Virtual Interactive Prototyping

Kurien Zacharia, Eldo P. Elias, and Surekha Mariam Varghese

Mar Athanasius College of Engineering, Kothamangalam, India
{kurienzach,eldope}@gmail.com, surekha@mace.ac.in

Abstract. In this paper, we introduce a new method of technology enhanced prototyping called Virtual Interactive Prototyping (VIP). Prototype is a model that mimics the static or working behavior of an actual product before manufacturing the product. Prototyping is implemented by displaying the individual components over a physical model constructed using Cardboard or Thermocole in the actual size and shape of the original product. The components of the equipment or product such as screen, buttons etc. are then projected using a projector connected to the computer into the physical model. Users can interact with the prototype like the original working equipment. Computer Vision techniques as well as sound processing techniques are used to detect and recognize the user gestures captured using a web camera and microphone. VIP is a fast, flexible and interactive prototyping method and has many advantages over existing immersive video prototyping methods.

Keywords: Prototyping, Augmented Reality, Image Processing, Sixth Sense.

1 Introduction

Virtual Interactive Prototyping is an interactive prototyping method that allows to create and test prototypes easily by using computer simulated objects displayed on to an actual physical model using a projector. This means user can create a model on the computer and test it out using a real physical model without actually assembling any component using the concept of augmented reality. Interaction with the physical model is achieved through use of computer vision technologies for tracking the user movements.

Prototype is a model (working or static) that allows the product to be visualized or simulated before an actual product is put to manufacturing line. Currently there are different prototyping methods available depending upon the kind of product that we want to manufacture. Physical mockups (Hardware Prototyping) have always played an important role in the early conceptual stages of design. It involves creating a static model of the product from cheap and readily available materials such as paper, cardboard, Thermocol etc. They are commonly used in the design and development electronic equipments such as mobile phones, physical mockups are used in early stages of design to get a feel of the size, shape and appearance of the equipment.

Another kind of prototyping that is particularly used for design of electronic equipment's is the software simulation. The software required for the equipment is

N. Meghanathan et al. (Eds.): CCSIT 2012, Part II, LNICST 85, pp. 594–603, 2012.

written and then run in simulator software inside the computer. This method allows testing the software of the product and the user interactions with it.

The disadvantage of the above methods is that the user experience remains disjoint. In Hardware prototyping the user gets the touch and feel of the product but does not include the working or interactions with the product. In software based simulations the user gets to know how the product with interact but he has no touch and feel of the actual product.

Other methods of prototyping like using a prototyping workbench allows to create hardware interface with relative ease, but the cost factor makes them beyond the reach of most small and medium scale industries. VIP is a fast, flexible and interactive prototyping method's and has many advantages over existing prototyping methods.

Virtual Interactive Prototyping (VIP) is a virtual reality prototyping method based on Sixth Sense[1] and an existing prototyping method called Display Objects[2]. A physical model in the shape and size of the actual product is constructed using Paper, Cardboard or Thermocol . The components of the equipment such as (viz. screen, buttons etc.) are then projected from a projector connected to the computer into the physical model. A web camera tracks the user's gestures as well as the physical model and reacts appropriately. The accuracy of touch detection using computer vision alone is not sufficient for sensing. Therefore a microphone is attached to the surface, and the change in input level from the touch of the finger is detected. By combining the input from the microphone and camera a usable interactive system is developed. VIP uses Computer Vision techniques as well as Sound processing techniques to detect user interactions so that the user can touch and interact with the prototype like actual working equipment. The system model used in VIP is shown in Figure 1.

