat surface, five curves as shown in the figure (curves 1-5) are critical in defining the digital model. By using the Touch-RE system, a user can select an “Open Curve” function and use an IR-LED pen to selectively touch 5-6 sampling points on each curve. Accordingly the 3D coordinates of the touching points are captured as shown in Figure 16.(b). A parametric model of the curves can then be computed based on the sampled points. Figure 16.(c) shows the computed curves based on the cubic B-spline representation [13].

**Figure 16:** The shape acquisition process of multiple 3D curves on the surface of a seat

7.4. Test 4: Acquiring a 3D Surface of a Mechanical Part

A mechanical part with a curved surface is shown in Figure 17.(a). To acquire such a curved surface, a normal pen was used to mark the surface with a set of uniformly distributed curves. A “Spline Surface” function was then selected in the Touch-RE system. Accordingly the point numbers in u and v
directions were specified for defining the curved surface (e.g. 5×6 points in the test). User can then use an IR-LED pen to touch a set of points on the surface based on the marked lines. The related point positions are captured as shown in Figure 17.(c). Based on them, a B-Spline surface can be computed as shown in Figure 17.(d).

![Figure 17](image)

**Figure 17:** The shape acquisition process of a curved surface on a metal part

8. CONCLUSION

In this paper a novel shape acquisition method based on user touching input has been presented. To obtain 3D coordinates of touching input, an extremely low-cost system has been developed based on widely available consumer electronics devices including Wii controllers and IR-LED pens. The system enables a user to easily interact with both real and virtual objects and quickly construct feature-based CAD models. To achieve a desired accuracy from the input of an arbitrary number of Wii controllers, a regression-based 3D coordinate computation method has been presented. The test results indicate that the error of a 3D touching point inside the working volume of the system is generally less than 1 mm. Various hardware and software techniques have been presented to improve the users’ capability on defining general 3D shapes, e.g. the integration of a rotating table and different object alignment methods. Several test cases were given to illustrate the effectiveness and efficiency of the presented method.

One limitation of the current system is that it has limited capability on processing complex freeform surfaces. We are exploring various approaches including the integration of a 3D scanner in the system to address it. In addition, methods based on more complex regression models are to be explored to achieve better accuracy in 3D coordinate computation.
REFERENCES