Smart Phone Augmented Reality

Raymond Imber, UNLV

Source Code @ https://github.com/rayman22201/brave-furry-waffle

ABSTRACT

There is an increasing amount of research in the area of augment reality as smart phones have improved in hardware capabilities and camera quality. Augmented Reality renders live video from the device camera onto the device display. Virtual 3D objects are then superimposed on the live camera preview and appear to be tightly coupled with the real world.

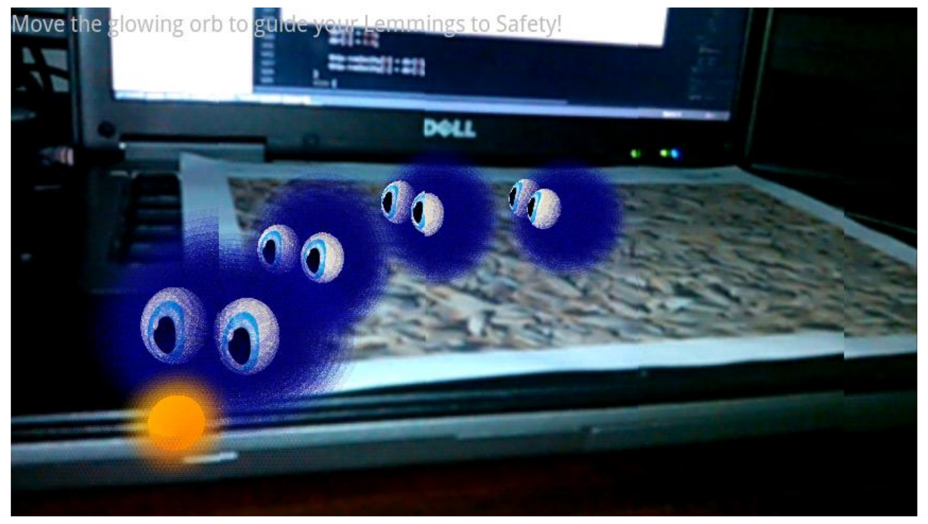
The goal of this project is to use commodity hardware and an openly available computer vision framework to create an interactive Augmented Reality game. The project consists of an Android smart phone and a printed card with a high contrast image. The card is used to calibrate the virtual 3D space to the image.

The game is comprised of small virtual creatures that appear on top of the printed card. The player must interact with the creatures to safely guide them through an environment projected onto the physical scene.

The project is designed to explore ways in which a virtual world can be combined with the real world. To emphasize this, the controls are designed to use the touch interface of the phone and be as unobtrusive as possible. There is no user interface. The user only sees the augmented world through the “magic lens” of the phone.

Objectives

The main objective was to explore computer vision technologies and OpenGl shader technologies in a mobile device.



Achieving these objectives required learning OpenGL ES shaders, and Programming for Android systems, including native development. The limitations of the device had to be taken into account because the nature of the application is very computationally intensive. Optimizations needed to be taken wherever possible. While mobile devices are continuously becoming more powerful, Optimization is still an extremely important concern.

