

# MapArt OpenSource Viewer: A Layered Solution for Viewing Scientific Imagery

Matthew Carroll (1), Reed Tomlinson (1), Melvin Wachowiak (2), E. Keats Webb (2); (1) MapArt Systems, LLC; