Abstract:

Recently 3D technology has been applied to various fields, including higher education. Many universities have created 3D virtual environments to provide effective interactive teaching materials and allow students to interact virtually in real time. At the same time, mobile devices have rapidly improved, giving users convenient yet powerful media for communication. This paper describes a method for creating Noh-masks, which are used for traditional Japanese performances, as 3-D virtual objects viewable either on a standard web browser or mobile devices such as iPad or Android terminals. These virtual Noh-masks have significant cultural values. The techniques described below are useful for creating 3D virtual replicas of similar valuable assets that can be placed in virtual museums.

Keywords: 3D virtual reality object, Objective-C, OpenGL ES for mobile devices.

I. Introduction:

Hosei University has long been actively researching and collecting materials related to Noh Theater, and established the Noh Theater Research Institute in 1952. The collection holds over 40,000 items found mostly in Japan, including rare and extremely valuable manuscripts dating from the Muromachi and the Edo eras. The Institute’s continuing research carries a meaningful mission to introduce Japanese culture to people around the world.

By 2015, 80% of people accessing the Internet will be doing so from mobile devices [1]. As the use of mobile technology increases, it gives educators around the world a new way to deliver lectures remotely. The potential of mobile computing is already being demonstrated in many projects at institutes of higher education [1]. An every-time, anywhere, and anyone system called mobile learning is being created. It is an updated version of e-learning in which there is no requirement for physical study locations. With touch screen user interface, ten-hour batteries, and desktop computer features, tablet PCs or smart phones are indispensable devices for the future of education.

At the same time, research in 3D multimedia is advancing at a rapid pace [4]. 3D media content technology simulates every aspect of reality [3]; this is known as virtual reality. In the virtual environment, users easily recognize and interact with each other to study, or collaborate. In order to create the virtual world, many virtual objects must be made. Virtual objects fall into two categories: one, pure virtual objects, which are created by people’s imaginations, or two, digital replications of real objects. The second is what we will discuss in this paper. The 3D Noh-mask for mobile devices is a project that captures real Noh-masks in many different angles, and uses 3D software to create a 3D object model. The user can then control the app with the fingertips, or mobile accelerometer to see the object at any angle.

II. Technique Procedure:

There are many ways to capture a real object and generate its 3D model; however, this paper introduces the use of Strata Foto 3D CX and Blender. The Strata Foto 3D CX is commercial software, which exports its 3D data so that Blender, open-source software, can recognize it. Then Blender exports the 3D data object to iPhone, and iPad. It’s important to have accurate data in every photograph so that the software will work properly.

1. Capture a real object
1.1 Studio set-up
1.1.1 Mat, turntable, and stand
1.1.1.1 Mat:

The Strata Foto 3D CX uses a flat sheet of paper printed with dots to determine the sequence in which each photo was taken. The mat can be printed from the software’s help file. A fifteen dot-pattern is on the mat, and the first dot-pattern is marked by a line.
1.1.2 Turntable:

The turntable is a plate that is the same size as the mat, and can easily spin 360 degrees.

1.1.3 Stand

The stand is a cube of the same color as the mat, used to raise the object away from the marking on the mat and reduce shadows.

The mat, turntable, stand, and object should stick together as a single unit when they are rotated.

![Figure 1: Object, stand, mat, turntable, backdrop setup](image)

1.1.4 Solid color backdrop

Depending on the object’s color, the color of the backdrop must be an opposite color. For example, in this project, we use a plain green background because the mask appears in a dark brown or black. The backdrop is always placed behind the object in every photo that is taken, making it easy for the software to detect the object and the background.

1.1.5 Lighting

There must be no shadow on the backdrop when the object is placed on the stand. In order to have a good quality photo shoot, light in the studio should be distributed evenly on the object and its surrounding area. This is best achieved when all lights used are of the same type.