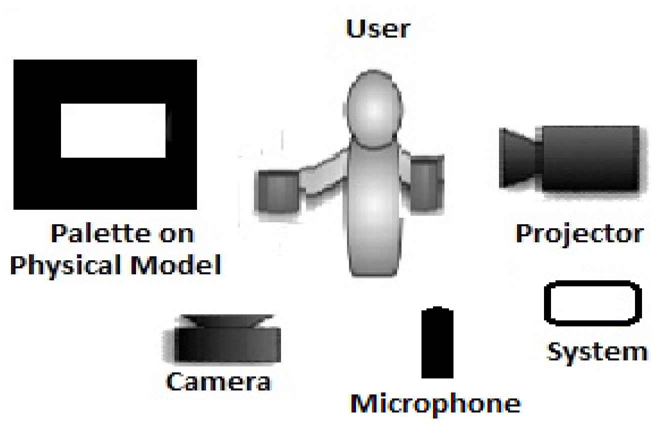

Fig. 1. The Model

2 Related Work

Display objects [2] introduced the concept of virtual models that can be designed on a Macintosh system and then projected to real physical models called display objects.

The disadvantage of the system is that user interactions in this system is based on expensive high end infrared based camera which require calibration each time the system is put to work.

Sixth Sense is an innovative computing method that uses user gestures to perform day to day activities. Sixth Sense uses a camera, a mini projector, microphone, and color markers to track fingers. Sixth Sense suggests a new method of interaction . It is used for a variety of interesting applications such as checking flight delays with flight ticket and automatically getting the reviews of books just by showing its Cover etc. In Sixth sense, markers with distinct color are placed in each of the fingers. Positions of the fingers are identified by locating the positions of the color markers.

The concept of interactive techniques for product design has been used elsewhere. Avrahami, D. and Hudson used interactive techniques for product design [3]. Use of static virtual artefacts has been proposed for the analysis of team collaboration in virtual environments [4]. The display methods uses projections to render a realistic model of the object according to the users viewing position. Also [5] (Maldovan, K., Messner, J.I. and Faddoul, M., (2006)) used a virtual mockup process for reviewing designs of physical structures such court rooms, restaurants etc. Pushpendra et. al. proposed an approach using immersive video with surround sound and a simulated infrastructure to create a realistic simulation of a ubiquitous environment in the software design and development office[6].

This paper proposes a system combining the technology used in the sixth sense system to implement the prototyping method as proposed in the display objects method. The result is a very low cost system with minimal setup or calibration time without compromising on the usability and performance of the system.

Similar technologies are being employed to create gadgets like Optimus Maximus Keyboard [7] in which each key can be programmed to display and be used for any user configurable purpose. Digital paper systems[8] employs sensors placed in papers to create an interactive paper.

3 Implementation

Common Image and sound processing techniques are extensively used in implementing the concept. EmguCV, which is a C# wrapper for the OpenCV image processing library developed by Intel is used for image processing. The Open Source Computer Vision Library (OpenCV)[9] is a comprehensive computer vision library and machine learning written in C++ and C with additional Python and Java interfaces. It officially supports Linux, Mac OS, Windows, Android etc.

The programming language used is C#.net as it facilitates fast development. For audio capturing and processing BASS audio processing library is used. All the core libraries that are used for implementing VIP are open source and platform independent. However VIP is implemented in Windows platform.

The major processing modules are

3.1 Image Processing

User interacts with the model using hand gestures. Color markers are placed on the user's fingers for distinguishing the commands. For tracking the interactions of the user, the position and colour of the fingers are sensed using standard image processing algorithms for edge detection and segmentation.

An HSV Color space based Color segmentation algorithm is used to isolate the color of the color marker. The center point of the segmented region is roughly taken as the position of the user's finger. For tracking the motion of the display object, first corners of the physical prototype are located. A grayscale based edge detection algorithm is used for this purpose. The image of the product is then fitted into this rectangle using image transforms.

Segmentation

For tracking the user actions, the image of the color marker should be separated from the picture captured by the live camera. The segmentation algorithm removes all other color from the input image other than the color of the color marker. Output of the segmentation algorithm is a black and white image with all areas black except the portions having the color of the color marker. The features extracted from this are the edges. They are extracted by looking for the difference b/w pixel values. Where there is an abrupt change in the pixel value (like b/w black and white) we define it as an edge.

