

Enhancement of the Enlargement of Images

Tara Naughton

June 14, 2010

Abstract

Methods typically used to enlarge images - pixel replication and interpolation - either produce images too jagged or too blurred. The intent of this project is to develop a method for enlarging images that retains the sharpness of edges while still keeping an image that looks smooth and high quality. The focus is on building off of the interpolation method by detecting edge pixels and deriving an equation to appropriately adjust their intensity in the enlarged images.

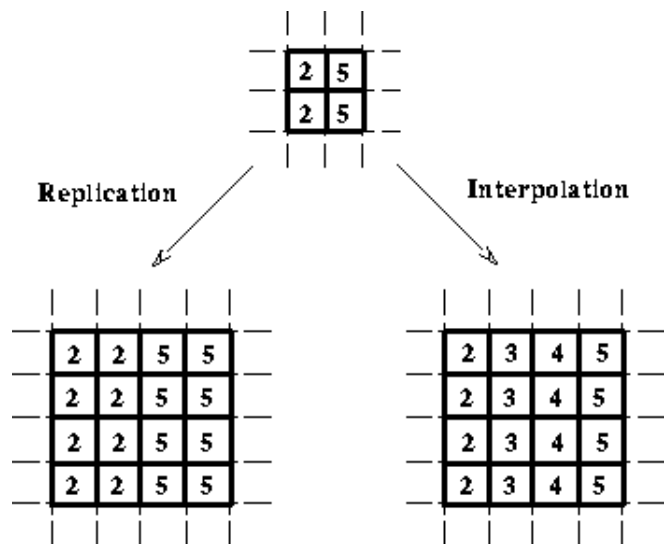


Figure 1: Pixel replication vs. interpolation

1 Introduction

In image processing, two different methods of image enlargement are most commonly used: pixel replication - which simply repeats each pixel value for the amount of the scale factor - and interpolation - which constructs new data points between two pixels that work as a gradient connecting one pixel to another in the enlarged image. Unfortunately, both methods tend to produce less than desirable results; images resized with the pixel replica-

tion method often look very jagged and overly pixelated, while on the other hand, images resized with the interpolation method come out too blurry and with undefined edges. This project aims to find an algorithm that produces high quality enlarged images that can both preserve sharpness and avoid producing an overly pixelated image.

2 Background

2.1 Similar Research

This project is not the first to call attention to the inadequacies of the common types of image enlargement; several research projects have been done in the past to develop algorithms to produce enlarged images that are both sharp and continuous. One method explored was enlargement with use of a step edge model. With this algorithm, edge pixels were located, ramp parameters were derived from those pixels, and intensity values were derived from those parameters. Those parameters are then kept and applied to the larger image.

Another method that has been worked on involves a step edge model. The image is divided into atomic areas, and step edge parameters are derived for each pixel. An intensity is found for each atomic area. Additionally, a step-likeness value is found to see if the area is actually near a step-edge. That step-likeness value is then used as weight to combine the intensity values for that atomic area.

While many of the algorithms that have been developed utilize different methods, many of them focus on treating edges separately from the rest of the image and preserving their sharpness. Edge detection methods are often used after interpolating an image.

2.2 .pgm Image Format

The project works with .pgm images. .pgm is the Portable Graymap Format, a form of

grayscale image. While not commonly used, it is an image format very convenient for reading in image data for an algorithm. The

```
P2
600 600
255
14
14 10 10 9 9 10 10 11 11 10 10 10 10 11 11 7 7 9 9 14 14 21 21 26 26 24 24
14 10 10 9 9 10 10 11 11 10 10 10 10 11 11 7 7 9 9 14 14 21 21 26 26 24 24
12 9 9 7 7 8 8 9 9 8 8 8 8 10 10 8 8 9 9 14 14 21 21 23 23 19 19 14 14 12
12 9 9 7 7 8 8 9 9 8 8 8 8 10 10 8 8 9 9 14 14 21 21 23 23 19 19 14 14 12
10 7 7 5 5 7 7 8 8 7 7 7 7 8 8 10 10 10 10 15 15 22 22 20 20 12 12 10 10 1
10 7 7 5 5 7 7 8 8 7 7 7 7 8 8 10 10 10 10 15 15 22 22 20 20 12 12 10 10 1
9 6 6 5 5 6 6 7 7 7 7 7 7 8 8 11 11 11 11 16 16 22 22 18 18 8 8 9 9 19 19
9 6 6 5 5 6 6 7 7 7 7 7 7 8 8 11 11 11 11 16 16 22 22 18 18 8 8 9 9 19 19
10 6 6 5 5 7 7 8 8 8 8 8 8 10 10 11 11 13 13 18 18 21 21 15 15 8 8 12 12 2
10 6 6 5 5 7 7 8 8 8 8 8 8 10 10 11 11 13 13 18 18 21 21 15 15 8 8 12 12 2
10 7 7 6 6 8 8 9 9 9 9 9 9 11 11 11 11 16 16 20 20 19 19 12 12 9 9 16 16 2
```

Figure 2: Data contained in a PGM image file

first value indicates what kind of image format the image is. "P2" indicates a .pgm file. The next two values indicate the width and length of the image. The next value indicates the maximum value of intensity present in the image file; a value of 255, for example, means there are 255 shades of grey between white and black. The rest of the numbers in the image data represent the intensity value of each individual pixel.

3 Development

The code for the project is written in Python. The algorithm takes a black-and-white .pgm image as input, call on a edge detection method and interpolation method that will be written separately and then called upon in a separate part of the code, and produce a .pgm image that is a resized version of the original. The interpolation method resizes

non-edges adequately, so it is only necessary to single out the edges and resize them independently in order to preserve the sharpness of the image. To test the program, images produced by the algorithm will be compared to the same resized image of the same scale produced by the two traditional methods (pixel replication and interpolation) and overall image quality will be evaluated.

4 Results

More traditional versions of the pixel replication and interpolation methods were successfully completed from scratch. They both asked for image input and threw errors. Additionally, a sobel edge detection method was completed, which included a method that derives the edge values of certain pixels using the values of neighboring pixels and a horizontal and vertical gradient mask.

Horizontal gradient G_x mask:

-1	0	1
-2	0	2
-1	0	1

Vertical gradient G_y mask:

1	2	1
0	0	0
-1	-2	-1

$$G = |G_x| + |G_y|$$

Figure 3: G-value masks and equation

An equation to derive intensity values for resized pixels based on this edge value was underway when the quarter came to an end. Unfortunately, problems concerning input in the code kept the program from being able to fully incorporate the G values derived for

each pixel. The results appeared not much different from the original interpolated image.

5 Discussion

I was able to successfully develop my own pixel replication and interpolation method from scratch; however, I did not quite attain the results I had hoped for. I had wanted to make my interpolation techniques better incorporate the intensity values derived from edge pixels, but I ran into several problems trying to modify my code. My results tended to either look discontinuous or not show enough of a difference from the traditional interpolation method.

5.1 Future Research

There were many things I wanted to do with this project that I did not quite get to accomplish. Of course, given more time I would continue to work on my project until it yielded the results I was aiming for, but there are additional steps that could be taken in future research. One aspect I would modify about my project in the future is the input. While the .pgm image format is a convenient one to use to get the basic algorithm running, it is not the most commonly used image format. I would like to expand my project to take additional image inputs, such as .jpgs and .gifs. Additionally, the project could be expanded to include color rather than just working in grayscale.



Figure 4: Original image

References

- [1] Jia-Guu Leu, "Image enlargement based on a step edge model", *Pattern Recognition*, pp. 2055-2073, 2005.
- [2] Jia-Guu Leu, "Image enlargement based on a ramp edge model", *Pattern Recognition*, pp. 1927-1938, 2001.
- [3] Fisher, R., et al., "Geometric Scaling", 2003.
<http://homepages.inf.ed.ac.uk/rbf/HIPR2>
(October 30, 2009)



Figure 5: Image resized by pixel replication



Figure 6: Image resized by interpolation