Visualizing Binary Trees Embedded in Hypercubes

CPSC 533C Project Update

James Slack

March 19, 2003

Motivation

- Binary decision trees are prevalent
- Most multiprocessor systems modelled after n-dimensional hypercubes
- Does there exist a static assignment of a binary tree with 2^k nodes in a k-dimensional hypercube?
  - Communication in a hypercube of processors

Problem Statement (not the visualization problems)

- The embedding problem:
  - Assign nodes of a binary tree to nodes of a hypercube of equal size with no dilation
  - Problem is NP-Complete [Wagner]
  - Parallel methods used last term
  - Stochastic methods used this term
  - Build and embed random binary trees

Problem Statement (the visualization problems)

- Can my embedding visualization be used to create more intelligent embedding heuristics
- Show 2 embeddings of the same tree
  - Show embedding of either tree
  - Show the differences in the hypercube
  - Show similarities in the hypercube
- Visualization will (hopefully) lead to insight

Status

- Using Geomview and Java, can display trees with coloured embeddings
- Two embeddings can be compared
- An entire hypercube of 6 dimensions can be shown with no occluded edges
- A partial hypercube of an embedding can be shown with same structure

Current solutions

- Drawing a 6 dimensional cube in 3 dimensions is possible with 3 dimensions slightly skewed
- Tree layout doesn’t overlap, could be better
- Everything is drawn incrementally
  - bad in X, better in GL
- Extensible GUI to allow for different hypercube views

Upcoming Features

- The ability to solve problems
- Edge picking
- Better tree layout
- A better user interface

An Interactive Demo

- How do things look now
- What’s next
- Let’s play “find the edge”…

Questions

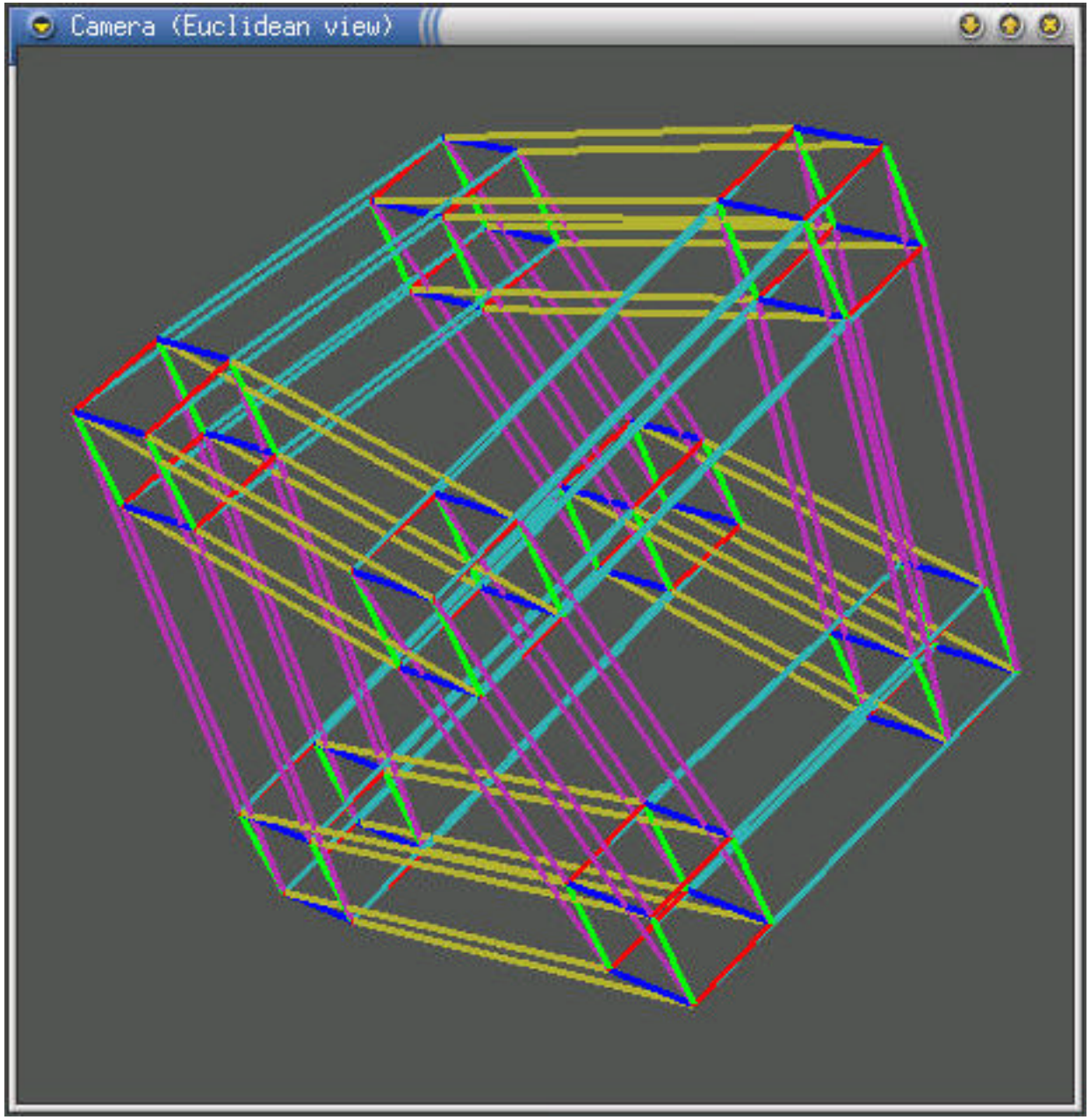
Comments?

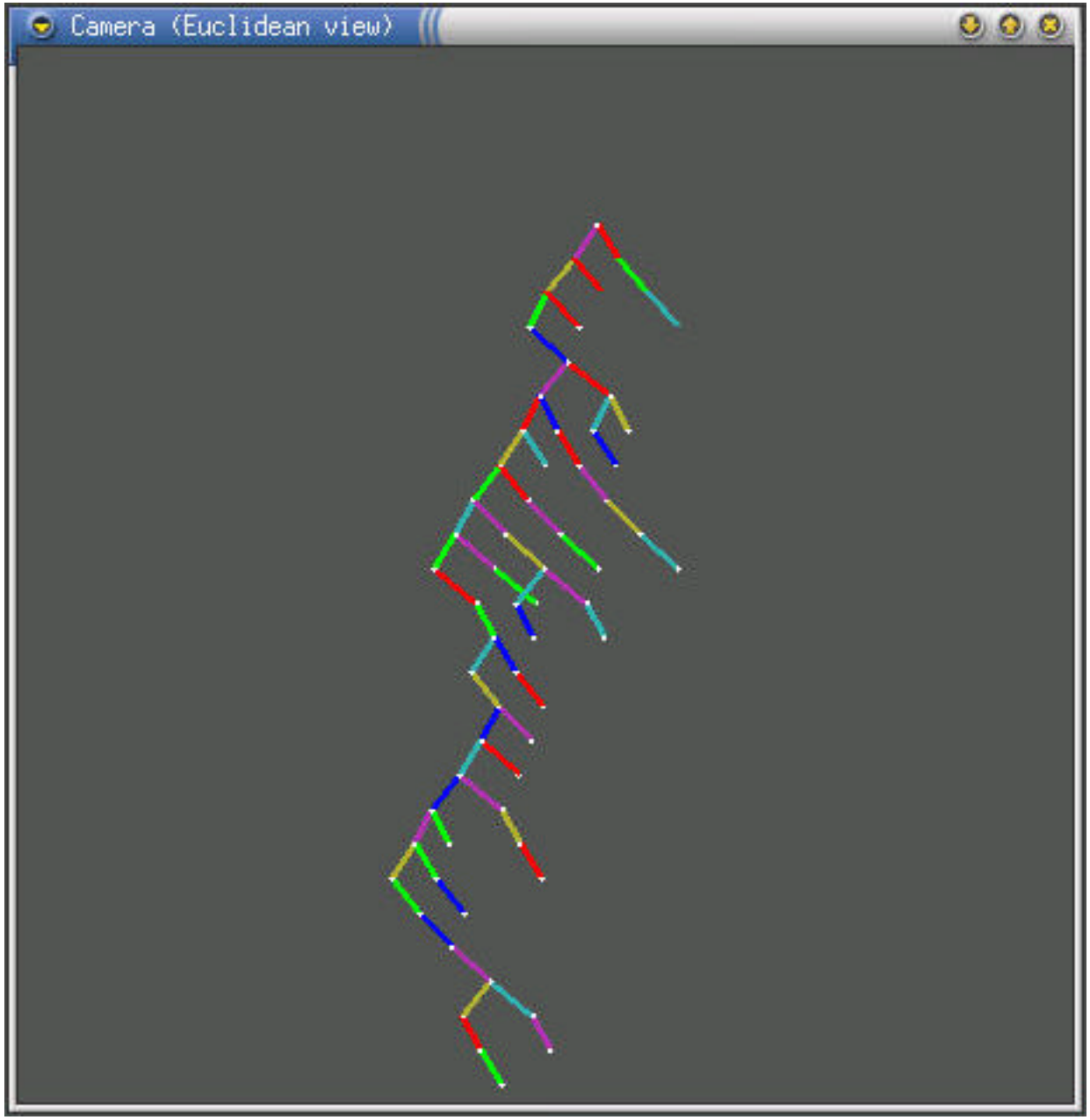
Suggestions?