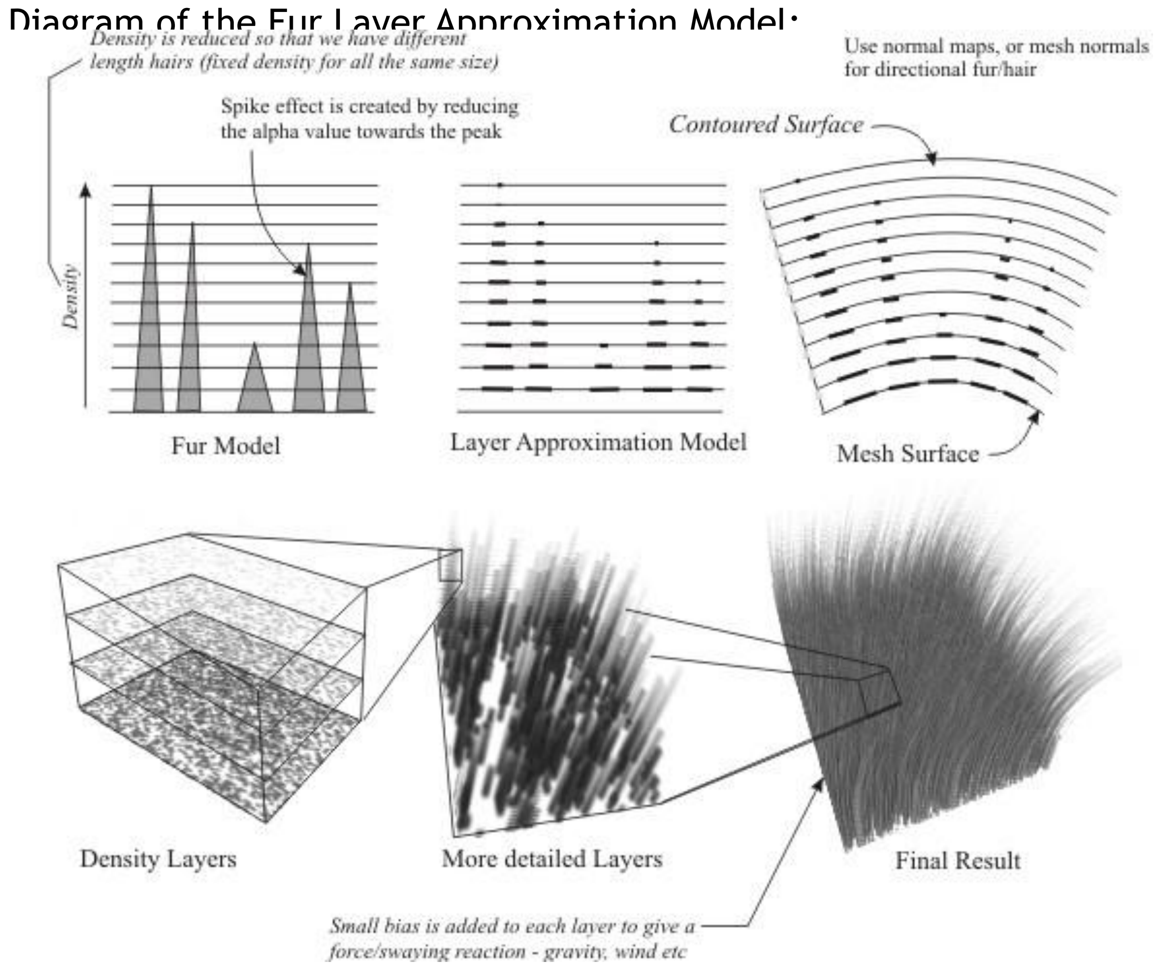
METHODS

The experiment was created with Java, C++, OpenGL ES, and the Qualcomm Vuforia AR SDK. The project was compiled for Android 1.4 using the Android Native Development kit to allow C++ programming on the Android System. The target phone was the Huawei Mercury M886. This phone has a 1.4 ghz processor, Adreno 205 GPU, 512 MB RAM, 2 GB storage, 8 MP camera, and 854x480 resolution screen.

There were two main phases of the project. The first implementing and calibrating the computer vision framework on the target phone. The second was creating a realistic Fur Shader using OpenGL GLSL ES.

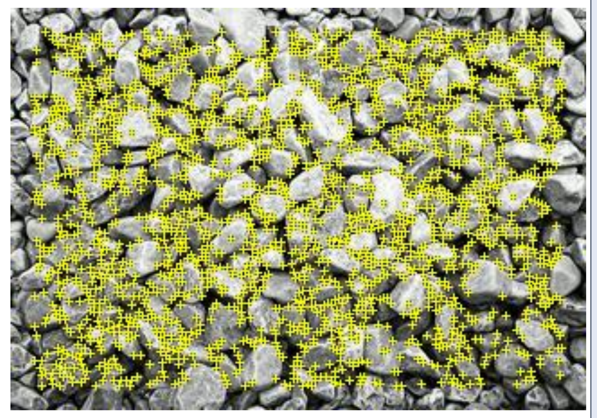
The Fur Shader was created using the layer approximation model. The original algorithm used 60 levels of approximations with a unique texture for each layer. This proved to be unacceptable for the target device. I was able to optimize the algorithm to use a single 512 x 512 texture and 12 layers. This was achieved by dynamically modifying the texture at each level inside the pixel shader.

Diagram of the Fur Layer Approximation Model:



The second phase of the project was the calibration of the Computer Vision Framework. The Qualcomm Vuforia AR framework is extensively documented a proved to be a very robust system. The system analyzes a special image that has been selected for high contrast and variation. This pre-processing is then stored as an xml file inside the phone. When the framework then uses the given data and the camera image to create a projection matrix that can be used with OpenGL.