Color values are represented in 8 bit format in the system. An HSV based color segmentation algorithm is used for better efficiency. Images are represented in 8 bit bitmapped format, which used a 3 dimensional matrix for storing pixel values. The threshold values for Hue, Satuation and Value are represented as HueThreshL, HueThreshH, SatThreshL, SatThreshH, ValThreshL, ValThreshH. Algorithm 2 describes the steps in the segmentation.

Algorithm 1. Segmentation Algorithm

```
for i=0 to imgmat->width
for j=0 to imagmat->height
color<-imgmat[x,y];
if(color.hue < HueThreshH && color.hue >HueThreshL &&
color.sat>SatThreshL && color.sat <SatThreshH &&
color.Value > ValThreshL && color.Value < ValThreshH)
imgmat[x,y] = 255;
else imgmat[x,y] = 0;
```

Tracking

The tracking algorithm includes a convex hull based algorithm to find construct a convex hull based feature extraction system. The extracted features are stored in a feature list. These features are then tracked for different kinds of interactions. The input to the tracking algorithm is the Binary mask produced by the segmentation algorithm. Tracking algorithm is described in Algorithm 2.

Fig. 2. Result of segmentation (left) and tracking (right)

Algorithm 2. Tracking Algorithm

```
for i=0 to imgmask.width
for j=0 to imgmask.height
for k=0 to 8
if (color[imgmask[x,y]-imgmask[x,k%3]]>colorthreshold)
if(x>maxx && y > maxy)
   point3 = point(x,y);
else if(x>maxx && y < miny)
   point2 = point(x,y);
else If(x<minx && y> maxy)
   point4 = point(x,y);
  else if(x<minx &y<miny)
   point1 = point(x,y)
```

3.2 Sound Processing

Image processing techniques have limitations in detecting whether the user has touched the virtual prototype or not. Therefore we use a microphone attached to the Virtual Prototype to detect whether the user has touched it or not. The input from the microphone is first passed through a Band Pass Filter to reduce background noise and then the amplitude of the resultant signal is compared with a threshold value to detect taps.

4 Constructing the Prototype

To construct a prototype the user will have to first drag and drop all the necessary components from a virtual palette into the virtual prototype using hand gestures. The components can be then rearranged, moved and resized inside the Virtual Prototype. After all the required components are set up, the user can then start the software simulation of the product. This means he can use it like real equipment, in the way the programmer has programmed it. The user can have both the touch and feel of a real hardware prototype as well as all the interactions.

5 Interaction Techniques

The workbench incorporates both one-handed and two handed techniques so that the user can interact with the model with one hand while browsing interaction elements with the other. The non-dominant hand is typically used to hold the physical model, while the dominant hand picks up interaction elements from the Palette. The Palette can be held or placed on a surface if hands are occupied with a physical model. Interaction elements are then pasted and placed onto the physical model using simple tapping and dragging techniques. To allow recognition of finger action by the system, fingers need to be augmented with a Vicon marker. Typically, only one or two fingers need to be enhanced in this way. Gestures allow fine-grained actions on the physical model and Palette.

5.1 Pointing

Pointing simply consists of moving the finger over the surface of a physical model or Palette. This does not result in an action until the surface is touched.

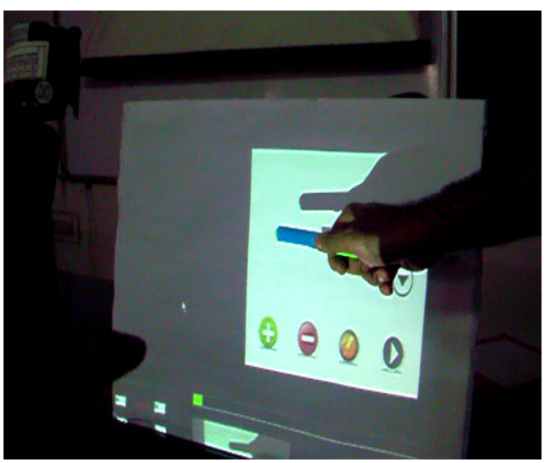

Fig. 3. User using color marker for pointing clicking etc.

