

CS 442 Final Project

Proposal due May 5

Project due May 26

The final project is an open-ended research project that should be completed by groups of two to three students. The project consists of defining a new DSL or adapting an existing DSL and implementing the DSL using Delite. The goal will be to demonstrate that using the DSL improves both productivity and performance. A few possible project ideas are listed below. We also highly encourage you to propose your own project.

Data visualization:

Protovis is a DSL for data visualization developed at Stanford that composes custom views of data with simple marks such as bars and dots. Unlike low-level graphics libraries that quickly become tedious for visualization, Protovis defines marks through dynamic properties that encode data, allowing inheritance, scales, and layouts to simplify construction. Protovis is currently embedded in JavaScript. For this project you will embed (a subset of) Protovis in Scala using the Delite Compiler Framework. You will likely need to extend the framework's current code generation capabilities to emit JavaScript.

<http://vis.stanford.edu/protovis/>

Data Cleaning:

Data Wrangler is a DSL for data cleaning and transformation developed at Stanford. It is designed to transform messy, real-world data into the data tables that analysis tools expect. For this project you will embed Data Wrangler in Scala using the Delite Compiler Framework. A possible point of focus is to implement a back-end optimized for performing transformations on very large datasets.

<http://vis.stanford.edu/wrangler/>

Image Processing:

OptiML currently has very basic support for implementing image processing applications, along with one example application, Binarized Gradient Grid Detector. A proper image processing DSL, however, should have a much richer interface and capabilities. For this project you will design such a DSL. You may use the current OptiML functionality as a starting point, or start from scratch. Since image processing is often useful in machine learning, e.g., robotics applications, an interesting point of focus would be to design this new DSL to be interoperable with OptiML. A single application could then use both DSLs at different points in the program.