Image Based Modeling of Urban Scenes

Which one is real?*

Sebastian Olsson
Anders Stenberg
Mattias Stridsman
Antonios Vakaloudis

2004-03-31

*The image on the left is real

Table of Contents

1. Introduction..... 3
2. Image based modeling..... 3
  2.1. Evaluation of available software..... 3
    2.1.1. Facade..... 3
    2.1.2. Scenestealer..... 3
    2.1.3. Used Reality..... 3
    2.1.4. Photomodeler..... 3
    2.1.5. Canoma..... 4
    2.1.6. ImageModeler..... 4
  2.2. Creating geometry..... 5
  2.3. Generating textures..... 5
  2.4. Problems 5
3. Image based Lighting..... 6
4. Results..... 7

1 Introduction

The aim of this project is to use image based modeling, lighting and rendering to produce a short movie. The environment is captured with a set of still photos from a standard digital camera. Mirror sphere HDR images are used to light some synthetic objects in a correct manner. The environment was originally the corridors in Campus Norrköping's main building. We later changed the environment completely to an outdoor urban environment. We will use already available software for all the different parts, like modeling, rendering and the creation of HDR images.

2 Image based modeling

2.1 Evaluation of available software

Before we started to create geometry from the still images we considered a couple of different programs available.

2.1.1 Facade

Facade was never tested due to unavailability. It has a reputation as the "reference program" in the area though, and seems very powerful.

2.1.2 Scenestealer

Scenestealer was never tested since it was reported to be very unstable. We have no knowledge about the features of this program, but it is supposed to be based on Facade.

2.1.3 Used Reality

User Reality seemed to lack features we needed and when we looked at screenshots produced with this software we were not impressed at all.

2.1.4 Photomodeler

Photomodeler is a very feature rich and powerful program, with great accuracy in calculations, and easy to use reference point matching tools. It is focused on the CAD parts of image based modeling, which is both good and bad. On the upside, it is very exact and has good camera calibration utilities to improve the results. Features like automatic detection and matching of tracking points (reflector markers in the scene) probably improve the results a lot. On the downside, it is not very evolved in the area of actually making the models look good. Texture extraction is very primitive and it

shows that this part of the modeling is not prioritized. It also gives a very technical impression, which somewhat dampens the creativity.

2.1.5 Canoma

Canoma was the application we first chose to use for the project, since first impressions were very good. The interface is intuitive and gives quick response about what is going on, which was a great help when starting out. Still, when using the program for more complex scenes its downsides started to show. Too many constraints in the scene resulted in very slow and unstable updates of the calibration. The texture extraction also turned out to be a disappointment. The manual talks about advanced texture blending features for optimal extraction of textures from different images, but in the end it actually only uses one single picture for the texture. We don't know if the program is unfinished and not all of the features mentioned in the manual are implemented, or if it is just some very strange bug disabling the features totally. Nonetheless it made us look for alternatives to Canoma.

2.1.6 ImageModeler

ImageModeler is the application we finally ended up using for the project. It is part of the "Realviz" suit of photo and video applications, and is supposedly the leading application commercially available. Just like Canoma this gave a very good first impression; simple matching of reference points, texture extraction by blending from different images and custom geometry blocks. Unfortunately, just like Canoma, the application turned out not to be what it looked like. The texture blending had problems (explained in the "Generating Textures" chapter) and the calibration became very complicated in complex scenes (explained in the "Problems" section). They were not really ImageModeler's fault, although it could have handled it in much better ways. The worst problem with ImageModeler was the extreme amounts of memory and CPU power it used. The program actually used a lot of CPU even when it was passive and allocated very much memory even for simple projects. The behavior was different from computer to computer, which indicated some kind of bug or bad coding of the application. Saving and loading projects could take up to 5-10 minutes on reasonably powerful computers, which was extremely annoying. Exporting files for importing in other programs also was a big problem. Exporting as Wavefront OBJ or MaxScript both produced corrupt files from time to time, and VRML consequently created wrong transformations. One interesting point about the latter is that in ImageModeler 3.0 VRML export was working fine, which means Realviz has managed to break it for ImageModeler 3.5, which we were using. We are unsure if these problems were related to the "alternate evaluation" version of ImageModeler 3.5 we managed to get hold of, or if they were present in an original copy. Testing of the official trial version showed that the problems seemed to appear there too, which would indicate that it was not the alternate distributors fault.