1.1.6 Camera

Photographs may be taken using a standard digital camera. In order to have the same angle for the photo shoot, the camera must be mounted on a tripod. Here are some recommended pre-settings for the camera:

- Disable the camera’s flash
- Use diffused light to make sure that there are no hot spots on the object
- Disable the digital zoom
- Lastly, clean the camera’s lens

1.2 Photograph the object

1.2.1 Test before taking the picture

From the camera’s angle, turn the turntable in many different directions to make sure that the backdrop covers the object’s background. In any shot, the view of the entire object must be captured; the mat dot-pattern must be shown in the photo so that the software can recognize the object. There are three views: side-view, high-angle-view, and top-view.

1.2.2 15 photos from the side-view

15 photographs from the side-view provide the most important data for the software to build the object’s 3D model. In the picture below, the angle between the camera’s center line and the object’s horizontal line is zero degrees. This is the angle of the side-view. After all configurations are checked, take the first shot. Then turn the object clockwise to the next dot-pattern, and take a second shot. Repeat the steps until all fifteen photos are taken.

![Figure 2: Camera position for side-view photographs](image)

1.2.3 3 photos from 60-degrees

In order to maintain the distance between the object and the camera after taking the side-view photos, keep the tripod with the camera in the same (x,y) position. Next raise the camera up and tilt it until the camera’s center line and the object’s horizontal line are about sixty degrees. At the same time, make sure the first dot-pattern is facing the front, and then take the first shot. Next move the turntable 120 degrees, or to the fifth dot-pattern, and take the second picture. Finally move the turntable 120 degrees, or to the tenth dot-pattern, and take the third shot.

1.2.4 1 top-down photo

In order to maintain the distance between the object and the camera after taking the side-view photos, keep the tripod with the camera in the same (x,y) position. Next raise the camera up and tilt it until the camera’s center line and the object’s horizontal line are about sixty degrees. At the same time, make sure the first dot-pattern is facing the front, and then take the first shot. Next move the turntable 120 degrees, or to the fifth dot-pattern, and take the second picture. Finally move the turntable 120 degrees, or to the tenth dot-pattern, and take the third shot.
There is only one photo to show the top view of the object; therefore, it is important to do it carefully. In this case, the camera’s line and the object’s horizontal line make a ninety-degree angle. The camera’s top and bottom line should be little outside the edge of the turntable so that the dot-pattern is shown in the photograph. After everything is aligned, shoot the photo.

1.2.5  1 bottom photo

As with the top-down photo, the bottom photo has to be taken carefully in order to show the detail on the base of the object. Turn the object upside down, and repeat the process for taking the top-down photo.

1.3 Creation of the 3D object model

Import all images from the digital camera into one folder and then open the Strata Foto CX and a new project. Next, import all images into the project. Use the software’s wizard to mask all images so the software automatically detects the photos that have been imported. The software makes those photos’ background transparent. If the software could not completely change the background, especially around the edge of the object, go to those photos and manually clean the remaining pixels. This process will improve the quality of the photos so that the software can build a better 3D object from them.

From the software menu, generate a 3D wire frame model. This is required for the completion of a 3D object. Finally, apply texture over the entire wire frame model.

1.4 Export the object

Export the object’s data as .3ds from Strata Foto CX. Next, open a new project in Blender, open-source software, and then open the .3ds object into the Blender. Then, from Blender, export the data object as .obj. Finally, convert the .obj file into an Objective-C header by using a Perl script.

2 Coding

2.1 Development environment

2.1.1 Xcode, and Simulator

Usually in iPhone or iPad development, a lot of time is devoted to building a user interface by using UI Builder, a user interface software which is included in Xcode. A developer may code the app by using Object-C, a language of Xcode, to integrate. However, in this project, we used Xcode as the main workplace, and all interfaces are built in Xcode.

Simulator is used for iPhone or iPad testing. However, Simulator cannot handle all iPhone, or iPad features such as the accelerometer. Use of a real iPhone, or iPad is recommended for a live test. In order to use real devices for testing, a developer must register for an Apple developer account. Then, he or she can download code samples, create project members, or generate provisioning to integrate with real devices.

