3D Visualization of Mars Satellite Images

March 21, 2003

Ramkrishna Chakrabarty
Mentor: Dr. Alan Shih

Graduate Student
Department of Computer and Information Sciences
University of Alabama at Birmingham

Student Assistant
Enabling Technology Laboratory
Department of Mechanical Engineering
University of Alabama at Birmingham
Problem Definition:

To visualize 2D monochrome high resolution images of Mars in 3D.

> Given are some satellite images of the Mars surface.

> The objective is to use the data available from the images like luminosity etc to create a 3D visualization of the same.
Top Level Block Diagram

Resize Image → Generate Heightmap → Visualize Heightmap in 3D using OpenGL
Image Resize Algorithms Used:

- Bilinear Interpolation
- Bicubic Interpolation
Bilinear interpolation linearly interpolates along each row of the image. Then uses that result in a linear interpolation down each column in the image. With this method, each estimated pixel in the output image is a weighted average of its four nearest neighbours in the input image.
The **bicubic interpolation** estimates the color at a pixel in the destination image by an **weighted average** of 16 pixels surrounding the closest corresponding pixel in the source image.

The bicubic method yields a **smoother** result since its convolution kernel is of size 4x4.

This is a computationally intensive resizing method, but produces **better** results when there is **drastic change** in dimensions.
Bilinear/Bicubic Comparison

Original Image

Resize factor = 1/4
What is a heightmap?

Heightmap it's a set of numbers arranged so that they form a two-dimensional grid, like a bitmap, except each number represents a ground elevation instead of a color.
Heightmap Generation Algorithm

> First, traverse the whole image pixel by pixel at the *angle of the sun*. This is possible due to the fact that the image has been converted into a *continuous space* after bicubic interpolation.

> Then, the height of each pixel is estimated from
  > Grayscale value of each pixel of the original image.
  > Grayscale value of the *previous* pixel in the path.
  > Grayscale value of the *next* pixel in the path.
Once the heightmap is generated, the same is fed into the **OpenGL based visualization module** to render it in 3D.

This program also does the **lighting** and is **interactive**.
And the result...
And the result...
...more results
On the ETLab VisBox [Stereoscopic Display]
Scope for Future

> The algorithms can be improved. This is a starting point and more research needs to be done to develop new and better algorithms.

> Error checking: the application should be able to distinguish between dark objects (rocks etc.) and shadows.

> One of the ideas is to traverse the image at an other angle than the sun's angle, and find out the gradient differences to figure out shadows etc.
Thank you...