Dynamic texture loading on zooming and panning for Hugin's fast preview

Darko Makreshanski

Electrical Engineering and Computer Science
Jacobs University Bremen
Campus Ring 1
28759 Bremen
Germany

Type: Hugin GSOC project proposal
Date: April 8, 2010

Abstract

One drawback of Hugin's current fast preview window is that the sizes of the textures directly depend on the number of images in the panorama. Thus, for a panorama with many images the texture quality is signifcantly decreased. This proposal suggests a project which will allow the fast preview to dynamically load textures with more detail.

The second part of this project covers a trimmed version of the project proposed in the preceding proposal for Panorama Overview.

Contents

1 About me

My name is Darko Makreshanski and I am currently a final year bachelor student studying Electrical Engineering and Computer Science at Jacobs University in Bremen, Germany. My major interests are the computer science part of robotics (artificial intelligence, machine learning, computer vision, etc) and information systems. After these bachelor studies I will continue to master studies at the Swiss Federal Institute of Technology in Zürich (ETHZ).

1.1 Coding

My main coding platform currently is Ubuntu 9.10, and a Windows Vista as dual boot and Windows XP on a virtual machine. My current machine is a Thinkpad T61 notebook (Intel Core 2 Duo @ 2.4GHz, 2GB RAM, NVidia Quadro NVS 140m).

I have worked with C/C++ in several courses and projects in and outside the university. My latest project in C++ is a framework for the Avahi Zeroconf implementation, which provides a high level, object oriented API for Avahi. This project is still in progress, and for evaluation purposes I have uploaded it to `http://rapidshare.com/files/373230640/avahi-framework.tar.gz`. I also have already worked with OpenGL in a related visualization course at my university.

1.2 Photography

My interests in photography have primarily originated in high school when I had extensive physics practice for competitions, and so from the interest in geometric and physical optics my passion for photography was born.

I currently own a Canon EOS 1000D, and before that I had a digital point and shoot Fuji. I have been photographing panoramas since I discovered Hugin couple of years ago. Some samples of panoramas and other general images I have taken are available at `http://dmakreshanski.deviantart.com`

1.3 Hugin and panotools

I have recently checked out and compiled hugin and the related panotools code. I have also gone through the code of the fast preview prior to writing this proposal during the discussion on the mailing list.

Currently I am also using panotools code for my bachelor's thesis. I am investigating a method for registration of terrestrial laser scans based on SIFT features

on the images generated from the reflectance values where am using panotools implementation for extraction and matching of SIFT features.

2 Motivation

Currently the texture quality in the fast preview directly depends on the number of images in the panorama. Thus, for panoramas with many images texture quality is significantly lower. Furthermore, the current system is unsuitable for zooming into the OpenGL scene since level of detail remains constant. The motivation for this project is to allow the fast preview to dynamically and parallely load textures with more details to improve its usability and to allow meaningful zooming in the preview.

Since the time estimated for this project is lower than the allotted time for GSOC, after finishing the above task the project will continue with implementing a trimmed version of the panorama overview project explained in the corresponding proposal.

3 Part One: Dynamic Texture Loading for Fast Preview for Zooming

3.1 Meshes

3.1.1 Inside test

The proposed system will be dependent on determining which images are visible at a given moment in the preview canvas. For this purpose, I propose a simple fast test method that will never return a false negative result. This involves computing a bounding box on each remapping of a mesh and afterwards with simple calculations decide whether the bounding box covers an area inside the viewing canvas.

The proposed bounding box would also be useful to cancel drawing of meshes that are out of the canvas area. The performance increase of this is questionable, however it would a feature that can easily be added.

3.1.2 Covering test

One other usable feature would be to test if a mesh is visible and not covered by other meshes, however I am currently not aware of a fast method that will garantee to never return false negative results. If such method appears it will be considered for implementation.

3.1.3 Mesh recalculation

Upon zooming and panning events all the visible meshes will have to be recalculated. For this purpose on each zoom event, or after passing a certain threshold upon zooming the meshes will be assigned a dirty state to be recalculated

3.2 Textures

3.2.1 Memory allocation

Without doubt, the loading of textures is currently the most time consuming part in the fast preview. Therefore I propose that the current scheme of allocating memory for textures remains as it is. Thus, the textures as they are created now will be fixed textures and will be changed only upon switching on and off the photometrics correction. The dynamic allocation of textures with higher details will actually create additional textures which will be created and removed upon paning and zooming. As current fixed textures have a amount of memory reserved for them, also another separate amount of memory will be reserved just for dynamical allocation and deallocation of the additional textures

3.2.2 Texture switching

Since texture loading is very time consuming, we can not allow for the user to wait for seconds on each zoom or pan event. Therefore the secondary textures with higher resolutions will be loaded in parallel in a separate thread and will be used instead of the "fixed" texture when they are loaded.

3.2.3 Parallely loading of additional textures

Loading textures in parallel will require solving some concurrency issues. First I propose that all additional texture loading is done in a single thread to prevent overcrowding the harddrive. Afterwards the main thread and the parallel (additional texture loading) thread will communicate via a synchronized map which will store which textures will need to have texture detail increased. The main thread will update this map upon zooming and panning events, and the parallel thread will load the corresponding additional textures. After the the parallel thread finishes its job and loaded textures correspond to the textures that need to be loaded the canvas is redrawn.

3.3 Mesh subdivision

To increase the efficiency of the proposed method, the system will be refactored to subdivide meshes and handle these separately. Therefore, in this case these subdivisions will have separate bounding boxes, textures and display lists which will increase the effect that the proposed method will introduce.

4 Part Two: Panorama Overview

Since the estimated time for this project is smaller than the time available for GSOC, after finishing the above project I will continue to implement parts of the project for panorama overview proposed in the corresponding proposal. Since the time available for finishing the above project can not be exactly determined, the exact level of details that the following project will include can not be known in advance. The initial estimate is that the following project will include a basic interactive panosphere overview as explained in the proposal without the interactive plane in mosaic mode and without the animations section.

5 Timeline

15.05.10 University obligations end and full time work for the project begin.

25-30.05.10 Strict deadline for the design of the architecture of the system.

30.06-10.07.10 Finished dynamic texture loading for fast preview

09.08.10 Basic interactive panosphere visualization completed

09-16.08.10 Improve code quality, write documentation, etc

20.08.10 Final evaluation deadline

16.08-15.09.10 Using remaining free time until the begin of school obligations to improve the system, add remaining features, etc.

I am planning to work in most cases more than 40 hours per week, since I will assume weekends also as work days. I will be unavailable for a few days at the beginning of June (4th-7th) due to graduation ceremony and traveling to home. I may also be unavailable for about a week in July, when I might take a nice summer holiday in the beautiful lake Ohrid.