

Progress Report: Mathematical Artwork using cs249 Design Strategies
Nicholas A. Dovidio

Recall that in this project I was going to refactor prior Java code used to make photomosaics and stippled images. Creating these photomosaics requires solving a large linear system. To do this I employ a powerful linear solver, Ilog's Cplex. Much of the original code and the Cplex library did not fit with the cs249 methodology. To rectify these shortcomings I rewrote much of the original code and am writing a wrapper around the Cplex library. Further I wrote a basic framework in Java for value types and logging as used in cs249.

Before I could even begin to tackle the problem of creating mathematical artwork, I needed to create a basic framework in the style of cs249. The framework currently consists of logging, exception handling, and value types. Implementing these features revealed multiple shortcomings in Java. For example, generics only allow objects to be taken in as values. This means that I could not have a value type of the form: `Image_Intensity<double>`. Instead I had to use the value types which inherited from `Number`. Further analysis of this shortcoming and other problems will be discussed in the final paper.

Once the framework was completed I began to write code to read in and analyze the images for the photomosaics. This code is completed and fully tested. The next step was to write a wrapper around the Cplex library. To determine the API for the wrapper I used outside-in-development. More specifically I knew that our interface had to solve the photomosaic problem. I also had a very simple linear problem I needed to solve. These two problems drove the interface design decisions. Nevertheless, designing the interface was significantly more challenging than originally thought. The Cplex library has over 600 pages of documentation. In my version I significantly limit and simplify the calling mechanism, while still allowing robust linear systems to be created and solved. The wrapper is nearly completed and is in the process of being debugged.

The next goal is to finish writing the wrapper class and then to implement a simple photomosaic program. My schedule has slipped because of two significant and unexpected difficulties. The first was writing the framework was more time consuming than I anticipated. This occurred because I had to learn how to work with Java generics and enumerations. Additionally I chose to work with the main Java interfaces when I could. This led to some major difficulties, for example `compareTo` always throws an exception. The other difficulty involved creating a makefile which links to the Cplex routines. Stanford has purchased rights to use the Cplex library but creating and working with the associated makefile is very difficult. The syntax needed to reference and find the licensing file was nontrivial.

I have redefined my project not to include stippled images. I realized that my current work has already touched on many of the topics covered in cs249. More importantly I do not believe that adding stippled images would use an additional feature from the course. Instead I want to focus on creating and refining the wrapper class for the Cplex library.

My refined schedule has me finishing the wrapper class by the end of this week. Then next week I will begin to work on creating the photomosaics. After that I will spend the rest of the quarter analyzing the resulting photomosaics and the associated code. Much of my paper will focus on the difficulties of implementing the cs249 framework inside of Java.