3D Scanning and Stereoscopic Re-rendering

6.111 Project Proposal

Adam Gleitman
Andrew Shum
Timur Balbekov

Overview

3D scanning entails analyzing a real-world object to collect data on its physical properties, which are then used to construct digital, three dimensional models. The collected 3D data is useful for a variety of industrial and human-computer interaction-related applications, including product design, video game production, orthotics and prosthetics, and quality control and documentation of artifacts.

In this project, we focus on rendering 3D imagery from 2D images of objects captured using two NTSC cameras that are positioned a few inches apart to mimic a pair of human eyes. The advantages of using a passive camera system are low cost and minimal mechanical complexity, as compared to active radiation-based depth mapping equipment systems.

We design our system to function in two different rendering modes. In the basic mode, 2D images are combined and filtered on the FPGA to render an anaglyph image that is viewable using red and cyan 3D glasses. In 3D scanning mode, we focus on capturing 2D images of polyhedra with color-coded vertices and edges. An interactive 3D model is then rendered on the monitor using vertex/edge detection and 3D rendering hardware implemented on the FPGA.

Functionality

Overview

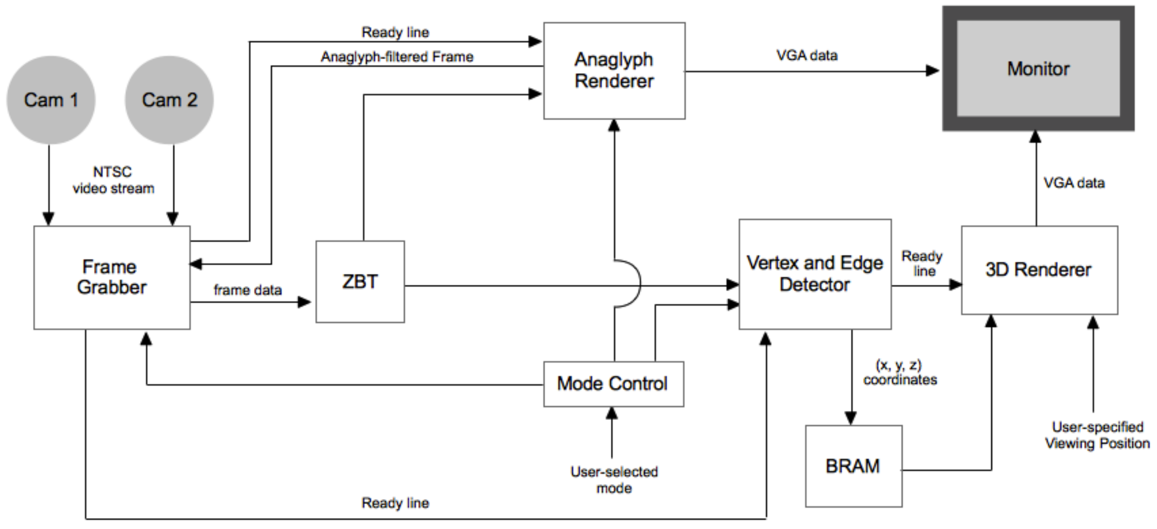

Figure 1. A high-level block diagram of the processing pipeline

The 3D scanning pipeline works in three conceptual stages. The first frame grabbing stage processes the incoming video streams and stores frames to a front frame buffer on the on-board ZBT memory. This stage controls the NTSC decoder chip and deinterlaces the input signal, as well as arbitrates memory access to the ZBT. In anaglyph rendering mode, the frame grabber writes the red-cyan anaglyph to the front buffer. In 3D rendering mode, the frame grabber writes the left and right video streams to the front buffer. The second stage vertex and edge detector works only in 3D polyhedron capture mode. It finds color coded edges and vertices in the left and right frame buffers, transforms the points to 3D coordinates, and stores the data in an on chip BRAM. The third stage generates and renders the anaglyph image in anaglyph mode, and renders the scanned wireframe in 3D capture mode. The 3D renderer takes a camera perspective input, allowing the user to view the 3D object from different positions.

