RAPTOR Progress 7/22/2014

Sam Kelly
Jeff Byers

Redesigned Pipeline

- Removed old messy research code
- Replaced with organized interfaces and classes
- Now maintain commits in a local git repository stored on gp1
- New modular design allows for easy configuration of the experiments that will be required for Plans A, B, and C

Training Data Generator:
Python/Blender (rendering script) → Java (image processing, covariance matrices) → annotation file

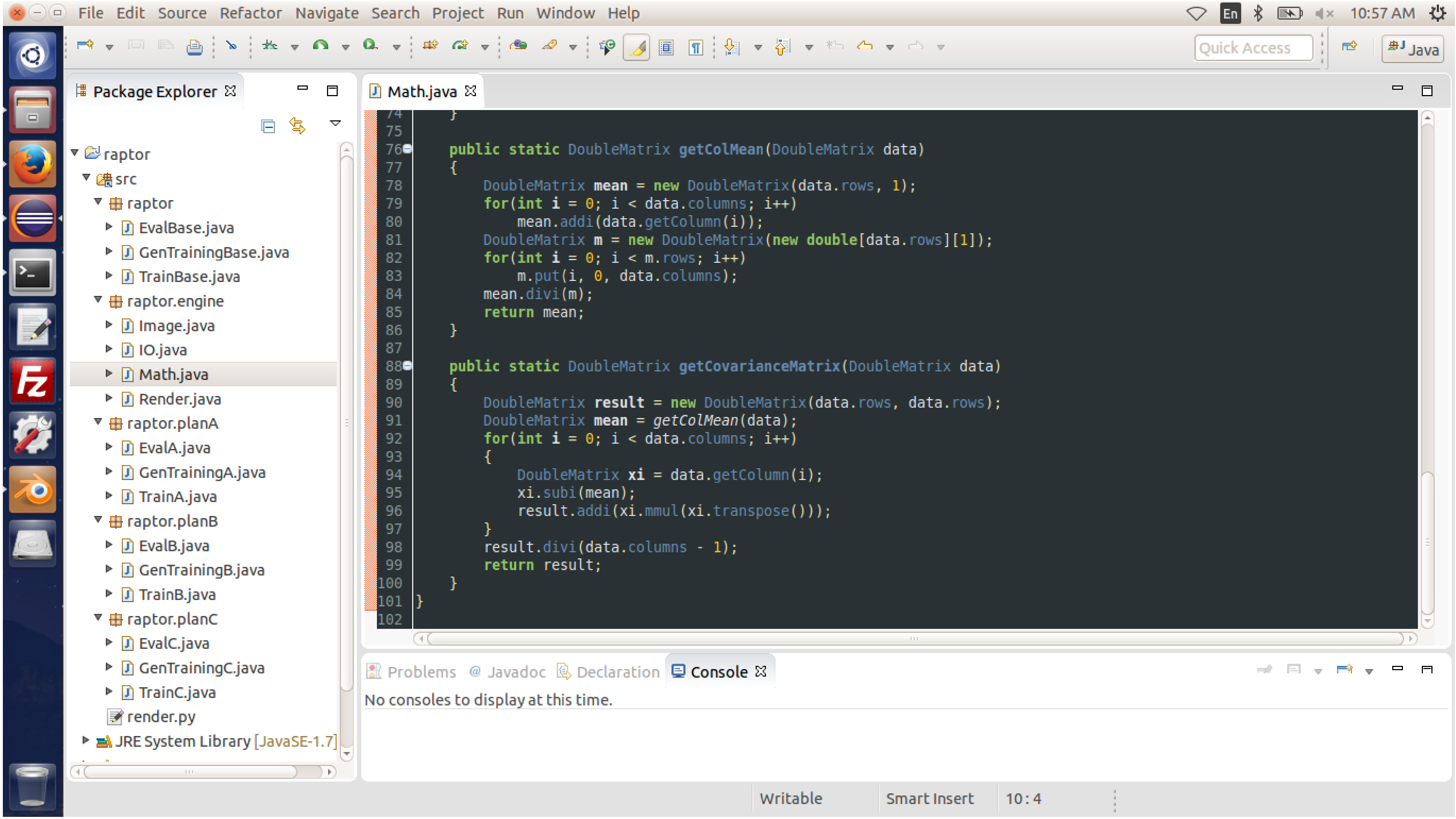
Training:
Java (training program) → training → neural network (Plan A) or data structure (Plan B)

Plan C: Java (search program) ↔ Python/Blender Pose Renderer

Redesigned Pipeline

- Now uses Jblas for matrix operations and eigenvalue calculation (before used a slower, hand-coded solution)
- Now uses Imgscalr for image resizing in Java instead of built-in libraries (faster, higher quality)
- Still using Heaton Research's Encog for feed-forward ANNs
- Training data generator now fully parallel – takes full advantage of all of gp1's cores when running

Redesigned Pipeline



```java
    }

    public static DoubleMatrix getColMean(DoubleMatrix data)
    {
        DoubleMatrix mean = new DoubleMatrix(data.rows, 1);
        for(int i = 0; i < data.columns; i++)
            mean.addi(data.getColumn(i));
        DoubleMatrix m = new DoubleMatrix(new double[data.rows][1]);
        for(int i = 0; i < m.rows; i++)
            m.put(i, 0, data.columns);
        mean.divi(m);
        return mean;
    }

    public static DoubleMatrix getCovarianceMatrix(DoubleMatrix data)
    {
        DoubleMatrix result = new DoubleMatrix(data.rows, data.rows);
        DoubleMatrix mean = getColMean(data);
        for(int i = 0; i < data.columns; i++)
        {
            DoubleMatrix xi = data.getColumn(i);
            xi.subi(mean);
            result.addi(xi.mmul(xi.transpose()));
        }
        result.divi(data.columns - 1);
        return result;
    }
}
```

No consoles to display at this time.

Occlusion

- Plans A and B both require finding a mapping between the 2 x 2 covariance matrix for a 2D image of a pose, and the 3 x 3 covariance matrix for the 3D point cloud of that same pose
- Better correspondence between 2 x 2 and 3 x 3 matrices could be achieved if occluded vertices were excluded from our 3D point clouds
- Might try to hook in to Blender's occlusion engine for this info, though this is proving difficult
- A variation of the Painter's algorithm should be well-suited for this problem since we don't (shouldn't) have to worry about cyclic overlap or piercing polygons
  - After z-ordering of faces is obtained from the Painter's algorithm, for each face, perform perspective projection on each vertex resulting in a 2D projected version of each face
  - Next store all the polygons in a Polygonal Map Quadtree (*Storing a Collection of Polygons Using Quadtrees*, H Samet 1985)
  - Now for each vertex in the model, find all polygons that intersect with that vertex using the polygonal map quadtree
  - if any of these polygons is closer (based on Painter's algorithm output) to the viewport than the vertex, exclude this vertex
  - Should be very efficient since we are not actually rendering the faces using this method – we are just finding which vertices are occluded, which is much simpler than the full occlusion problem faced in computational geometry and 3D rendering
  - Should run in O(n*log(n)) time with a mild constant factor where n is the number of vertices

Subdivision

- Subdividing the model so that there are more than one vertex on each edge might also improve correspondence between 2 x 2 and 3 x 3 covariance matrices
- Model can easily be subdivided using a Blender routine exposed by the Python API
- Will explore this

New 3D Ship Models

- David has provided us with $1893 worth of high quality military ship models
- Models are supposedly highly accurate, used by producers within the CG community (these are the same models used in movies, etc.)
- Have models of Russian, Chinese, Canadian, and U.S. military vessels

Future Work

- Plan to work remotely as a student contractor part time with NRL to continue this project for the next year while I'm in Baltimore (this is still preliminary / being worked on by David)
- Barring that, plan to continue work on RAPTOR on my own
- If we obtain good enough results, will try to publish a paper on this work
- Will likely only have tentative results by the time my NREIP term ends, but definitely appropriate for a 10 week internship
- Have only scratched the surface of what might be possible with this sort of model-driven approach to pose estimation