Teaching Mathematics Through Image Manipulation

Patrick Honner
Brooklyn Technical High School
29 Ft. Greene Place
Brooklyn, NY, 11217, USA
E-mail: patrick.honner@gmail.com
http://www.MrHonner.com

Abstract

Inspired by the work of artists and presenters at the Bridges conferences, I have begun to experiment with custom Python code that transforms digital images in mathematical ways. My initial foray into this intersection of mathematics, computer science, and art has employed direct applications of trigonometry, algebra, calculus, and many other mathematical concepts from standard secondary mathematics curricula. My goal is to develop an instructional program that will allow students to design and apply mathematical tools to create and explore this simple artistic space.

Introduction

Participation in the Bridges conference has inspired me to re-imagine digital images in my own way. In particular, Hans Kuiper's *Optical Minimal Art* [1] motivated me to develop simple Python scripts that execute basic mathematical transformations on my photographs. Experimentation and exploration of these basic manipulations led to more intricate and interesting transformations, and in trying to produce these more complicated transformations, I was forced to solve a number of authentic mathematical problems. As a teacher, this fun and meaningful process of learning, applying, and creating immediately struck me as something students would enjoy and benefit from. Thus, my goal is to build an instructional program based on my experiences.

What follows are some examples of the basic transformations I explored that will serve as a skeleton for an interdisciplinary instructional unit in mathematics, art, and computer science.

Technology

The following examples were all produced using the programming language Python. Python is free and open-source, and thus is a good option for classroom use. Web-based platforms like Sage [2] make both classroom integration and home use feasible.

The Python Imaging Library (PIL) provides an *Image* module that allows images to be loaded and treated as objects [3]. A variety of traditional image-processing capabilities are offered through the module's built-in functions, but much can be accomplished and explored using only the most basic operations on pixels.

Translations

Translations are a standard topic in secondary mathematics, and beginning with these mathematically simple transformations students can become comfortable with understanding the image as a mathematical object while gaining facility with the necessary computer programming skills.

By understanding each row of the image as a list of numbers that can be shifted and wrapped around (using modular arithmetic), we can transform images in simple ways. Below we see the results of a horizontal translation by a fixed amount (Figure 2), a linearly varying amount (Figure 3), and an oscillating amount (Figure 4).

Figure 1: *Original image*

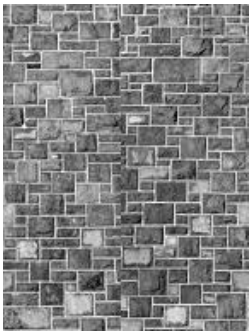

Figure 2: *Fixed horizontal translation*

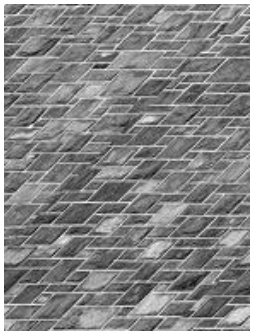

Figure 3: *Offset translation*

Figure 4: *Oscillating translation*

In each case, the original image is loaded in Python as an image object named *im.* The new image, *im1*, is created by copying the specified pixel from the original image. Here we see the relevant snippet of code for the fixed horizontal translation (Figure 1).

```
for x in range(width):
    for y in range(height):
        im1.putpixel((x, y),(im.getpixel(((x + 250)%width, y))))
```

The function *putpixel* creates the pixel in *im1* by copying the the color values of the pixel retrieved from *im* by the *getpixel* function. Here, *getpixel* retrieves the pixel from *im* that is 250 pixels to the right of the current location in *im1*, thereby effecting a translation 250 pixels to the right. The use of modular arithmetic enables the wrapping of the image.

To effect the offset translation (Figure 2), the following code is used inside the nested *for* loops.

```
im1.putpixel((x, y),(im.getpixel(((x + y)%width, y))))
```

This creates a translation whose distance increases linearly as the rows are traversed. The introduction of other functions, like the sine function (Figure 4), can produce more interesting transformations of the image.

```
im1.putpixel((x, y), (im.getpixel(((x + 20*math.sin(y/6))%width, y))))
```

This simple framework can be used to visually explore different kinds of functions and sequences, providing motivation and context for the underlying theory.