Module Descriptions

Frame Grabber

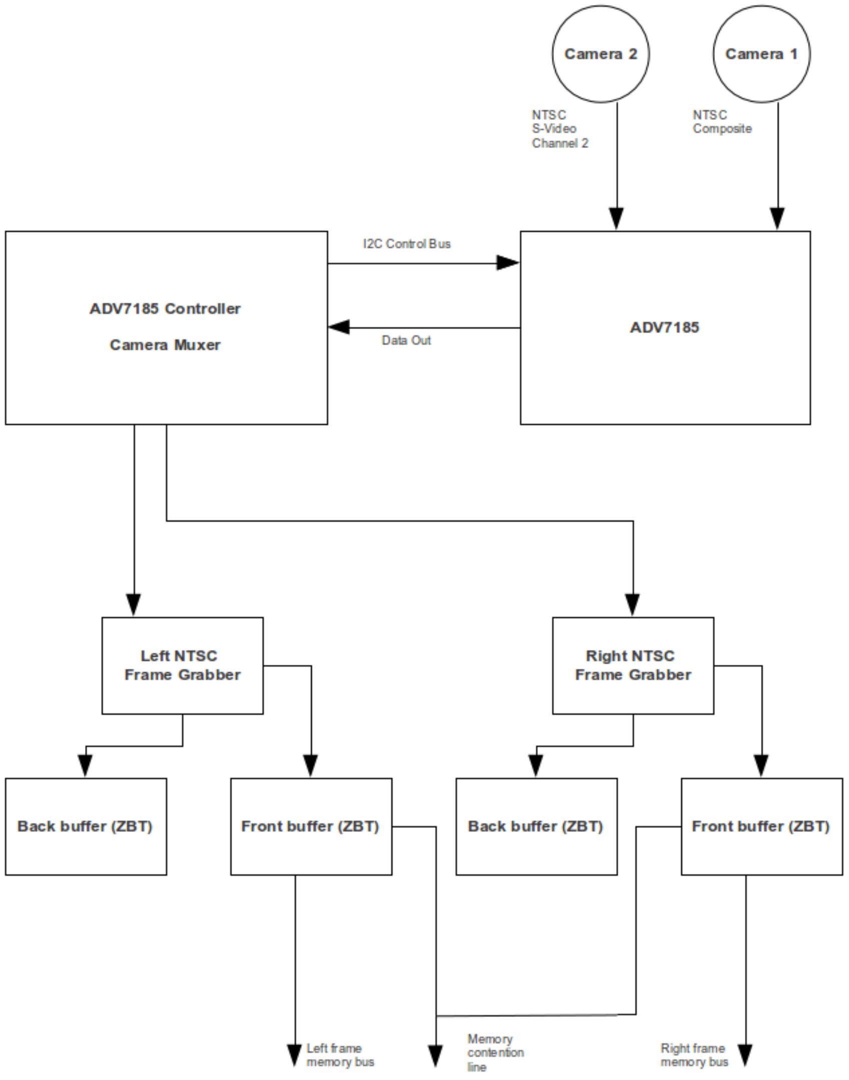

Figure 2. Block diagram for the Frame Grabber module implemented on a single labkit.

NTSC Decoder Controller

The NTSC decoder module generates control signals for the onboard video decoder chip. This module is based on the staff provided NTSC capture code, but is modified to capture two video feeds simultaneously. This is possible by processing image data from one camera at a time, and switching between cameras at the expense of frame rate. There are three analog video input lines that are physically accessible on the Labkit: composite, S-Video chrominance, and S-Video luminance. The composite video signal of the second camera will be connected to the chrominance input of the S-Video connector, and the appropriate configuration bits will be set in the video decoder. A register internal to the decoder chip selects which video stream is selected. By modifying the contents of this register, we can configure the video decoder to selectively convert one of these channels at a time.

The controller captures two streams at a reduced frame rate by alternating the incoming video stream. The NTSC standard specifies a 24 frame per second (FPS) frame rate for the interlaced signal that is generated by the cameras. Since the video decoder is shared by two video feeds, it is not possible to capture video at the full frame rate. The subsequent image processing modules do calculations on a whole frame, so the input video stream must be deinterlaced. De-interlacing (which is performed by the frame grabber modules) halves the frame rate, and switching the input feed discards an input frame. The resulting frame rate is 6 FPS at the original 640x480 resolution.

Left and Right Frame Grabbers

The left and right frame grabbers will take the interlaced frame data as input and store a whole frame in the front buffer. The NTSC decoder controller module enables these modules alternately, so only one of these modules is operational at a time. To deinterlace the input signal, the frame grabber records two interlaced video frames into the back buffer. After the entire frame is recorded, the frame grabber writes the contents of the back buffer into the front buffer, where the frame is accessible to subsequent modules. The active frame grabber outputs a pulse on the ready line to signal subsequent modules to begin processing.

