

Image Processing with *Mathematica* 8

Bruce Torrence
Department of Mathematics
Randolph-Macon College
Ashland, VA 23005
E-mail: btorrenc@rmc.edu

Abstract

The *Mathematica* software system has been with us since 1988, but only recently has it become viable as a mainstream tool for image processing. In each of versions 7 and 8 a large number of new image processing commands were introduced to the system. This has created an opportunity for the mathematics community to do some things within *Mathematica* that were simply inconceivable in the recent past. At the same time, *Mathematica*'s traditional strengths in symbolic manipulation add a unique dimension to the business of working with electronic imagery, both static and dynamic. We'll look at a number of examples, and attempt to identify some of the most promising avenues for exploration in this new domain, as well as its limitations.

Background

Since its creation, *Mathematica* has been an outstanding environment for programmatically creating vector graphics of mathematical objects, such as parametric plots of surfaces. But *Mathematica*'s commercial success derived primarily from its fundamental strengths: replacement rules, symbolic computation, and a forward-looking functional programming language. As these areas were explored and developed in subsequent releases, the already proficient graphics were largely left alone. But a number of incremental developments set the stage for a new paradigm. In version 5, packed arrays and sparse arrays were introduced, laying the foundation that would make it possible to work efficiently with large image files. In version 6, mouse-driven image manipulation tools were rolled out, so that graphics could be edited either programmatically or by hand. Version 6 also saw the advent of dynamic content, whereby expressions could be modified in real time with dynamic, interactive elements. And significantly, beneath the hood, graphics quietly became full-fledged citizens within the functional programming language. Finally, version 7 and (as of late 2010) version 8 each introduced a slew of new commands specifically tailored to the enterprise of manipulating digital imagery. A threshold has been reached, and the new possibilities are both subtle and profound.

An Illustrative Example

An immediate and delightful consequence of this long development is that one now has an environment in which any digital image may easily be wed with any "traditional" *Mathematica* vector graphic. We will illustrate this idea by producing a 3D graphic that illustrates the concept of stereographic projection. We begin with a panoramic photograph, an equirectangular projection of a complete $360^\circ \times 180^\circ$ "viewable sphere."



Figure 1: An equirectangular panorama [1].

It is a simple matter to apply this to image in the “natural” way to the surface of a sphere: One creates the standard parametric function from a rectangle (with longitude and latitude coordinates) to the sphere, and essentially takes the image above as the domain of this function. The “texture mapped” result can be rotated freely with the mouse, and while less tactile than a real ball, it gives virtually the same satisfaction with just a single line of code.

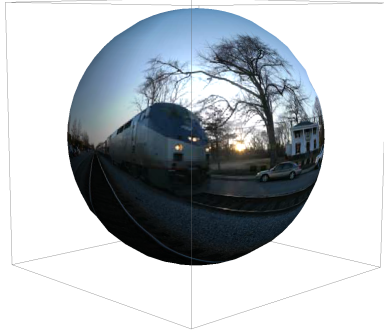


Figure 2: *The panorama on a sphere.*



Figure 3: *Stereographic projection.*

One can also make a stereographic projection of the panorama with ease, using the new `ImageTransformation` command. And just as easily, one can place the sphere on top of the projection, properly oriented, to create a compelling illustration of the stereographic projection:

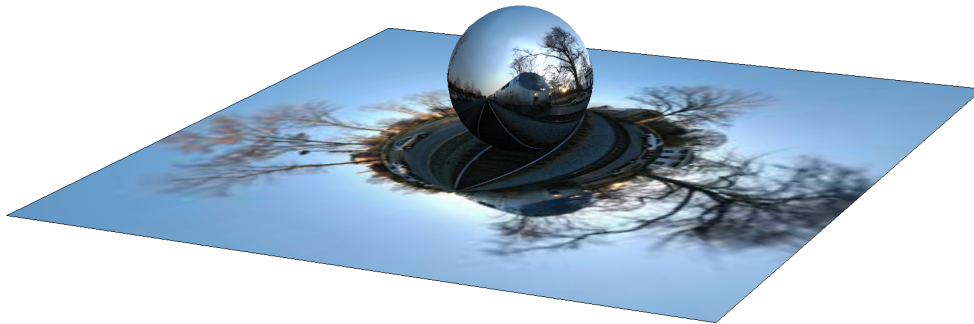


Figure 4: *Stereographic projection illustrated.*

This 3D graphic can be grabbed with the mouse and rotated freely (and this is infinitely more satisfying than viewing the static image). Each of the images above is generated by a single line of code. Another line of code could be used to rotate the image to generate the frames for a movie. Or one could add lines from the north pole to illustrate the one-to-one correspondence between the pixels on the sphere (minus a small open set around the north pole) to those of the plane.

Looking Forward

This single, simple example illustrates the potential for efficiently harnessing the computational power of *Mathematica* with literally any digital image. There are more than a hundred new image manipulation commands that can be used alone or in conjunction with *Mathematica*'s programming language and traditional vector graphics, and the possibilities for mathematical artists are only beginning to be appreciated.

References

- [1] B. Torrence, *Train Planet 2*, <http://www.flickr.com/photos/thebruceemon/5498107266/>, Feb. 2011.