5.2 Tapping

Users can select elements on the surface of the Palette by tapping them once. This causes the element to be copied to a virtual clipboard on the tip of the finger. If a physical model is subsequently tapped, the clipboard element is placed where that tap has occurred.

5.3 Clicking

Users can lock interaction elements on the panel by tapping the "Play" button on the Palette. This provides for functional interaction with elements on the surface of the panel, allowing users, for example, to press buttons that trigger some visual action. After devices are locked, any tap is interpreted by the Quartz Composer engine as a click on the element that executes its associated functionality.

5.4 Dragging

Users can select elements on the surface of the Palette by tapping them once. This causes the element to be copied to a virtual clipboard on the tip of the finger. If a physical model is subsequently tapped, the clipboard element is placed where that tap has occurred.

5.5 Pinch/Release

The Pinch gesture is initiated by bringing a thumb and index finger into contact with the extremities of a target element, and then drawing the thumb and index finger together. The Release gesture is performed by reversing this action. This acts to resize the interaction element upon which it is performed. In addition to these refined interaction techniques, the workbench supports a number of more coarse-grained (two handed) gesture-based interaction techniques.

5.6 Wiping

Users can move through layers of content on the surface of the panel by performing a wipe gesture, from the bottom left corner of the physical model upward. When editing objects, this is used to reveal the Quartz Composer patch that underlies the functionality of the panel. Layers can be closed by an inverse wipe.

5.7 Drag and Drop

Users can pick up an element through a tap on one of the items, for example, the Palette, then touching an empty area of another, causing the element to be copied to the empty area at the location of the tap.

5.8 Two-Handed Pinch/Release

The effect of this gesture is the same as the regular Pinch/Release, but performed by two-handed operation using the index fingers. This allows for a greater range of the gesture, and is very suitable for interactions on larger surfaces.

6 Programming Interaction Elements

After display elements are placed on the physical model, a script generates a Quartz file for the panel that can be edited in one of three ways: on a computer, on the Palette, or on the physical model itself. The physical model is placed in edit mode by pressing a button on the Palette, or by providing a wipe gesture on its surface. To connect input elements with output elements, users drag connectors from the outlets of the former onto the inlets of the latter in Quartz Composer on a laptop. A screen element, in turn, is scripted through the contents of a movie file connected to one of its inlets. Simple interactive behaviors are limited to simple actions, such as starting, stopping or scrolling through the movie file displayed on the screen element. More complicated behaviors are made possible by dragging Quartz Composer subpatches from the palette onto the physical model, and connecting inlets and outlets. These subpatches are programmed on the computer, rather than on the display panel. Because subpatches are a regular part of Quartz Composer, they may contain Objective-C code to further extend behaviors ranging from video feeds, to screen captures, Flash content, live webpages, etc. Since display panel is made out of real materials, it is easy to extend its behavior with real world artifacts as well, mixing properties of bits with those of atoms. For example, paper sketches or physical buttons can be easily affixed on the physical prototype, and linked with interactive content. This not only allows quick iterative revisions of the physical model, but also allows for physical elements to be mixed with digital elements, for example, to provide feedback for an on-screen input element. One example of this is the use of physical pushpins to simulate surface effects of buttons displayed on the surface. This allows for feedback when interacting with the physical prototype.

