Davidson Housing Coalition: A Picture Perfect Resolution

Executive Summary

Daniel Martin, Brian Sachtjen, and Carter Eggleston

April 2010

Executive Summary

At first, making a photomosaic may seem like a relatively straightforward task that is more suited to someone who works with photography or graphic design. After all, all that one needs to do is just divide an image into a certain number of same sized tiles and then replace each of those tiles with a smaller, similarly looking photograph. However, constructing a photomosaic in this fashion would take many hours of careful examination and work, and still would not guarantee a good solution. A more practical approach would be to create photomosaics mathematically. Various algorithms can be used to mathematically match pictures to tiles, and then computer programs can be run to assemble all of the chosen pictures in the correct order to create the photomosaic. With this project, we explored the mathematical possibilities of this process in order to acquire a better understanding of the mathematics behind photomosaic creation to efficiently produce ones of high quality.

The first step in making a photomosaic is to obtain images. Some of the images will be chosen as the overall image that the many smaller images are able to replicate when viewed from a distance. We call these the target images. The rest of the images will make up the image library. The library is an integral part for any photomosaic, as a large and diverse pool of images is needed to accurately model almost any target image. For our library, we acquired images from the Davidson Housing Coalition and from Bill Giduz, the Director of Media Relations at Davidson College. Thanks to them, we were able to have an image library with close to 900 pictures.

Then, we considered three different methods to create the photomosaics. All three begin in the same manner. First, the program will import the target image and divide it into a number of same-sized tiles. The amount of tiles can be specified at the beginning of the program. Next, the program will import the library of images, resizing each image to be the same size as the tiles. At this point, our methods diverge, but all three still rely heavily on the concept of pixel value. Each image is made up of a number of pixels. For example, many of our target images are 1800 pixels tall and 2400 pixels wide. Each pixel displays color based on its red value, green value, and blue value, or together, its RGB value. This value is important to our methods.

The first method we considered is the Ranked Averages Method. This method first finds the average RGB value of all of the images in the library and all of the tiles in the target image. Then,

it ranks each image in the library for each tile from best-fit to worst-fit. Finally, it selects at random one of the top matches and places that selection into the photomosaic for each tile.

The second method is a binary integer linear program (BILP). Like the Ranked Averages Method, it too first finds the average RGB value for each image in the library and each tile in the target image. Next, the BILP produces a matrix of variables equal to the number of images multiplied by the number of tiles – in our case, over 3,000,000. The program selects images for each tile to minimize the total amount of error across the photomosaic, thus maintaining a global outlook. For this method, one is able to specify how many times an image is repeated.

Our third method is the Pixel Comparison Method. This method ranks each image in the library for each tile pixel by pixel rather than averaging across the entire space. It then selects an image for each tile that has the closest RGB value for each pixel.

After seeing the photomosaics that these methods produced, we arrived at the following conclusions. The Ranked Averages method produced a very good mosaic. However, because we randomly pick one of the top matches, we have little control over the number of repeats. Conversely, the BILP lets us define the number of times each picture can be repeated and produces a mosaic of high quality. Unfortunately, the same picture would often repeat in adjacent tiles. The biggest drawback for the BILP method however is the time required for the algorithm to run. Finally, the Pixel Comparison method creates a very poor photomosaic that leaves out a lot of color. This method was a failure and should not be used without significant improvements. For these reasons, we recommend using the Ranked Averages Method.