The preprocessing phase of the Vuforia SDK:



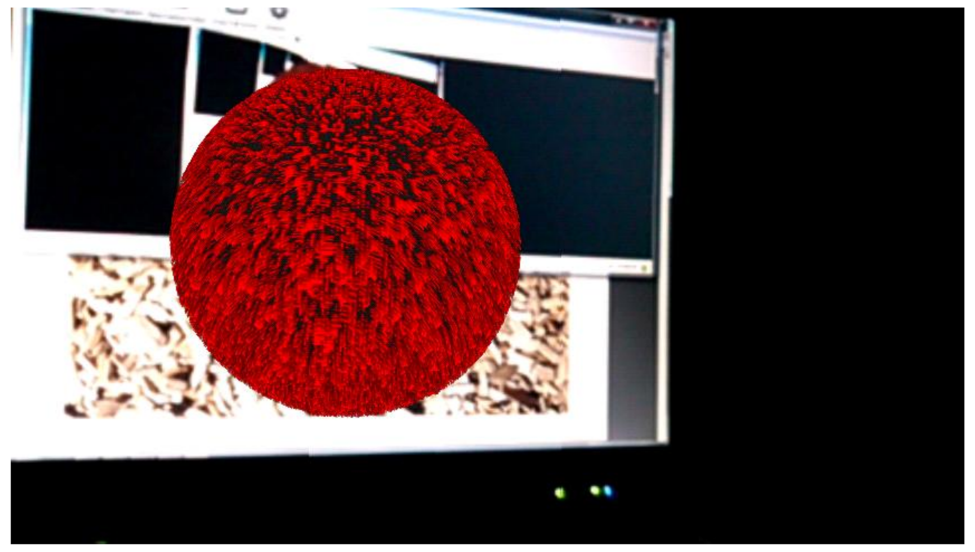
RESULT

The Fur Shader proved to be the most challenging aspect of the project. The algorithm was simple in theory but implementing it was more difficult than anticipated.

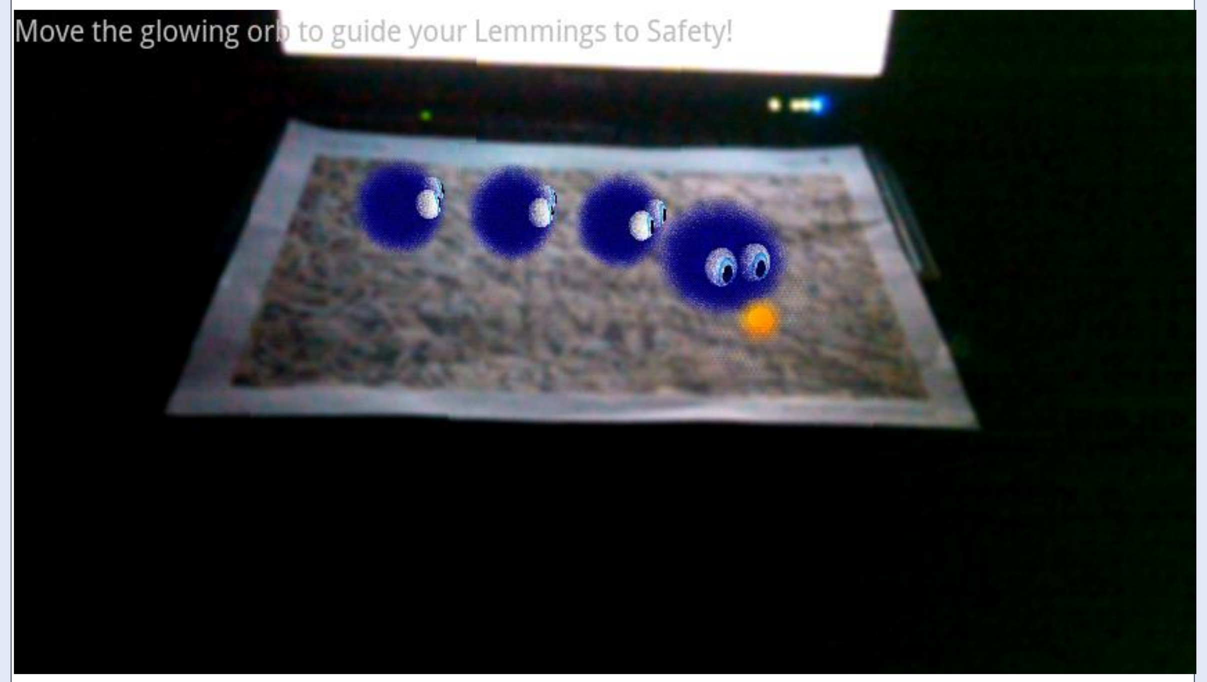
OpenGl Shaders are difficult to debug because they are compiled dynamically inside the OpenGL program. This was made even more difficult because the platform was a mobile device. The process of compiling to the device and processing the device log was tedius.

