DEVELOPMENT OF VIRTUAL PLAYGROUND SYSTEM BY MARKERLESS AUGUMENTED REALITY AND PHYSICS ENGINE

Masafumi Takahashi † *Kazunori Miyata* ‡

† School of Knowledge Science
‡ Center for Knowledge Science
Japan Advanced Institute of Science and Technology
Ishikawa, Japan
E-mail: {masa-t, miyata}@jaist.ac.jp

ABSTRACT

Augmented Reality (AR) is a useful technology for various industrial systems. This paper suggests a new playground system which uses markerless AR technology. We developed a virtual playground system that can learn physics and kinematics from the physical play of people. The virtual playground is a space in which real scenes and CG are mixed. As for the CG objects, physics of the real world is used. This is realized by a physics engine. Therefore it is necessary to analyze information from cameras, so that CG reflects the real world. Various games options are possible using real world images and physics simulation in the virtual playground. We think that the system is effective for education. Because CG behaves according to physics simulation, users can learn physics and kinematics from the system. We think that the system can take its place in the field of education through entertainment.

Keywords: Augmented Reality, Markerless, Computer Vision, GPGPU, Physics Engine

1. INTRODUCTON

Augmented Reality (AR) is a useful technology for various industrial systems, for example education, entertainment and various situations. However, AR does not readily spread generally, because AR systems are complicated to setup for users. AR needs more attractive applications.

This paper suggests a new playground system which uses markerless AR technology. “Markerless technology” aims to reduce the burden of setup for users. Users can operate physics simulation base CG in real scenes by using this playground system interactively.

We think that the result of physics simulation is easy because CG is layered on top of a real scene. We think that the system is not only fun, but also educational.

2. RELATED WORKS

The system aims at being easy to use. Therefore, we have simplified the setup of the system. Scene analysis is performed by a single camera by the system. However it is difficult to analyze the three-dimensional structure of real scenes with a single camera. People usually use stereo camera or a depth camera. As for the stereo camera, calibration is troublesome, and a depth camera is expensive.

There is technique of Georg [1] for scene analysis in AR using a single camera. This technique makes an analysis of the ground plane in a camera scene from a single camera. This method does not require a stereo camera or depth camera.

Handy AR [2] is an AR system using markerless hand input. This system detects fingertips from a camera. This system is easy to use, because it is markerless and uses a single camera. However, it must use other techniques for outside environmental acquisitions, which is necessary for AR.

Touchlib [3] is a library realizing multi-touch user interface. This library can track fingers in markerless enviroments. We intend to include elements of the multi-touch input.

Phun [4] is a sandbox application that uses a physics engine. Phun is a sandbox tool, and users use physics simulation for an interface, such as for drawing software. Design software that includes physics simulation includes software such as 3D CAD. Operation of 3D CAD is difficult for computer beginners and children. They can easily use Phun at beginning. This is important when we think about the user interface. There is other software, such as *OE-CAKE!* [5]. This is better at fluid than Phun.

Method of Derek [6] makes a popup card from a photograph. This method estimates ground, still objects and sky from a photograph. This method estimates simple 3D structure of 3D from a photograph that is a 2D view.

Method of Sangwon [7] builds a 3D model from a line drawing. This method is limited to line drawing, however it can acquire an exact model, in comparison with the technique.

3. SYSTEM DESIGN

This chapter explains our system design. It describes our goal and the design of the system, and the “virtual playground”.

3.1 Goal

Our goal is to allow users to play with real scenes and CG interactively. This game is like building blocks or sandbox in AR. We aim to make it easy to realize physical phenomenon by using AR.

3.2 Virtual playground

The “virtual playground” which we imagine is a place where there are building blocks or sandboxes at virtual space. Users can do construction freely in this space. CG objects become the building materials in this space. CG objects obey physical laws, such as collision or gravity. CG can develop more complicated buildings than real building blocks. CG interacts with user hands and real background. We aim at making a game like the *Incredible Machine* [8] in AR. CG objects must be controlled by physics simulation to realize this.

3.3 CG and real scenes

There is already a virtual playground that uses physics simulation, only using CG. The system uses real scenes from a camera for the background of the playground. Background interacts with CG objects. Background is not a simple still image. Therefore 3D structure analysis of background is necessary.

3.4 User interface

There are interfaces, such for 3D CAD or 2D drawing software, in digital content creation tools which incorporate physics simulation. Dimensions of space treated by each are different. 2D operation is easier than 3D. We adopted interface 2.5D. The operation of user is 2D, however CG objects use 3D. We decided not to use depth.
It can be difficult in operation to track 3D movement with a single camera viewpoint, in 2D.

3.5 Physics simulation

Physical laws act on a CG object in the virtual playground. CG objects fall to the ground due to gravity. CG objects collide and rebound.

4. IMPLEMENTATION

There are three stages in the implementation of the system. The first stage is real scene analysis for Augmented Reality. The second stage is structure of markerless input. The third stage is handling of physics simulation. And this chapter explains about implementation of three stages.

4.1 Augmented Reality

This chapter explains about AR. We explain 2 things about AR. First, it speeds up computer vision. Our approach implements computer vision on GPU. Second, AR analyzes real scenes. 3D information of the real world is necessary for CG and real scene composition.

4.1.1 GPU Computer vision

AR uses computer vision for scene analysis. Computer vision has a high processing calculation burden. We solved this by treating computer vision in GPU. OpenVIDIA [9] and GPUCV [10] are libraries for computer vision with GPU. Or there are other methods, such as CUDA [11] or CTM [12]. We implement our system originally without using these.
Our implementation method accesses a camera device with DirectShow, and begins to write video frame data to texture of Direct3D. We perform image processing on pixel shader. GPU is more parallel than CPU, and it works to advantage for computer vision.

4.1.2 Scene analysis

Scene analysis is important for CG and real scene composition. Analysis requires various techniques. There are stereo camera, depth camera and prior measurement, however we analyze using a single camera. Single camera preparation is simple, however analysis is very difficult. We compromise by some limitations.
We do not move the camera. We do not perform dynamic analysis of scenes. Scene analysis is done in prepare.
Camera posture is got by G-sensor. G-sensor is TDS01V [13]. TDS01V has 3-degree G-sensor and 3-degree gyro sensor. Figure 1 shows TDS01V.
This process is not necessary to be real time. Method of scene analysis from the camera uses techniques such as SLAM. However we try techniques of image processing, such as those of Derek and Sangwon. If line drawing finishes processing from a real scene, precision is high in Sangwon. However, it is difficult for real scenery to become line drawing neatly. Derek cannot express depth, however it is strong in real scenes.

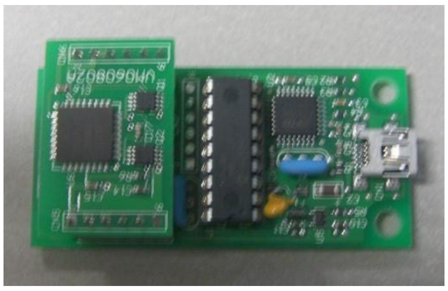
Figure 1: G-sensor. TDS01V

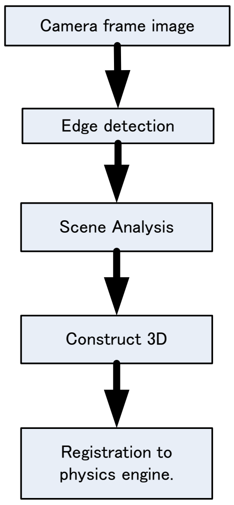

Figure 2: Flow of scene analysis.

Figure 2 shows flow of scene analysis. At first, process captures a frame image from camera, and edge detects from camera frame image. The process analyzes scenes from edge image. 3D model is constructed in this way. At Last, registration to scene of physics engine.