In anaglyph mode, the frame grabber will apply the anaglyph transformation by passing the left and right pixel values to the Anaglyph Filter module. The anaglyph transformation result will then be stored in the front buffer.

This module and the preceding NTSC decoder will be tested in a testjig that outputs the contents of the front buffer to a VGA monitor, with a user selectable video feed switch.

ZBT Memory Controllers (front and back buffers)

There are two one-port ZBT memory chips that allow for 36 bits to be written or read in one clock cycle. Since the frame grabber needs time-critical access to the back buffer, both left and right back buffers are reserved to the frame grabber modules. The front buffer is stored on the second ZBT chip, and access to the chip is shared by the vertex and edge detector and the anaglyph renderer. Since this chip implements a one-port memory, all modules attempting to access the memory need to pull the memory contention line high before a read or write operation. The ZBT memory controller will operate at a faster clock frequency than the rest of the modules to accommodate for timing constraints.

If the maximum synthesized clock speed does not allow live video feed, we have several options to optimize the design. We can implement a read/write BRAM cache in non-timing critical modules (the frame grabber and VGA renderer have absolute memory access priority to the back and front ZBT memories respectively). Alternatively, we can slow down the target frame rate until timing constraints are met by periodically grounding the frame grabber enable lines.

The ZBT controller will be tested using the logic analyzer as a standalone module.

Vertex and Edge Detector

The vertex and edge detector is responsible for taking frames captured by the cameras, finding the vertices and edges in the polyhedron being photographed, and constructing the data for the wire-frame model to be written into a BRAM.

Each vertex found will be given an index between 1 and *N*, where *N* is the number of vertices. Vertex 0 will be the topmost vertex in the camera frame, and vertex *N* will be the bottommost vertex in the camera frame. If two or more vertices are found to have the same vertical position, the leftmost vertex will be assigned the lower index. This method of indexing gives both the vertex deprojector and the edge determiner an unambiguous way of referring to each of the vertices. Since the two cameras are very close to each other, it is relatively safe to assume that this ordering will be consistent in both images. If the ordering happens to be not consistent, our algorithms should still produce an interesting effect.

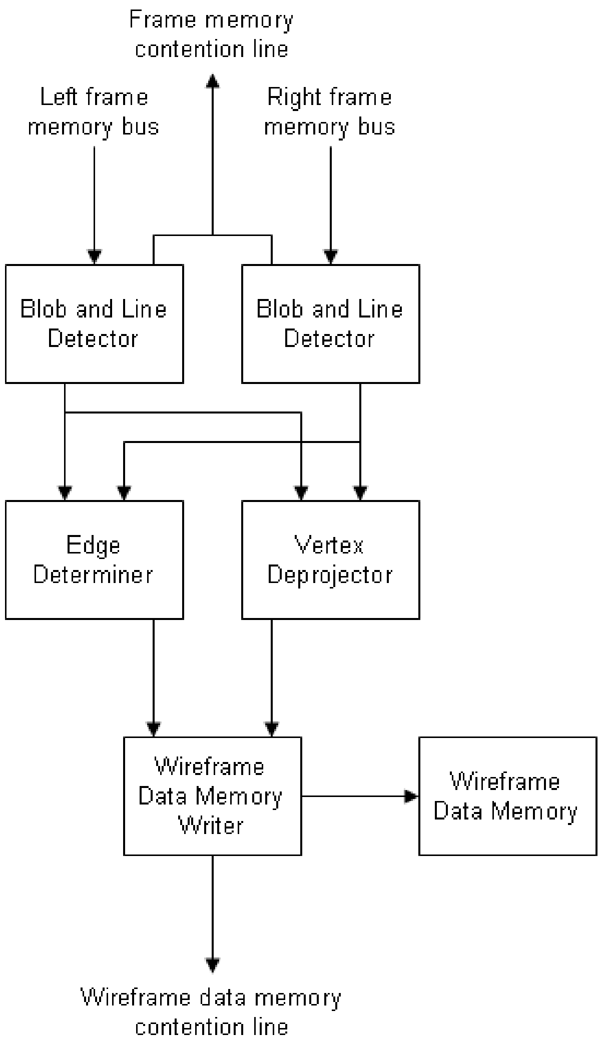

Figure 3. Data flow and modules within the Vertex and Edge Detector module.

Blob and Line Detector

The blob and line detectors are responsible for processing image data from the frame memory modules and determining the locations of the vertices and edges on the two camera frames. There are two of these modules, one for the left camera frame and one for the right camera frame. The information collected is available to the vertex deprojector and edge determiner through the appropriate control signals.

