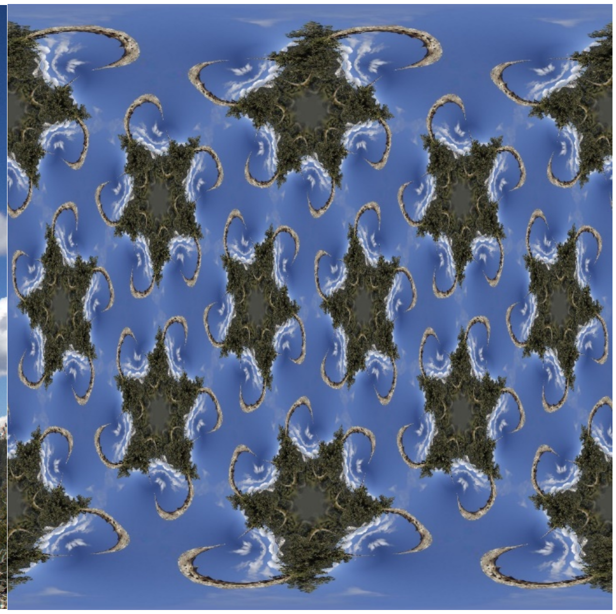
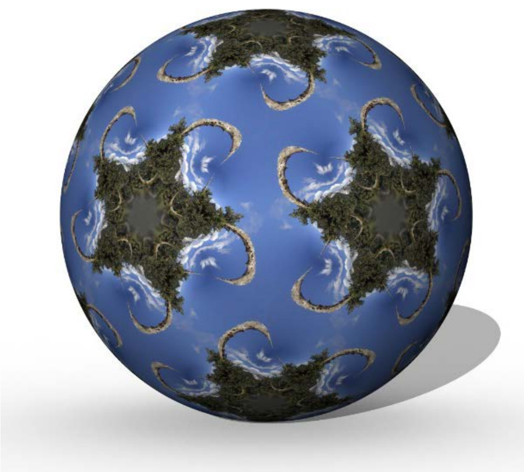
User Guide for Sphere Skins Software

Based on the open-source software SymmetryWorks by Son Ngo and Bridget Went, which was based on code by Cameron Wong and improved by Jay Ladhad. Sphere Skins written by Jay Ladhad and Frank Farris. Algorithms and documentation by Frank A. Farris

Overview

The purpose of the program is to turn source photographs into symmetric colorings of the sphere. However, the output does not look like a coloring of the sphere. Instead, it is a “sphere skin,” which is a bitmap that can be pasted onto a standard spherical mesh. Here is a typical triple: a source photograph, sphere skin, and how the skin looks on a sphere. In this example, the intent was to create a coloring of the sphere with icosahedral symmetry.



Download and Install

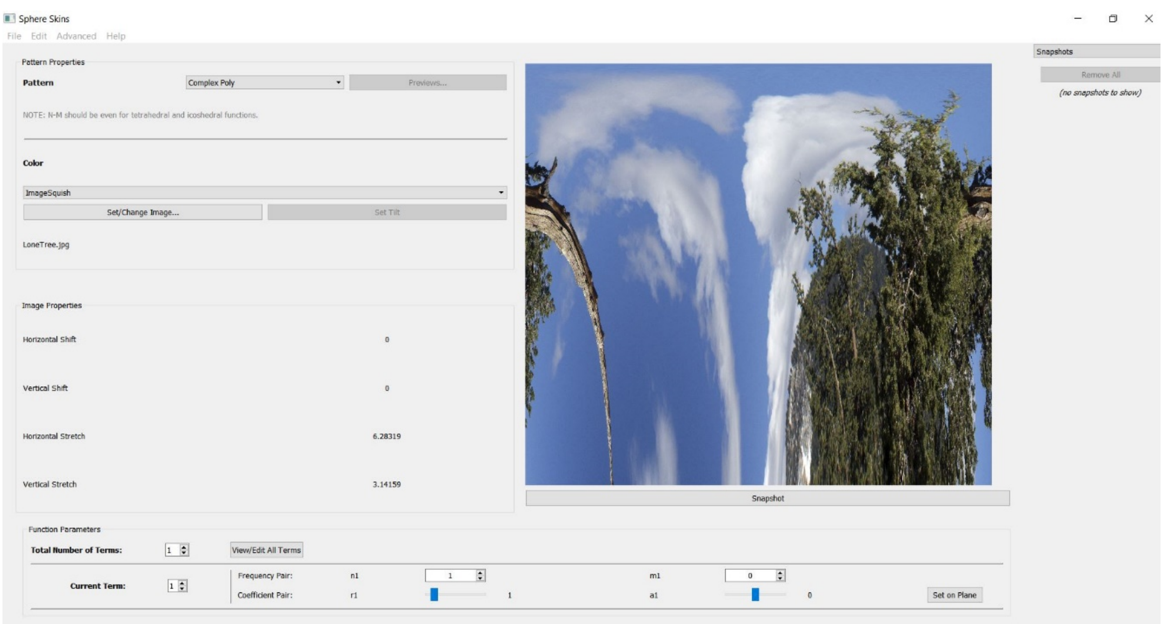
Find our public Github site https://github.com/jrnladhad/symmetryworks-research-bowdoin . There are folders for Wallpaper Generation and Sphere Skins. This document is about Sphere Skins, so open that folder. Windows users can use the latest version, without re-compiling the code, by downloading **WindowsRelease.zip.** Unzip the file to your PC and in the resulting folder you will find an executable file **wallgen.exe**. When you run this software, you may have to override error messages, warning you not to download unsafe software.

If you wish to modify the software, then use the "Clone or Download" button and save the entire project to a folder on your computer. Download and install the full version of QtCreator (you must include QtCharts) and configure it correctly on your machine. Then open the project file, wallgen.pro and Qt should take you through a process to configure the project on your system.

Getting Started

You can learn about spherical symmetry in *Creating Symmetry: The Artful Mathematics of Wallpaper Patterns*, by Frank A. Farris. In this very brief guide, we assume that the reader understands the basics of polyhedral symmetry, stereographic projection, and spherical coordinates.

Source Photograph: The program requires you to set an image to serve as the source for coloring sphere skins. Later, we'll see that spherical photos work best, but if I don't have one of those, I can start with a conventional photograph, like the mountain scene above. The default way to use a photograph is called ImageSquish, which produces the following output.



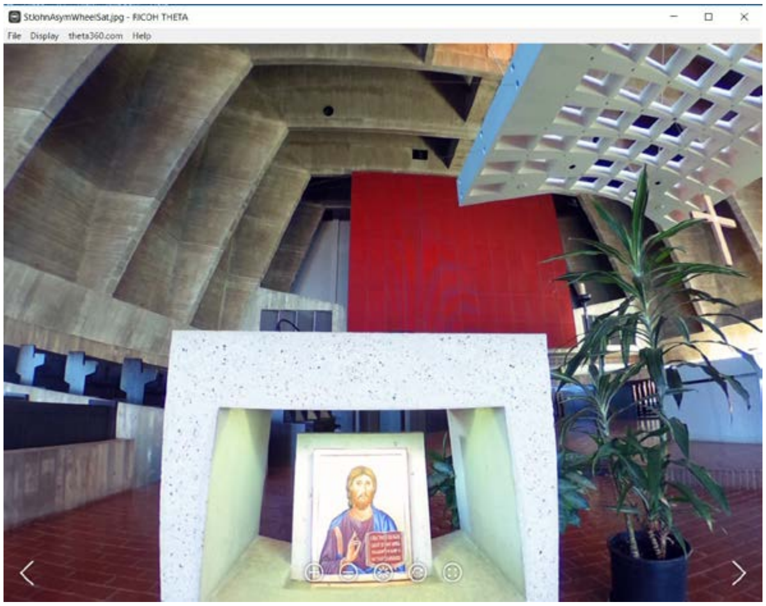
The center of the photograph now appears along the top strip of the display window, which will become the north pole when we paste onto a sphere. The bottom strip takes pixels from a circle into which the image has been “squished.” When this is pasted onto a sphere, the result will be singular.

An alternative, when using a conventional square photograph, is to use the Disk to Sphere option. This pastes the central disk of the photo onto both the top and bottom of the sphere, resulting in a mirror effect. (This can later lead to mirror symmetry in polyhedral images.) Here’s how it looks with my “Lone Tree” image.


Trace your eye along the central horizontal stripe of the right-hand image. You should be able to follow around a circle that is inscribed in the photo.

Spherical source photos: For this application, it is desirable to have photographs that portray the full visual sphere. If I take a picture with a spherical camera (I have a Ricoh Theta S), it looks like this:

This is St. John’s Abbey in Minnesota. In this one rectangle, we actually see a full sphere’s worth of visual information. You can view this in free viewing software from Ricoh. When you drag around the image, you can look in any direction you wish. Here’s a screen capture of one view:



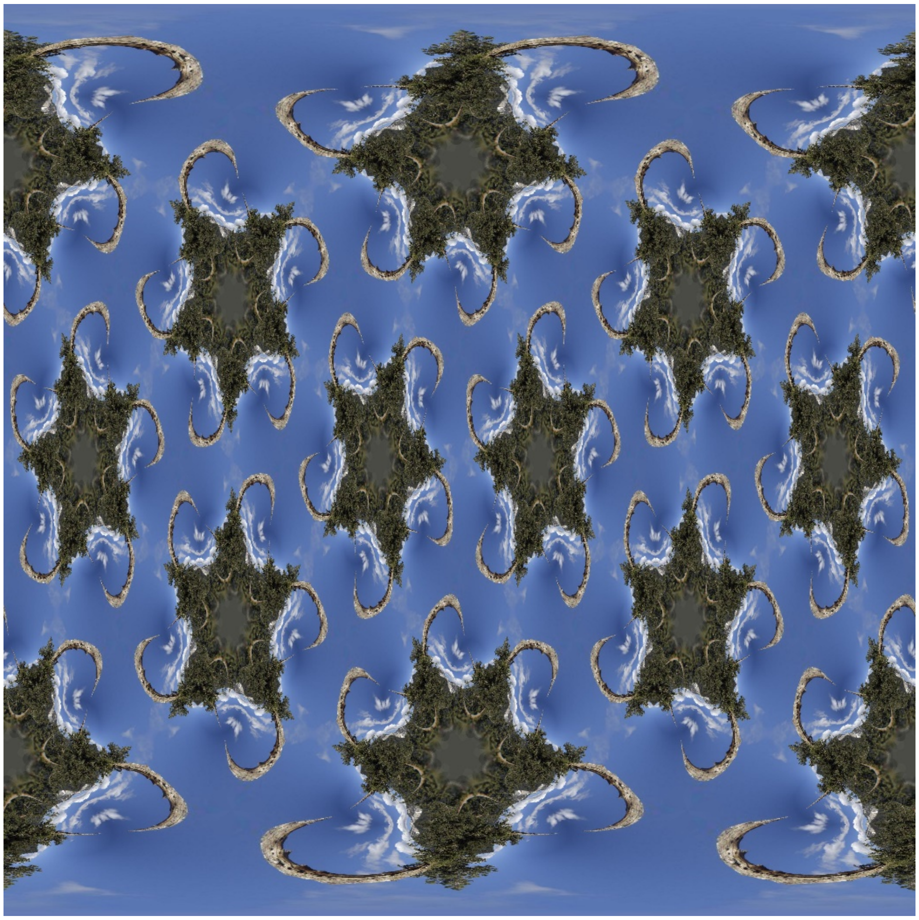
If you have uploaded a spherical image, the color functions SphereImage and SphereImageT are the best ones to use.

Using photographs to create symmetry on the sphere

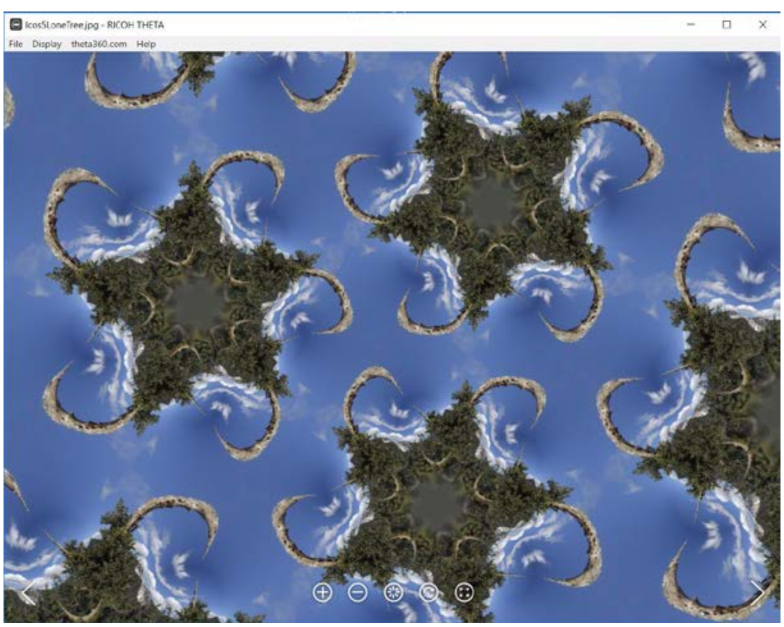
To make the icosahedrally symmetric image shown above, I uploaded my Lone Tree image and selected the Disk to Sphere color option and the function Icos 5-C poles. Mathematically, this is a *meromorphic function of z* whose poles are at the centers of 5-fold rotational symmetry, transferred to the sphere using stereographic projection, but you don’t need to know that to play with symmetry.

For this example, I left all the defaults in place, but I played with the Set on Plane feature, shown at the bottom right of the preview window. One part of understanding the symmetry is knowing that the regular icosahedron and regular dodecahedron are dual solids, so that an image with icosahedral symmetry will also have dodecahedral symmetry. We should see centers of 5-fold, 3-fold, and 2-fold rotational symmetry, for the icosahedral vertices (or dodecahedral face centers), the icosahedral face centers (or dodecahedral vertices), and the icosahedral edge centers.

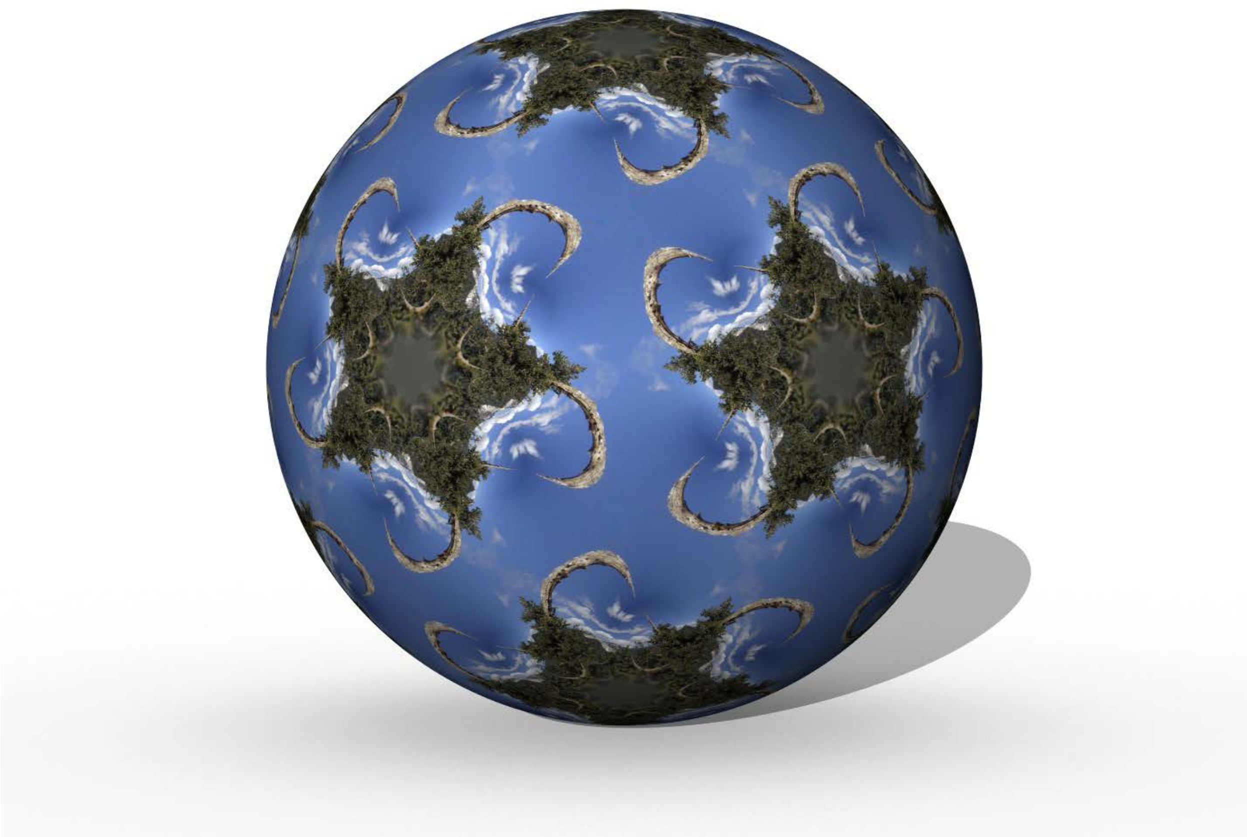
For the final output file, I used the Export Image option from the File menu, specifying a 3000 by 3000 pixel output:



You can learn to see this as having icosahedral symmetry, but it’s not easy. I put this into the Ricoh viewer and saw the following. It’s as if I’m on the inside of a giant sphere.



In this view, the 5-, 3-, and 2-centers are easier to see. But my main application for these images is to use them as texture maps on spheres in Photoshop. Here's how this one looks when pasted onto a ball:



Functions: This version of the program provides various ways to create tetrahedral, cube/octahedral, and icosahedral symmetry. For cube symmetry, just use the Tetra function and be sure to choose n and m values in each term so that $n - m$ is a multiple of 4.

Snapshot: Whenever you like what you see, push the Snapshot button (Ctrl-D/Cmd-D). If your later choices cause you to wander away into something you don't like, you can return to where you were. Searching for a great pattern really can feel like wandering in a maze. This drops a breadcrumb. To save a high-resolution version, select **Export** from the File menu. Beginners should use an aspect ratio of 1:1.

Overall: Finding a great pattern is a matter of luck and experience. Get to know the pattern types by trying to make a beautiful example of each. We hope you enjoy our mathematical kaleidoscope!