4.2 Markerless input

We decided to use fingers for input to the system. Therefore it is necessary to track user's fingers. A user operates the system from user interface, such as multi-touch by using multiple fingers.

HandyAR and Touchlib use multi-points input method to track fingers. Touchlib tracks fingers from the back of the screen, therefore it can't be used.

Fingers are detected by skin color. We extract the skin color domain from camera frame. Skin color detection is possible by using HSV or CIE L*a*b color spaces. We used CIE L*a*b color space. Figure 3 shows skin color detection.

Next process performs collision detection as silhouette of hands and CG object. Input decides not to think about depth. Figure 4 is performing collision detection for a hand and CG object.

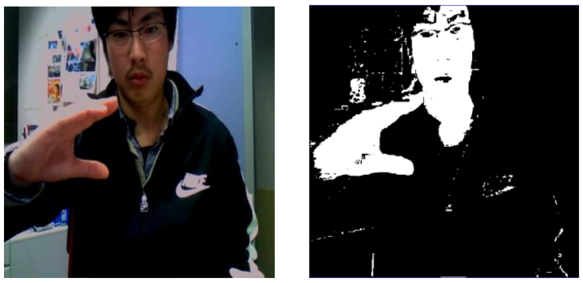

Fig 3: Left: Camera frame and composed CG. Right: Skin color detection from camera frame

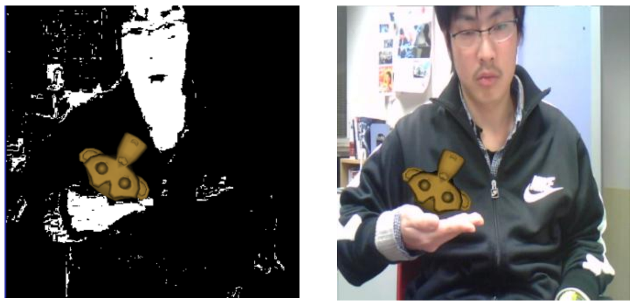

Figure 4: Left: Skin color collides to mesh. Right: Composited CG and camera frame.

4.3 Implement physics simulation

System uses a physics engine for physics simulation of CG objects. We implemented a physics engine in the system. Physics engine is a library for physics simulation. Physics engines can be commercial use or open source. We used Bullet physics engine [14]. Bullet is an open source physics engine library.

CG objects are treated as rigid bodies basically. CG objects are moved by finger.

Physics elements take in virtual playground is as table1. User can play by using them.

Table 1: Physics elements

Element	Function
Mass and density	Objects have mass and density.
Gravity	Gravity acts on CG objects. It is the same in the real world.
Friction	Each CG objects interferes by friction with other.
Restitution	This is a parameter about processing to bounce when CG objects collide.
Joint	CG objects can joint with other objects.

5. APPLICATIONS

This chapter explains examples of possible uses of the system. We suggest two examples here. There are two applications education and cognitive education.

5.1 Physics

The system can be used for the study of dynamics from using physics simulation. We think that in particular gravity, friction and mechanical work can be used in possible educational projects.

Figure 5 shows physics demo. There is a box on the left of a board. Because the left declines by gravity, the box slides down from the board.

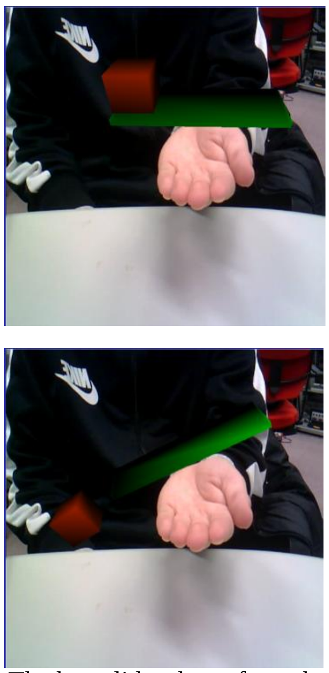

Figure 5: The box slides down from the board.

5.2 Cognitive education

Because play using building blocks and sandboxes is possible, system can be use for cognitive education.

