Poster: Physically-based Natural Hand and Tangible AR Interaction for face-to-face Collaboration on a Tabletop

Thammathip Piumsomboon[12*], Adrian Clark[1†], Atsushi Umakatsu[13‡] and Mark Billinghurst [1§]

[1]The HIT Lab NZ, University of Canterbury, Christchurch, New Zealand
[2]Department of Computer Science, University of Canterbury, Christchurch, New Zealand
[3] Takemura Lab, Informedia Education Division, Cybermedia Center, Osaka University, Osaka, Japan

Abstract

In this paper, we present an AR framework that allows natural hand and tangible AR interaction for physically-based interaction and environment awareness to support face-to-face collaboration using Microsoft Kinect. Our framework comprises of six major components: (1) marker tracking (2) depth acquisition (3) image processing (4) physics simulation (5) communication and (6) rendering. The resulting augmented environment supports occlusion, shadows, and physically-based interaction of real and virtual objects. We propose three methods of natural hand representations including mesh-based, direct sphere substitution and variational optical flow. A tabletop racing game and AR Sandbox applications are created based on this framework, showing the application possibilities.

Keywords: augmented reality; physically-based interaction; Kinect; physics engine; environment awareness; collaboration

Index Terms: H.5.1 [Multimedia Information Systems]: Artificial, augmented, and virtual realities;

1 Introduction

By overlaying virtual information into the real world, Augmented Reality (AR) [1] allows multiple users to share and collaborate on virtual tasks while carrying out natural face-to-face communication [2]. This overlap of task and communication spaces makes AR ideally suited for collocated collaboration. However there is a need to explore new three-dimensional (3D) interaction methods suitable for this environment.

In the past, researchers have explored a variety of techniques for AR interaction and some of the most popular methods are based on the Tangible AR interface metaphor [3] where real objects are used to manipulate virtual content. This can be intuitive because it utilizes the users' natural ability of interacting with the physical world. However there are some limitations with the Tangible AR approach, such as the need for physical input devices.

In this paper, we present an AR framework that supports face-to-face collaboration allowing people to use their natural hands to interact with virtual content such as a user can grasp a virtual object and pass it on to another user as they normally do with real objects. The system also aware of its environment so as to provide correct visual cues for each user and the virtual content behave realistically in the physical environment that it is placed in.

Our framework uses a Microsoft Kinect, a single image-based marker, and AR viewing cameras. The Kinect provides both RGB and depth-sensing cameras, is used to create a 3D task space above a tabletop. When the transformation between the Kinect and the AR viewing cameras is known, virtual content can be realistically composited in the environment.

We cover our framework in section 2. The performance is discussed in Section 3 and the conclusion and future works are described in section 4.

2 Proposed AR Framework

2.1 System setup and major components

Our framework uses a client-server model to offload the server from the responsibility of rendering. The server process comprises of five stages, while the client has three as shown in Fig. 1(a). To create an interaction volume, the Kinect, is connected to a server, and is positioned above the desired interaction space facing downwards, as shown in Fig. 1(b). A printed reference image marker is placed in the interaction space to calculate the transform between the Kinect coordinate system and the coordinate system used by the AR viewing cameras. Users can use several types of displays connected to the client PC for viewing the AR content, such as a handheld or head-mounted display, or a fixed monitor. The data flow process within the framework is illustrated in Fig. 2 (a) and the descriptions of each component are as follows.

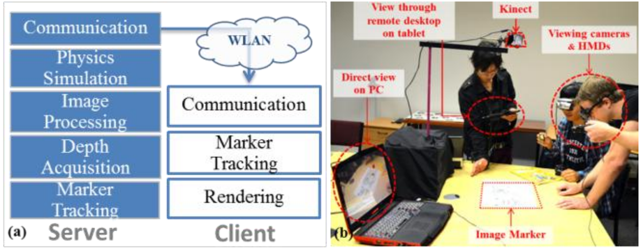

Figure 1: (a) System Architecture (b) System setup

1) *Marker Tracking*: The OPIRA natural feature tracking library [4] is used for natural feature based registration.

2) *Depth Acquisition*: OpenNI library is used to interface with the Kinect. Color and depth images can be accessed through the API, and the point cloud information of the environment is obtained.

3) *Image Processing*: The depth image obtained by the Kinect is prone to missing values due to shadowing of the infrared data. To resolve this, missing values are identified and Navier-Stokes inpainting algorithm is applied to estimate their values.

4) *Physics Simulation*: We use the Bullet physics library for collision detection and physics simulation. The point cloud information of the environment is reconstructed using a triangle mesh (trimesh). The mesh is used as a physical proxy for real time interaction of virtual and real objects in the interaction space.

5) *Communication*: VRPN library is used to create network connections between the server and clients. The transmitter on the

[*]thammathip.piumsomboon@pg.canterbury.ac.nz
[†]adrian.clark@hitlabnz.org
[‡]umakatsu@lab.ime.cmc.osaka-u.ac.jp
[§]mark.billinghurst@hitlabnz.org