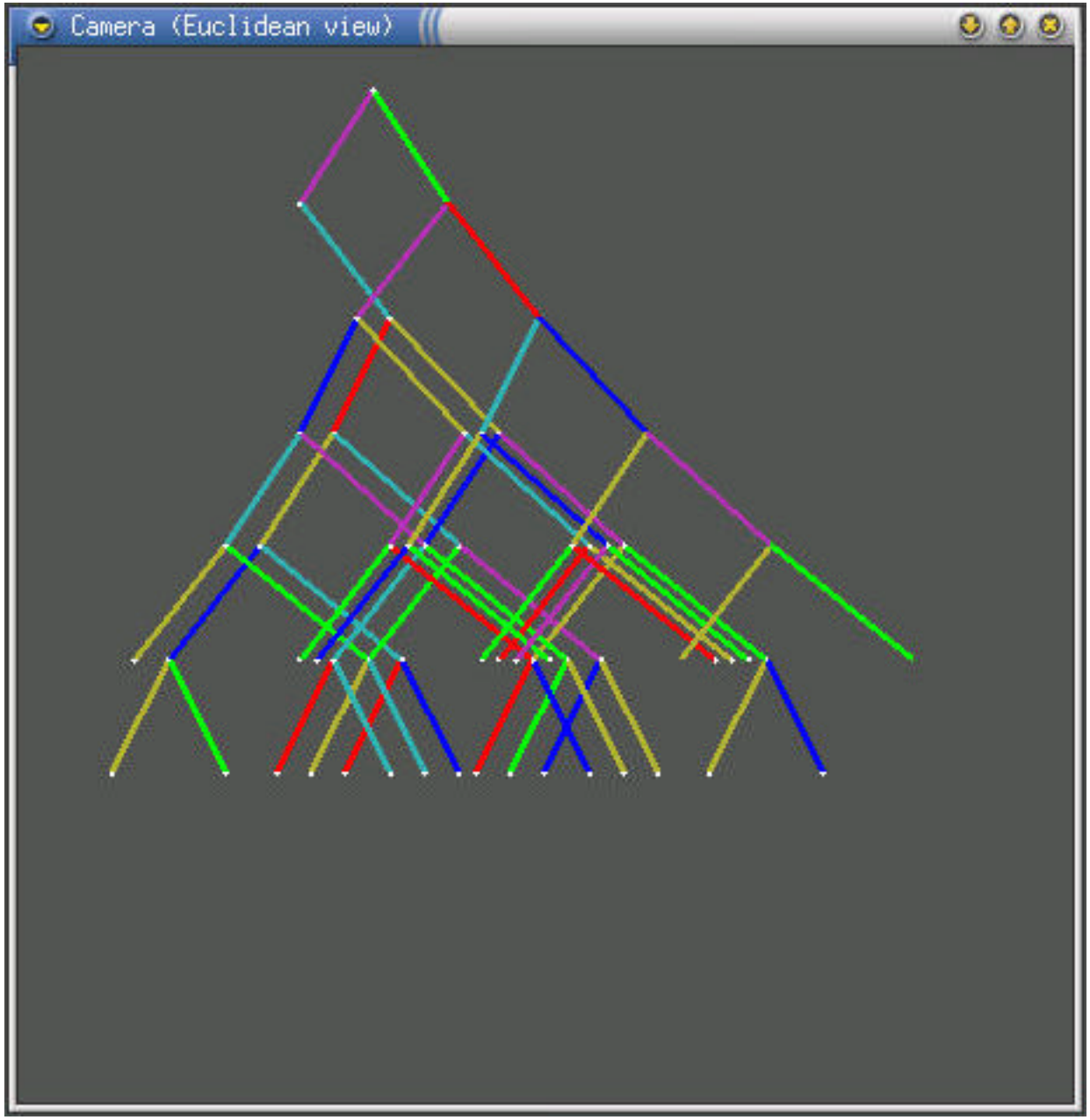
What would you do better?