Vertex Deprojector

The vertex deprojector is responsible for performing the necessary mathematics to calculate the locations of the vertices in three-dimensional space from the two-dimensional coordinates fed to it from the blob and line detector. This module knows the algorithms and appropriate equations needed to carry out these calculations depending on the positions of the cameras. The vertices found will be available to the wireframe data memory writer through an input and several output buses. The input will be a vertex index *i*, and the output will be the coordinates of vertex *i* in 3D space.

Edge Determiner

The edge determiner is responsible for determining which pairs of vertices are connected by an edge. This module will take in line segment information from the blob and line detectors and determine whether or not any two vertices are connected. This information will be available to the wire-frame data memory writer through two inputs and an output. The inputs will be two vertex indices *i* and *j*, and the output will be high if the corresponding vertices are connected by an edge, and it will be low if they are not connected.

Wire-frame Data Memory Writer

The wire-frame data memory writer is responsible for writing the wire-frame data it receives from the vertex deprojector and the edge determiner into a BRAM that the 3D renderer can access. Address 0x0 contains *N*, the number of vertices found. The next *N* addresses contain the coordinates of vertices v_1 through v_N. A later address, say, 0x100, contains another number *M*, the number of edges found. The next *M* addresses contain the indices of the vertices that are connected. For example, if the BRAM is 48 bits wide and there is an edge connecting v_1 and v_3, then address 0x101 may contain 0x000001000003.

The wire-frame data memory writer also outputs a "ready" line, which goes high once all the data from the frame memory buffers has been processed.

Anaglyph Renderer

Rendering an anaglyphic image involves applying separate color filters to two input images. The left image, which is viewed through a red filter, has its blue and green color components are reduced, while the right image, which is viewed through a cyan filter, has its red component reduced. The composite image is then composed by combining the remaining color components into a single color image. When viewed through the red-cyan 3D glasses, the visual cortex of the brain fuses this into perception of a three dimensional image.

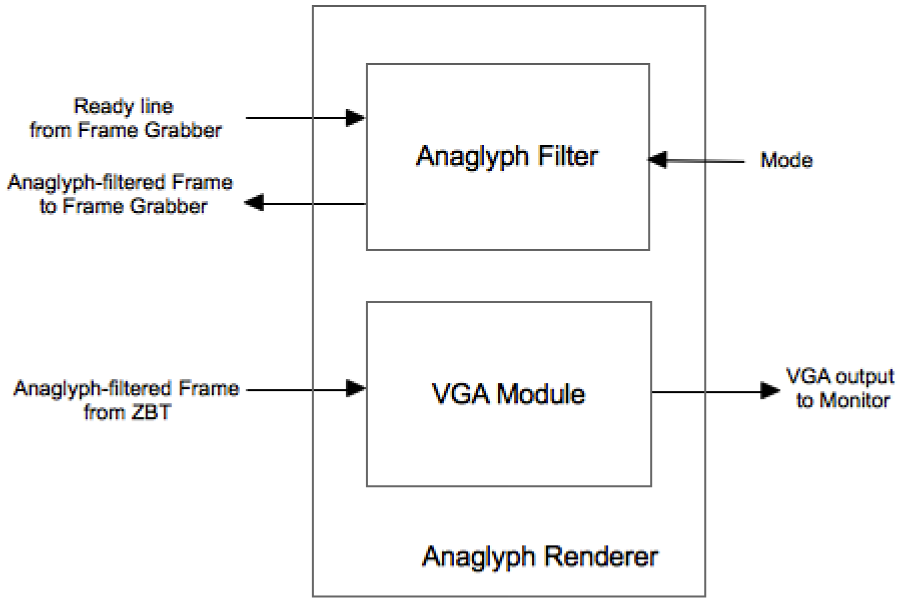
Ready line from Frame Grabber
Anaglyph-filtered Frame to Frame Grabber
Anaglyph Filter
Mode
Anaglyph-filtered Frame from ZBT
VGA Module
VGA output to Monitor
Anaglyph Renderer

Figure 4. Data flow and modules within the anaglyph renderer.

Anaglyph Filter

As long as the Frame Grabber delivers a signal high on the ready line, this module pulls two frames from the front buffer of the second ZBT chip and passes them through an anaglyph filter according to the Optimized Anaglyphs method of [1]. The method works as follows. For a pixel on the left frame with RGB components $\mathbf{v}_L = [r_L, g_L, b_L]$ and the corresponding pixel on the right frame with RGB components $\mathbf{v}_R = [r_R, g_R, b_R]$, the resulting pixel that is rendered has RGB components

$$\mathbf{v}_{\text{anaglyph}} = [0, 0.7, 0.3]\ \mathbf{v}_L^T + [0, 1, 1]\ \mathbf{v}_R^T$$

The output of this module is a single frame that is pushed back into the front buffer as dictated by the Frame Grabber module.

