Photomosaics Project Proposal

Nicholas Dovidio
cs249B

Overview and Background

Mathematical artwork employs mathematics to create pieces of artwork. In one such method, photomosaics, images or other objects are used to approximate a target image. In effect the target image is simulated by other objects.

Professor Robert Bosch of Oberlin College has created a variety of photomosaics using different mathematical formulas. These include the traditional photomosaics, where smaller photos are combined to approximate a target image. He has also written papers on first using a stippled image to approximate a target image. Then the dots from the stippled image are connected by the shortest possible path. He calls this traveling salesman artwork. He has also used complete sets of dominoes to approximate a target image.

All three of these problems are formulated as minimization problems with a very large number of decision variables. Sometimes these pieces of artwork require over one million decision variables. Since the problems require such a large number of variables, outside libraries are needed to solve the problem. One such library which Stanford has is ILOG CPLEX.

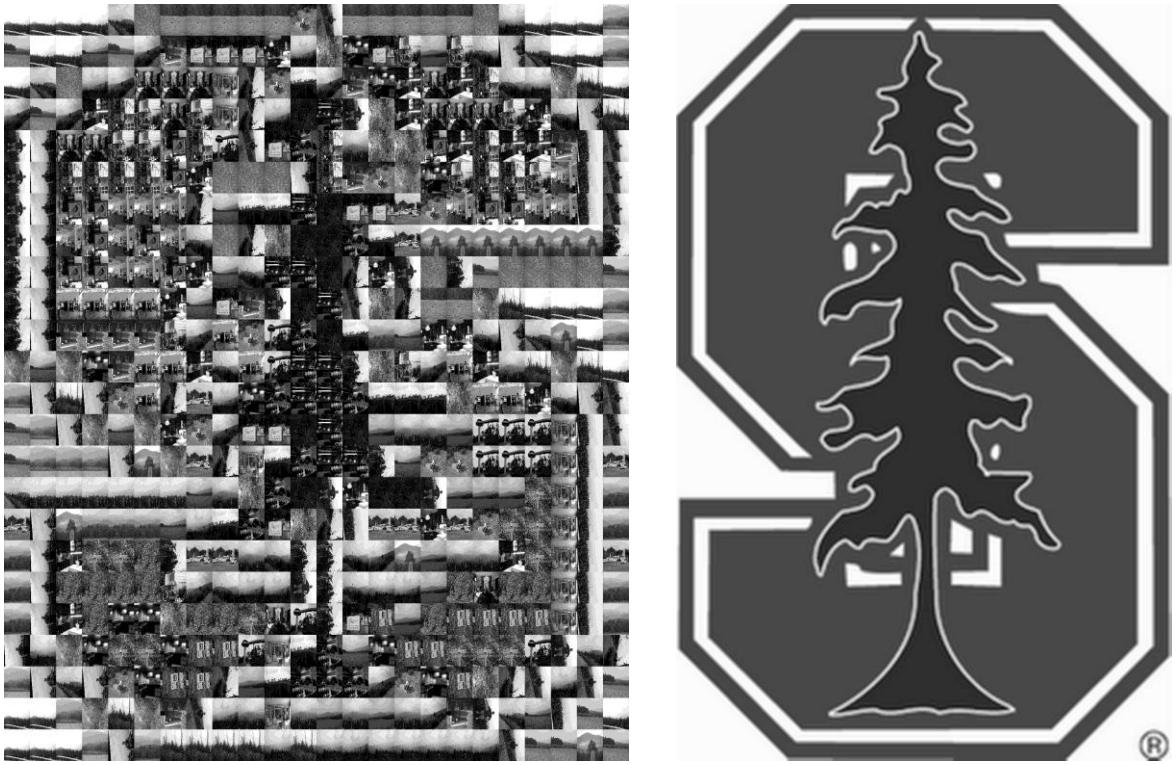
Project Proposal

As an undergraduate I wrote software that mimics Professor Bosch's work. This software is difficult to use and does not resemble the attribute-only interface proposed in cs249. Further it is written in multiple languages and relies on many programs to solve the associated minimization problems. At the heart of these problems is a simulation, using some object to approximate another object. This property makes these problems particularly well suited for the attribute-only approach. I propose rewriting the software in the style of cs249 and then analyzing how well the target image is approximated and how well the cs249 techniques work. This will involve a variety of steps and incorporate many ideas explored in class.