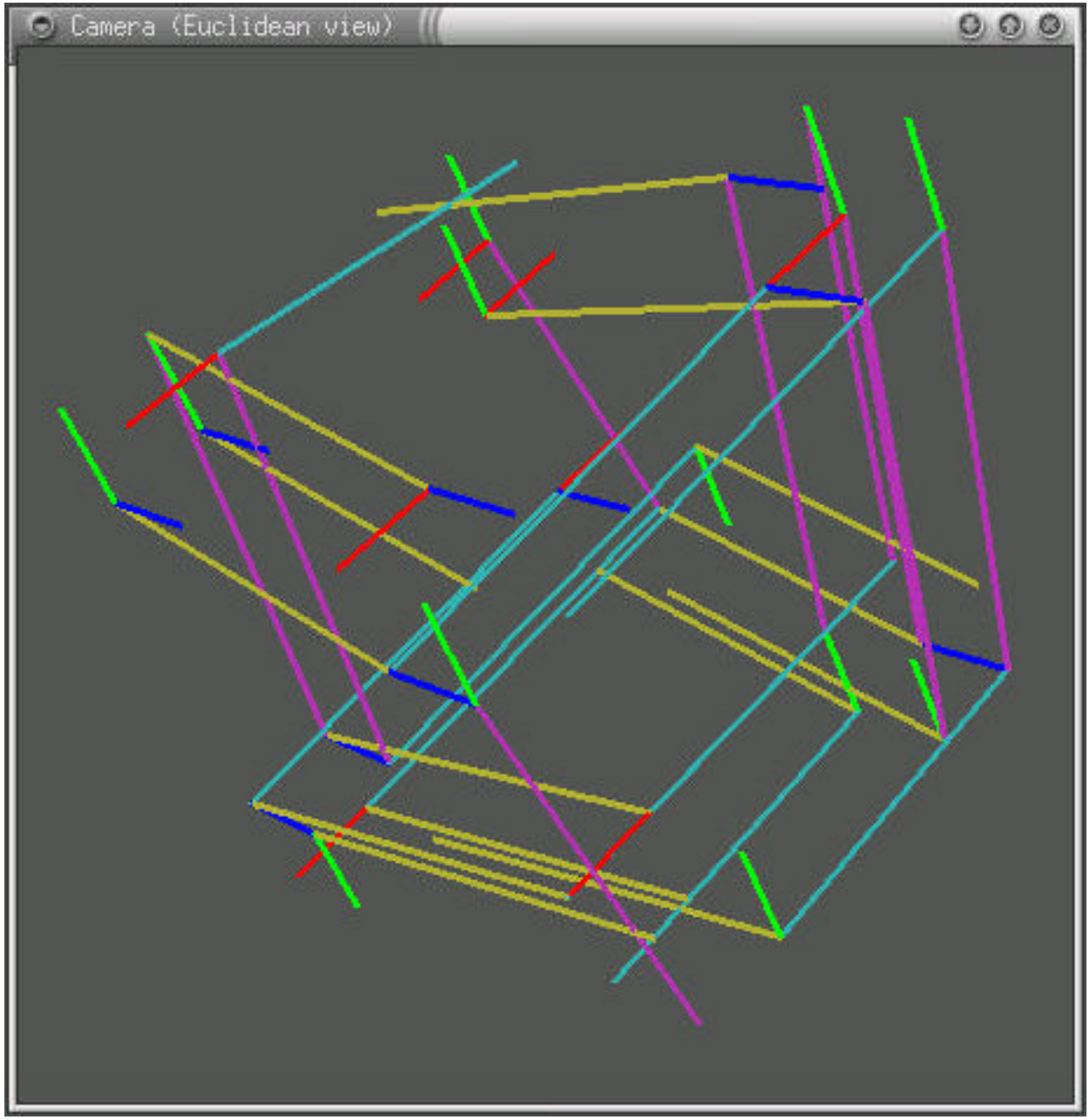
Something went wrong with the demo…

- Here are some pictures


Camera (Euclidean view)


Camera (Euclidean view)


Camera (Euclidean view)


Camera (Euclidean view)