2.2 Coding procedure

2.2.1 Create a new OpenGL ES project

OpenGL ES is a cross-platform standard 3D API for handheld and embedded devices. OpenGL ES is a C-based API, and, in Mac OS, it integrates seamlessly with Objective-C based Cocoa Touch applications. When a developer develops an OpenGL ES application, his or her OpenGL ES content is rendered to a special Core Animation layer, known as a CAEAGLLayer object. In the project, we have used OpenGL ES 1.1, which provides a standard fixed-function 3D application.

There are a couple of ways to create a new OpenGL ES 1 from Xcode wizard or from Apple OpenGL ES 2.0 example code. In the project, we used the example code. Then, we imported all masks’ Objective-C header files and JPEG textures into the project’s Resource folder. When we had all necessary data, we overrode some of the project’s classes.

♦ ESRenderer.h

This is the class to define functions of ES1Renderer. In this project, there is no implementation for this class.

♦ ES1Renderer.h, and ES1Renderer.m

The ES1Renderer class creates an OpenGL ES 1.1 context and draw using OpenGL ES 1.1 functions. They are the main classes to handle the external data such the object’s JPG map texture, and Objective-C header file of the vertex array. There are some declarations in the header file, then implement them in the method file:

- “loadImageFileL(NSString *)name ofType:(NSString *)extension texture
  uint32_t)texture” function is to import the external JPG texture map into OpenGL ES app when the app is running.
- “touchedAtX: (float)x andY:(float)y” is used to detect user’s touched-coordination
- “drawTexture” is re-drawing the screen whenever any graphic is updated.
- “renderMain:model:(uint32_t)model;” is the active place in OpenGL ES app after the “init()” function completed its task.

Especially in ES1Renderer.h, all vertex coordination for the object header files need to be imported. For example, mask1.h.

♦ EAGLView.h, and EAGLView.m

The EAGLView class is a UIView sub-class which renders an OpenGL view. The class first tries to
allocate OpenGL ES 2.0. If it fails, then it allocates an OpenGL ES 1.1.

In these two classes, “ES1Renderer.h” file must be imported so that EAGLView can use the OPENGL ES 1.1 context.

In EALView.h, “id <ESRenderer> renderer;” should be replaced by “ES1Renderer*renderer;”

♦ “ProjectName”AppDelegate.h and “ProjectName”AppDelegate.m

When creating a new project, “ProjectName” is the name a developer gives to his or her project. Then, Xcode will concatenate the “ProjectName” and AppDelegate to make the “ProjectName”AppDelegate header file, and its implement file. They are the main classes to tie everything together for running. There is no modification in these classes.

2.3 UI design

In this project, we tested two different ways to interact with users. The first way is to use an accelerometer. When a user moves the iPhone or iPad device, the mask will display according to the angle of movement. When he or she wants to view the back side of the object, a shake of the device is needed to turn the object 180 degrees. However, some users may find it difficult to control the object in a wanted angle. Therefore, we also implement a swipe-out and automatic 360 degree rotation. In this way, the mask object will display its front view as the default setting, then it will be rotated slowly. At any time, the user can stop the rotation for observation by touching the screen.

![Figure 2: A screen shoot for a Noh-mask running on an iPad](image)

III. Evaluation:

1. Disadvantage

There are some objects which may not work well with Strata Foto CX such as colorless glass, extremely complicated items like hairbrushes, mirrors or other highly reflective items, and objects with deep recesses like coffee cups. It takes a long time to build a 3D object from taking photographs to creating the data for OpenGL ES. There are many tasks that have to be done manually.

2. Advantage

Although some objects such as Japanese Noh-masks are difficult to replicate virtually, once they are replicated in 3D they can be used in further development as components in mobile games or any 3D documentary content.

IV. Conclusion:

On January 8th, 2011, we had a demonstration of the iPad application for audiences in the Traditional Arts at a State of the Art Technology seminar held in Burlingame, California. After a real Noh performance, we let users try our iPad app in which nine different Japanese Noh masks had been built. The audience was impressed to see those masks from many different angles. It was a great experience.

For future development, we are working to cut down the time it takes to create 3D reality-objects. We want to build a studio that can take all photos, and import them into a computer automatically. In addition, we plan to write scripts to create Objective-C header files from .3ds data files. We are also researching to create educational games, and other 3D content from virtual-reality objects.

Reference: