CP411 & CP467 Final Project

# PROJECT IT5000

Tyler Potter
Kirill Stepanchuk

An N2085 Production

## Real Virtuality

Problem:

Capture the position and motion of an actor in real space and represent as a model in a virtual environment

Solution:

Layered application to process live video data, extrapolate 3D position, and animate in a virtual space

## The Actor

- An actor is outfitted with a set of white control balls, strategically placed.
- Different actors with different control point configurations are available for use.
- 5point Stickman, The Hand, etc.

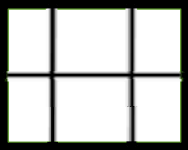
## Image Processing

- Platform
  - Stereo cameras
  - Matlab with The Image Acquisition Toolbox
- Process
  - Calibration
  - Frame Buffering
  - Control Point Extraction
  - Triangulation

## Image Processing

Calibration

A Calibration board is held in front of the cameras. The cameras focus points have the same separation as the camera lenses. This aligns the cameras in parallel



## Frame Buffering

- Each camera device submits frames to the imaqTool at the device's independent frame rate
- Incoming frames from each camera are kept in separate frame buffers
- Processing command is triggered by external application to retrieve and process top frame on each buffer as a pair

## Control Point Extraction

- Various combinations of morphological operations were used while determining suitable sequence
- Processing steps:
  - Convert to grey scale using blue component
  - Threshold image (dynamically chosen threshold)
  - Majority Filter (reduce noise and isolated pixels)
  - Shrink (iteratively shrink regions until region is single pixel)

## Position Extrapolation

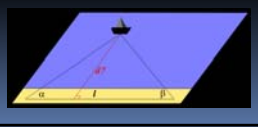
- Platform
  - Matlab
- Process
  - Match Points between frames
  - Triangulate position
  - Adjust for distance attenuation

## Point Matching

- Control points (reduced to pixels) are sorted by their x co-ordinate in each frame.
- Points with the same sorted index are identified as representing the real control point
- Frames not having equal number of pixels are discarded

## Triangulation

- Distance between observation points is known (separation of cameras)
- Observation angels are determined using relative position and camera view angle
- Result of calculation gives the z co-ordinate of each control point



## Position Adjustment

- Relative distances appear larger as the real objects move closer to a lens
- Using the calculated depth, the x and y co-ordinates from one camera feed are adjusted to produce the 3D location of control point
- Set of control points for each frame are stored in Matlab Workspace

## Animation

- Platform
  - C++
  - OpenGL
  - Matlab C++ Engine Interface
- Process
  - Run commands on Matlab interface and retrieve resulting data
  - Orient and Update control points of virtual model
  - Render positioned model in virtual space

## Matlab Commands

- Matlab engine interface allows instruction to Matlab command line from c++ application
- Image processing and triangulation methods are called from separate thread.
- Resulting sets of points are placed into queue for model to retrieve

## Model Orientation

- Orientation pose used by actor to match processed points to points on model
  - Point sorting and matching is different for each model type
- Model updates position of control points by comparing incoming co-ordinates with previously mapped points
- Points with the shortest separation from one frame to the next are matched, and the model's control point location is updated

## Rendering

- Models have a different control point configuration
- Models can be rendered in a number of ways including
  - Points only
  - Points and Lines
  - Points and Cylinders,
  - Pretty (more complex polyhedrons)
- Model rendering is custom for each model type