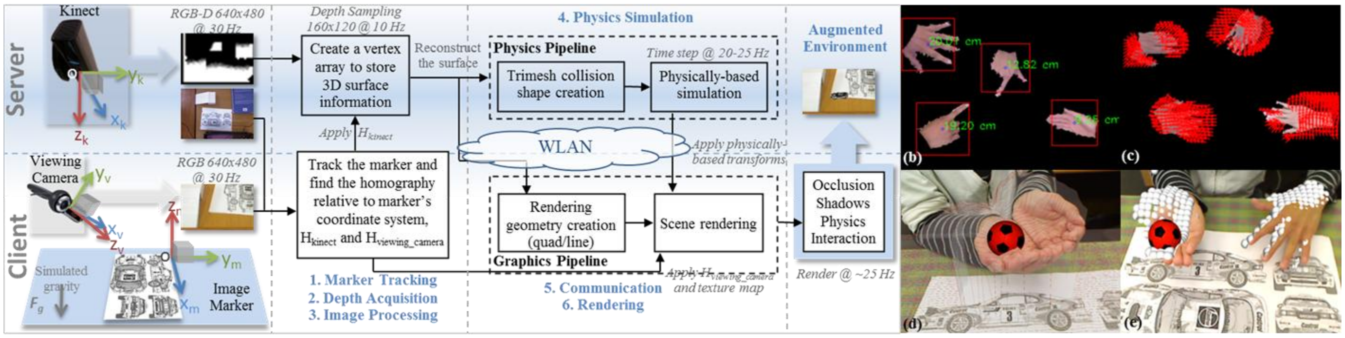

Figure 2: (a) Data flow process of our AR framework (b) Skin color segmented and bounding square (c) Optical flow of skin color pixels (d) Natural hand interaction using mesh-based representation (e) Particle-based representation

server queries the current pose of each physics proxy as well as the surface information and encodes them into the message. Once the client receives the message, the transformations are applied to each object and the ground mesh is updated.

6) *Rendering*: OpenSceneGraph, a 3D graphics API that utilizes a tree structure for managing the scene, is used for rendering. The transformation of the physical proxy in the simulated world is applied to its corresponding graphical representation. The input video image is rendered as the background using an array of quads, with an alpha value of zero allowing occlusion. A custom fragment shader was written which allows shadows from the virtual objects to be cast on the invisible terrain map.

2.2 Natural hand interaction

We have explored three methods for physically-based natural hand interaction, a trimesh-based approach and two particle-based representations. In the particle-based method, we experiment with direct sphere substitutions and variational optical flow. There are three steps in hand segmentation that is shared by all three methods of representation. Step 4 (a) is unique to direct sphere substitution and 4 (b) to optical flow. The details are as follows.

2.2.1 Hands Segmentation

Step 1: *Threshold the color image at the marker depth.*

Step 2: *Find skin color pixels.* We use skin and non-skin color probability tables created using Gaussian Mixture Model for RGB color space provided by [5].

Step 3: *Apply connected component filter and find hand contours.*

2.2.2 Mesh-based Representation

With the segmented hand image, a new trimesh can be created and overlay on top of the original ground mesh (Fig 2(d)). Representing hands with a second mesh that updates more frequently than the ground mesh does increase the accuracy in the physics simulation. However, a single viewpoint of the Kinect means that the mesh occupies the volume under the hand as well.

2.2.3 Direct Sphere Substitution

Step 4 (a): *Find bounding squares for hand contours.*

A hand representation is created in the physics simulation by an N by N grid of spheres (Fig 2 (b)). A median filter is applied to remove extreme height values. Each sphere displaces along the z-axis, perpendicular to the ground plane (Fig 2(e)).

2.2.4 Variational Optical flow

Step 4 (b): *Estimate the optical flow.* We use a combined local-global (CLG) method with bidirectional multi-grid solver [6]. Optical flow gives us the motion of image pixels on the x-y plane (Fig 2 (c)), the corresponding change in depth value, δz, can be found from the corresponding (x, y) tuple and further yields the scene flow information of the 3D surface. A sphere proxy is created for each skin pixel and displace according to the scene flow. Although, variational optical flow is a promising method, it is computationally demanding, making interaction difficult. For the future improvement, we will optimize our method by using parallel computing on the GPU.

3 Performance

We run the server on an Intel 2.4Ghz quad core desktop and the client on an Intel 1.8Ghz quad core laptop. Our client can render at over 25 frames per second (FPS). We created a tabletop task space of 75x50 cm and 120x80 cm. The Kinect provides an error of less than half a centimeter when placed between 70 - 120 cm. from the ground plane.

4 Conclusion and Future Works

We presented an AR framework that implement physically-based interaction and environment awareness to support face-to-face collaboration. Natural hand and tangible AR interaction were used for object manipulation. Three methods of natural hand interactions were proposed including mesh-based, direct sphere substitution and variational optical flow.

In the future, we plan to investigate additional methods of object segmentation to provide more realistic interactions between the real and virtual worlds through hand gestures. Multiple Kinect will also be used for a more complete representation of the environment. Eventually, we plan to evaluate these approaches to improve the AR experience.

References

[1] R. Azuma, "A Survey of Augmented Reality," *Presence*, vol. 6, pp. 355-385, 1997.

[2] M. Billinghurst and H. Kato, "Collaborative augmented reality," *Communications of the ACM*, vol. 45, pp. 64-70, 2002.

[3] M. Billinghurst, H. Kato, and I. Poupyrev, "Tangible augmented reality," presented at the ACM SIGGRAPH ASIA 2008 courses, Singapore, 2008.

[4] A. J. Clark, R. D. Green, and R. N. Grant, "Perspective correction for improved visual registration using natural features," in *Image and Vision Computing New Zealand, 2008. IVCNZ 2008. 23rd International Conference*, 2008, pp. 1-6.

[5] M. Kolsch and M. Turk, "Fast 2D Hand Tracking with Flocks of Features and Multi-Cue Integration," in *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04. Conference on*, 2004, pp. 158-158.

[6] A. Bruhn, J. Weickert, C. Feddern, T. Kohlberger, and C. Schnorr, "Variational optical flow computation in real time," *Image Processing, IEEE Transactions on*, vol. 14, pp. 608-615, 2005.