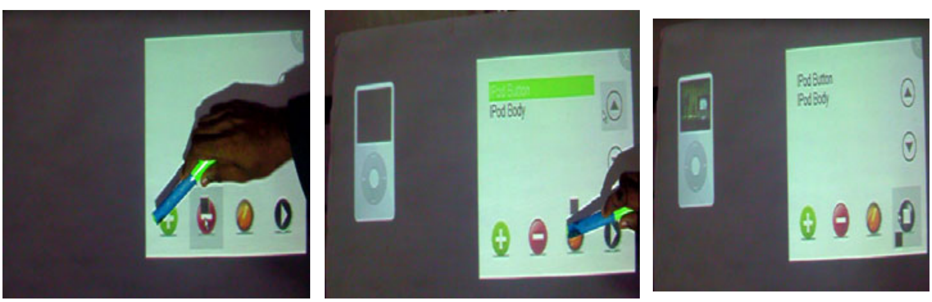

Fig. 4. Contruction of a prototype

7 Features of the System

The main advantage of the system that we propose is the combination of both hardware and software prototyping. Apart from that this systems needs only $1/8^{th}$ of the cost of creating a real prototype from scratch. This method also allows for real time modifications of the prototype, which is not possible on a real prototype. Also we plan to implement this software as Open Source software so that it will reduce any software costs, and the cost of the system would be purely hardware costs. Since all the libraries that we use are open source, we can make this system also Open Source.

7.1 Limitations

The system is sensitive to lighting conditions, since it mainly rely on image capturing and processing. Too bright or too low light can hamper the system performance. Color segmentation cannot be accurately done when the background is fast moving.

8 Conclusion

A key issue in the new electronic era is the effort required for rapid prototyping and evaluation of user interfaces and applications. Existing technologies for these tasks are either slumpy (e.g. paper prototypes, mental walkthroughs) or require a near full-scale deployment. VIP, a rapid prototyping workbench for designing new devices which projects functional interfaces over 3D physical objects is discussed in this paper. It provides a low-cost and rapid means to prototype user interfaces and applications. VIP targets a much earlier stage of physical design with functional prototyping than normally possible. It helps in creating physical prototypes with functional content fast, and with little effort.

VIP can be of great help for small and medium scale industries, since they cannot afford to build a prototype using a prototyping workbench. Such industries include the flourishing small scale mobile phone manufactures, which are based on relatively small capital and does the job of assembling pre made parts rather than building own components.

VIP, saves the electronic equipments used in prototyping and reduces electronic waste contributing to environment preservation and overall sustainable development. VIP can be termed as Green Technology.

References

1. Mistry, P.: Sixth Sense, http://www.pranavmistry.com/sixthsense
2. Akaoka, E., Ginn, T., Vertegaal, R.: Display Objects: Prototyping Functional Physical Interfaces on 3D Styrofoam, Paper or Cardboard Models. Proceedings of ACM (2010)
3. Avrahami, D., Hudson, S.E.: Forming Interactivity: A Tool for Rapid Prototyping of Physical Interactive Products. In: The Proceedings of the ACM Symposium on Designing Interactive Systems, DIS 2002, pp. 141–146 (2002)

4. Gopinath, R.: Immersive Virtual Facility Prototyping for Design and Construction Process Visualization, M.S. Thesis, The Pennsylvania State University, University Park, PA (2004)
5. Maldovan, K., Messner, J.I., Faddoul, M.: Framework for Reviewing Mockups in an Immersive Environment. In: 6th International Conference on Construction Applications of Virtual Reality, Orlando, FL, August 3-4, 6 pages (2006)
6. Singh, P., Ha, H.N., Olivier, P., Kray, C., Blythe, P., James, P.: Rapid Prototyping and Evaluation of Intelligent Environments using Immersive Video. In: Proceedings of the MODIE 2006 Workshop, MobileHCI 2006, Espoo, Finland, pp. 36–41 (2006)
7. Optimus. Optimus Maximus Keyboard (2008), http://www.artlebedev.com/everything/optimus/
8. Holman, D., Vertegaal, R., Altosaar, M.: PaperWindows: Interaction Techniques for Digital Paper. In: The Proceedings of CHI 2005, pp. 591–599. ACM Press (2005)
9. OpenCV Libray, http://www.sourceforge.com/opencv