Figure 6 shows building block sample. We decide the posture of the camera in virtual world by G-sensor.

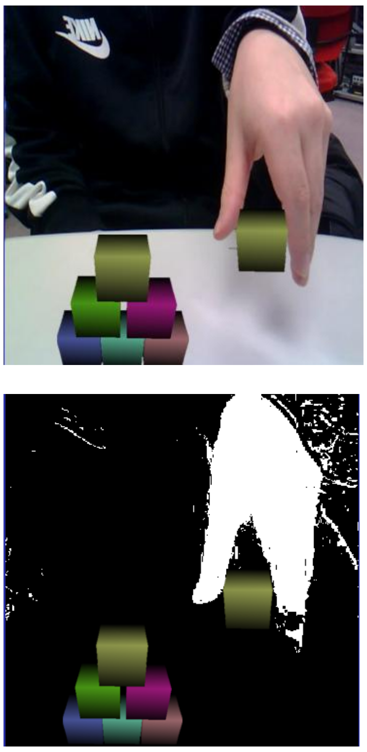
Figure 6: Building block samples.

6. RESULT

The system works with Windows Vista PC. PC spec is Core2 Quad 2.66 GHz, RAM 4GB and GeForce 260 GTX 768MB in real time. We use Direct3D10 for rendering of 3D CG. Camera uses a Sony PlayStation Eye. Figure 7 shows PlayStation Eye. The reason why we used PlayStation Eye is its high frame rate. Resolution of the camera is 640 x 480, and frame rate is 60 per sec.

Figure 7: PlayStation Eye

7. CONCLUSION

We developed a new AR playground system in this paper. This system is a new creation tool in which a user operates physical motion. The system can operate on objects through physics simulation of physical movement. Various applications are possible at present, and the system is still being developed further.

Problems, as follows. First, dimensions of operation. Second, the kind of physics elements.

We decided to use interface 2.5D due to the problem of tracking. It is big problem to make this system 3D. We want to solve in the future.

Physics simulation supports only rigid bodies now. We want to support softbodies and fluid in future. For example, we want to include in liquid such as the water and ropes and nets for play.

8. REFERENCES

[1] Georg Klein, David Murray, Parallel Tracking and Mapping for Small AR Workspaces, ISMAR 2007. 6th IEEE and ACM International Symposium on (2007), pp. 225-234.

[2] T. Lee, T. Höllerer, Handy AR: Markerless Inspection of Augmented Reality Objects Using Fingertip Tracking, IEEE International Symposium on Wearable Computers (ISWC), Boston, MA, Oct. 2007

[3] Touchlib, http://nuigroup.com/touchlib/

[4] Phun, http://www.phunland.com/wiki/Home

[5] OE-CAKE! , http://www.octaveengine.com/en_casual/oecake/

[6] Derek Hoiem , Alexei A. Efros , Martial Hebert, Automatic photo pop-up, ACM Transactions on Graphics (TOG), v.24 n.3, July 2005

[7] Sangwon Lee , David Feng , Bruce Gooch, Automatic construction of 3D models from architectural line drawings, Proceedings of the 2008 symposium on Interactive 3D graphics and games, February 15-17, 2008, Redwood City, California

[8] Incredible machine, http://en.wikipedia.org/wiki/The_Incredible_Machine

[9] James Fung , Steve Mann, OpenVIDIA: parallel GPU computer vision, Proceedings of the 13th annual ACM international conference on Multimedia, November 06-11, 2005, Hilton, Singapore

[10] GPUCV, https://picoforge.int-evry.fr/cgi-bin/twiki/view/Gpucv/Web/WebHome

[11] CUDA, http://www.nvidia.com/object/cuda_home.html

[12] Brook+, http://ati.amd.com/technology/streamcomputing/

[13] TDS01V, http://akizukidenshi.com/catalog/items2.php?p=1&q=TDS01V

[14] Bullet physics library, http://www.bulletphysics.com