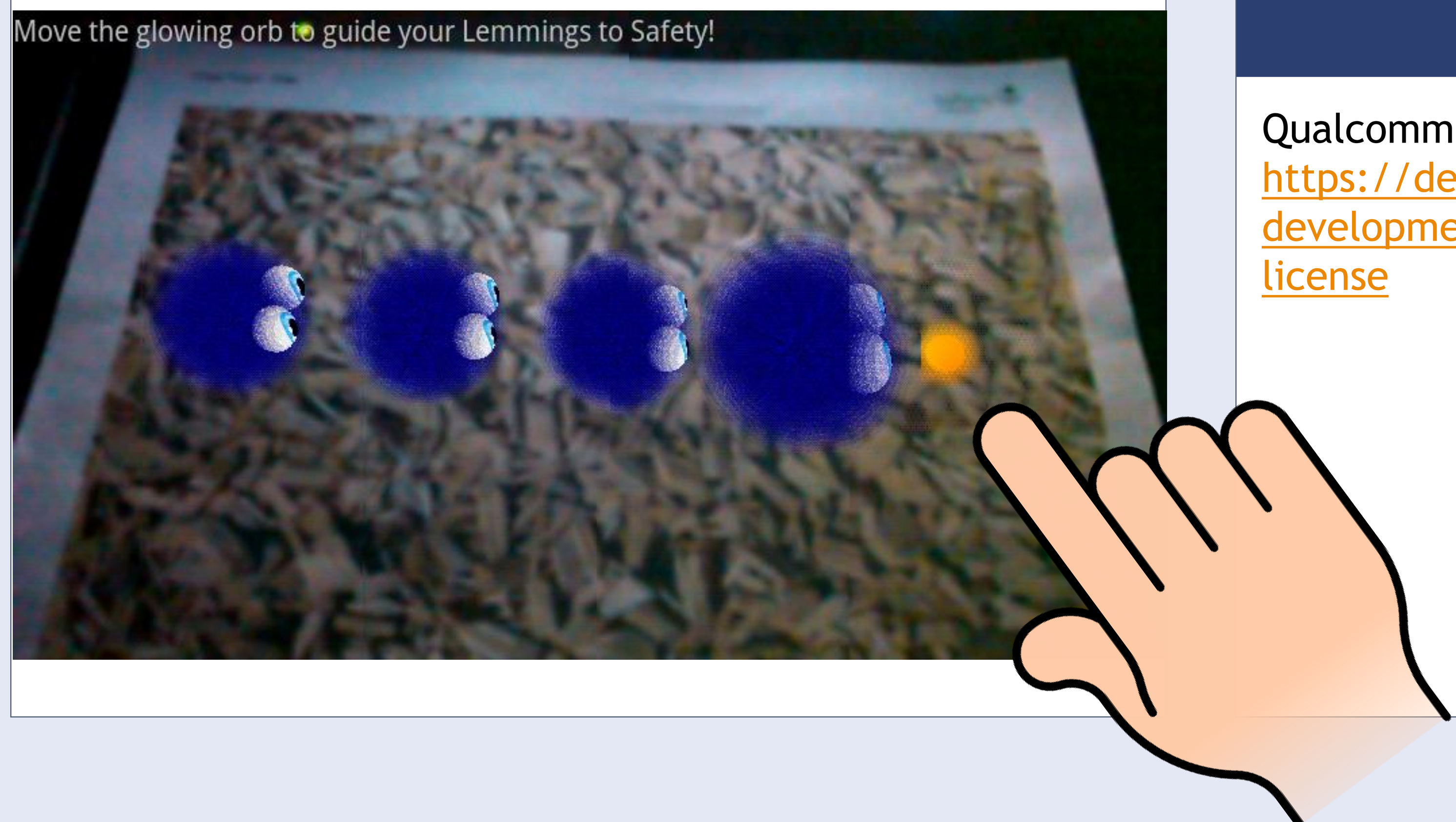
Once the shader was working the results were very effective. With the optimizations I was able to achieve a full fur simulation with reaction to force at real time frame rates.

Initial Fur Shader Results:



The simulation in Action:





CONCLUSIONS

The project achieved it’s basic goals of producing a Augmented Reality experience, but many of the initially proposed features were unable to be implemented due to time constraints and technical difficulties.

Future work might include adding an objective and obstacles to make the experiment more game like. Proper Z-Depth ordering of the rendering would reduce many of the transparency artifacts produced by the Fur Shader.

Other interaction methods may also be explored, such as tracking the hand position in the camera, allowing the user to interact directly with the virtual world, not just through the touch screen.

REFERENCES

http://www.xbdev.net/directx3dx/specialX/Fur/index.php

https://ar.qualcomm.at/developer_guide

http://www.opengl.org/wiki/

http://www.kickjs.org/example/shader_editor/shader_editor.html

Cozzi, Patrick. *OpenGL insights*. Boca Raton, FL: CRC Press, 2012. Print.

ACKNOWLEDGMENTS

Qualcomm Vuforia AR SDK:
https://developer.qualcomm.com/mobile-development/mobile-technologies/augmented-reality#ar-license