2.2 Creating geometry

We finally decided to use Realviz ImageModeler. This program allows you to load several images taken from different angles, and by placing calibration points in the images the original camera positions can be determined. When the scene is calibrated, the geometry in the images can be reconstructed. ImageModeler has simple primitives like boxes, planes, spheres and cylinders. Mesh objects can also be imported into the scene. In our case the environment was buildings, which makes the modeling reasonably simple.

2.3 Generating textures

Textures can be generated from the still images. ImageModeler can produce a texture for a geometry surface in two ways:

1. When you want to generate a texture for the surface you can specify several images which you decide are relevant. The texture is created from the best image, which is the image with the highest resolution for that area. While the texture will be sharp, using this method can produce visible edges in the texture because different images are used for different parts of the texture, and they are not blended.

2. Use several images and blend them to create the final texture. While this produces a smooth texture without visible edges like in the method above, it is almost unusable since the texture will lack detail and the blending of the textures will not match since the calibration never is perfect.

There are often small areas which are not visible from any of the still photos, and these will appear as black areas in the generated textures. Filling these holes was done manually in Photoshop.

We also wanted to try to use moving textures in some scenes. We have a scene with an animated water fall. We solved this by filming the water fall from the same angle as one of the photos used to create the scene. The generated texture by ImageModeler of the water fall was then used as a template when the movie of the water fall was "warped" in After Effects by hand. The water fall texture image was then replaced by this warped movie, and while they don't match exactly it was close enough.

2.4 Problems

When modeling surrounding environments like floor, ceiling and walls you need to be able to see several planes (like a wall, the floor) in a single image. This is necessary in order to be able to calibrate the scene. This is also true when modeling a single object like a building from a distance or a cereal box on a table, but getting several planes in this case is much easier. Choosing photos which are possible to calibrate will not suit for texturing when the environment consists of narrow corridors like in our case since the viewing angle relative the walls are too low.

The original idea was to have the movie take place indoors at the Campus area. We discovered that the corridors we wanted to use was very hard to model since they were quite narrow. This made it hard to photograph them. We either got images which are possible to calibrate but useless for texturing or photos which would produce nice looking texture but they can't be calibrated.

We produced a short test movie showing a corridor and a pinball. From the selected viewing direction the textures were acceptable but from any other viewpoint it would look totally wrong.

Calibration in ImageModeler can be quite tricky. If a point is not correctly placed in an image you will get a calibration error, but finding out which point is wrong is almost impossible. If your calibration is correct, a badly placed marker will appear red, but if the calibration is not done correctly you don't know what markers are causing the problem. This means you don't have a chance to correct errors you make in the calibration process.

Another problem is reflection and specular that you get in windows. As soon as you start moving the camera in the virtual scene, those reflections will become incorrect. In order to get proper reflections you have to remove the reflections in the photos and then create reflectance properties for the windows in the rendering process. We added reflections in 3ds Max, but we didn't remove the original ones in the photos. Since we don't have any information of what's inside of the windows, we kept the original information and added some reflection on it.

Extracting a good ground texture turned out to be problematic. A helicopter or a sky lift of some sort would be required, but the lack of budget made this difficult. Almost all pictures were taken from ground level and that made the textures blurred. This was corrected somewhat in Photoshop using copy and paste of areas with adequate resolution. This means the shadowing from the original scene was sometimes lost, and had to be manually added.

3 Image based Lighting

The geometry and textures produced by ImageModeler was imported into 3ds Max and rendered without additional light sources. Synthetic objects were then added and rendered with HDR lighting. In order to get the final result we used differential rendering.

We captured light probe images on the sites we modeled in order to produce HDR images. The HDR images were then used to give the synthetic objects a correct lighting and to make them blend in. It was problematic to get good results. The default renderer in 3ds Max gave us a decent result, but was inconsistent and setting up a good scene took a lot of time and testing of different parameters.

4 Results

The result of our work is a demo-like short movie. The environment is the square outside Bomullsspinneriet, a nearby alley and nearby waterfall. We were able to produce scenes that looked almost photorealistic, but on the other hand we couldn't produce some of the scenes we wanted. During the project we learned a lot of the software that is available, its advantages and disadvantages. We can now say that we have a good insight in how they work and what they can do. We have learned which scenes are possible to produce and which are not.