Dilations and Shears

Another simple transformation familiar to secondary mathematics students is the dilation, and below we see the effects of two simple kinds of dilation on the original image: in Figure 5, a dilation from the center of the image; and in Figure 6, a horizontal shear of the image.

Figure 5: *Basic dilation*

Figure 6: *Basic shearing*

Producing these simple effects allows students to experience the difference between various linear transformations in a meaningful context. It also requires that students understand mappings of the plane as functions whose domain and range correspond to the original and desired images; in fact, image processing offers a rich environment to explore the concept of function, as discussed by S. Tanimoto [4].

Furthermore, these activities give students a sense of how various digital technologies work. Zooming in on an image, for example, is an applied dilation: it takes a small portion of an image and enlarges it by a one-to-many mapping of the pixel array. Similarly, image resizing is another example of an applied dilation; thus, these experiences help connect high school mathematics to real student experiences, which strengthen and solidify learning.

Rotations

Secondary students are also familiar with rotations, and implementing these kinds of transformations of images present a number of interesting mathematical challenges. Producing a basic rotation of a circular region of a given image, as seen in Figure 7, establishes meaningful context to develop and apply various trigonometric concepts as well as the system of polar coordinates.

In order to produce a smooth, continuous appearance in the transformed image, the ideas of weighted averages, limits, and asymptotic functions can be introduced. By appropriately varying a rotation's "strength", a rotation can be smoothed out as in Figure 8. Again, in addition to applying numerous mathematical concepts in an authentic problem-solving context, students also develop a sense of the basic theory of digital image processing.

Figure 7: *Basic rotation*

Figure 8: *Smooth rotation*

Combined Transformations

Once basic transformations have been introduced, students can then begin to combine transformations as they desire to manipulate images in more complex ways. In Figure 9, the original image is horizontally "compressed" along a vertical sine wave. In each row, pixels are moved horizontally toward an anchor point, and the distance each pixel moves is a function of its original distance to the anchor. From row to row, the anchor point moves along a sine wave, which produces an interesting visual effect.

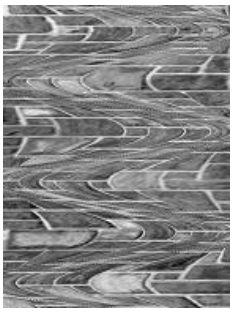

Figure 9: *Compression about a sine wave*

Figure 10: "*Building Sines"*

Ultimately, students can apply their knowledge of basic image transformations to their own photography, thereby creating art that is both personal and mathematical. In Figure 10, a favorite image of mine has been compressed along a vertical sine wave, producing an interesting visual result.

Conclusions

Digital image processing offers an opportunity to create learning experiences for secondary students that combine mathematics, computer science, and art. A project-based approach in this environment can be successfully employed in a variety of science contexts [5], but it is particularly well-suited for use in standard, as well as advanced, high school mathematics courses. These activities address a variety of fundamental mathematical ideas and practices, such as those articulated in the Common Core State Standards [6] relating to congruence, similarity, transformations, and functions. The activities can be easily extended to more advanced ideas, and they can help improve secondary mathematics education by presenting mathematics to students in meaningful contexts, leveraging creativity to promote student engagement, and integrating STEM skills into authentic mathematical problem solving.

References

[1] H. Kuiper, *Optical Minimal Art*, Proceedings of Bridges 2012: Mathematics, Music, Art, Architecture, Culture, pp 405-408, 2012.

[2] http://www.sagemath.org (as of 4/8/13)

[3] http://www.pythonware.com/library/pil/handbook/image.htm (as of 4/8/13)

[4] S. Tanimoto, *Image Analysis and Teaching the Concept of Function*, presented at ED-MEDIA 2001, available at http://www.cs.washington.edu/education/courses/cse590d/01au/Tampere2.pdf (as of 4/8/13)

[5] J. Raphael and R. Greenberg, *Image Processing: A State-of-the-Art Way to Learn Science*, Educational Leadership 53-2, pp 34-37, 1995.

[6] National Governors Association Center for Best Practices, Council of Chief State School Offices, *Common Core State Standards Mathematics*, National Governors Association Center for Best Practices, Council of Chief State School Officers, Washington D.C., 2010.