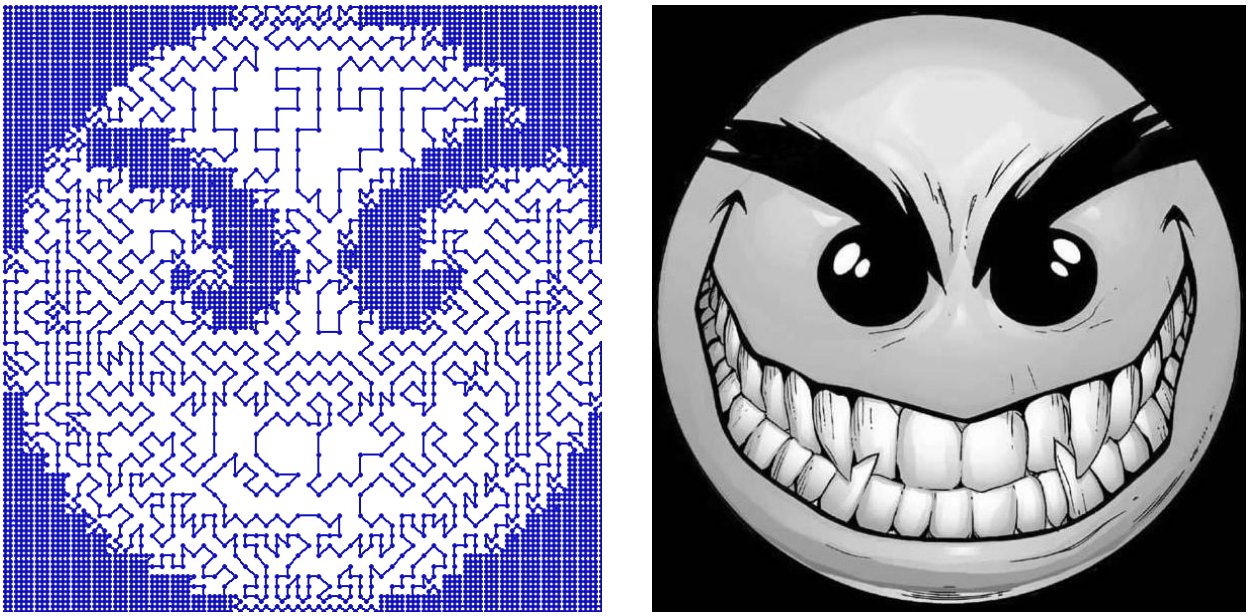
My first step will be to examine CPLEX and write a wrapper around the library in the style of cs249. This will explore two key areas of cs249. The first is changing this into an attribute-based interface. Currently the interface supports actions such as "solve" and "add." These are not very specific and can be confusing. For example, solve does not convey if the solution is a maximization or a minimization of the objective function. Additionally CPLEX has many different exceptions that are confusing for a client user. As described in the course notes the wrapper class will forward useful exceptions to the user as an optimization and internally handle exceptions that the user has no ability to handle. The exceptions will be logged in the style suggested by the course notes.

The next step will be to write software employing the CPLEX wrapper to create the photomosaics. Two types of photomosaics will be created and the corresponding code will follow the attribute-only interface design. The first is the traditional type where small images are used to approximate a larger target image. In this interface attributes will include multiple attributes such as orientation and number of uses per image. The second type of photomosaic

will be based upon using dots to approximate a target image. These dots are then connected via a travelling salesman problem. An example attribute in this problem is how the dots are distributed. Examples of the two types of problems are shown below:



Traditional Photomosaic



TSP Dot Photomosaic

Image From: http://www.logichigh.com/logichigh/2006/06/color_inversion.html

The last step will be converting the client code into the form presented in class. This will involve code that converts images to grayscale, analyzes the image intensity, and writes the solution to a file. The code's current implementation employs "verby" methods such as print. These will be replaced by the double dispatch / functor approach discussed in class. Some of the analysis code used in my original work was written by my friend Gavin Taylor, who is now at Duke. All of his code will need to be rewritten in the attribute-based manner.

The project paper will consist of two parts. The first will be analysis and exploration of the photomosaics created. For example, Professor Bosch suggests a variety of stippling techniques; these will be implemented and discussed. The next part will be analysis of using cs249 style of coding in the manner described previously. Finally, the project will be implemented in Java because I am only familiar with the Java implementation of CPLEX. There will be analysis of how the cs249 methods carryover to other object-oriented languages.

Milestone Goal

The milestone will contain an initial wrapper class for CPLEX. This class will be able to solve simple optimization problems and will be written in the cs249 style. It will also handle exceptions and logging in the appropriate manner. Further a simple traditional photomosaic interface will be implemented.

Expected Results

The final project will contain the completed wrapper for CPLEX and the interfaces to create the two previously discussed types photomosaics. Further there will be analysis of employing the cs249 methods to these problems and implementing these methods in Java.

Tentative Reference List

Below is a tentative reference list after a brief literature search.

Kaplan, Craig S. Robert Bosch. *TSP Art*.

Herman, Adrianne, Robert Bosch. *Continuous Line Drawings via the Traveling Salesman Problem.* Available at: www.optimization-on**line**.org/DB_FILE/2003/09/721.pdf.

ILOG CPLEX and associated User Manuals. Specifically ILOG Concert Technology for Java Users and CPLEX 7.5 Java Guide. http://www.ilog.com/.