VGA Module

The VGA module is responsible for providing a constant feed of the anaglyph-filtered image we are capturing. Thus, it will be reading the anaglyph frame located on the front buffer and sending it to VGA at 60Hz refresh.

These modules can be tested by loading test frames on the ZBT and attempting to output an anaglyph-filtered image on the monitor.

3D Renderer

3D Projection Transformer

This module is responsible for pulling the vertex coordinates and edge list from the BRAM, as well as the user-specified Viewing Position, and performing all the necessary matrix operations to output a 2D image of how the polyhedron appears as seen from a virtual camera positioned at the user-specified Viewing Position.

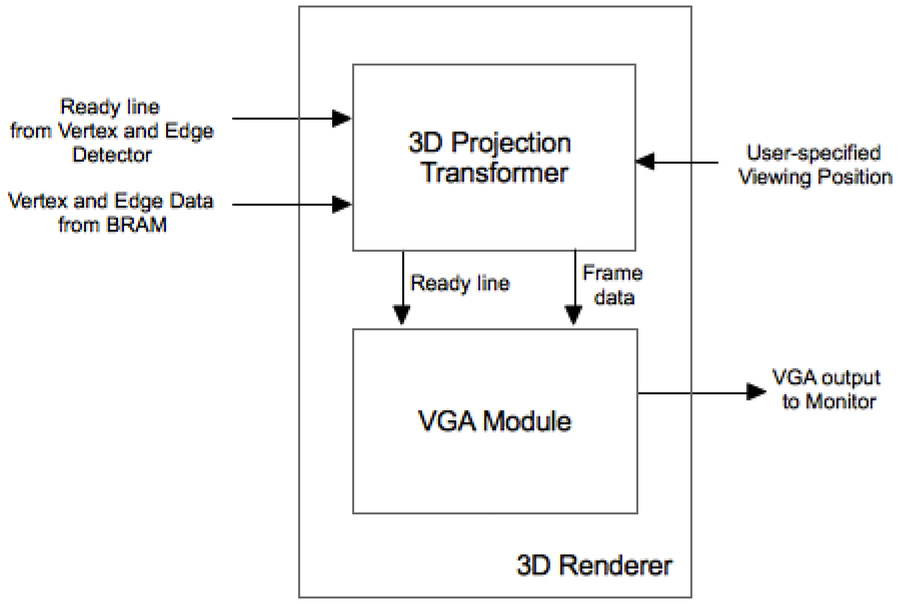
Ready line from Vertex and Edge Detector
Vertex and Edge Data from BRAM
3D Projection Transformer
User-specified Viewing Position
Ready line
Frame data
VGA Module
VGA output to Monitor
3D Renderer

Figure 5. Data flow and modules within the 3D renderer

VGA Module

Like the VGA Module in the Anaglyph Renderer, this VGA module is responsible for pushing out the 3D projection whenever 3D Projection Transformer is ready to display a new frame.

These modules can be tested using synthetic vertex and edge data and a testjig module, as well as attempting to output an interactive rendering of the hard-coded polyhedron to the monitor.

Organization

Division of Work

The work will be assigned according to the three conceptual modules. Tim will work on the Frame Grabber, Adam will work on the Vertex and Edge Detector, and Andrew will work on the Anaglyph Renderer and 3D Renderer.

Milestone Timeline

1. Implement the anaglyph generation mode
    - Video capture and ZBT controller module (Tim)
    - Anaglyph renderer and wireframe simulation (Andrew)
    - Vertex and Edge detector simulation (Adam)

By end of Week 3

2. Implement the 3D conversion and rendering mode
    - Integration (Tim)
    - Wireframe renderer (Andrew)
    - Vertex and Edge detector (Adam)

By end of Week 5

External Components

Two NTSC compatible cameras with composite video output are required. These are supplied by the 6.111 staff. Additionally, red-cyan glasses are needed to view the generated anaglyphs. These are relatively inexpensive and will be purchased online.

References

[1] Anaglyph Methods Comparison. [Online]. Available from: http://3dtv.at/Knowhow/AnaglyphComparison_en.aspx. Accessed 2011 Nov 1.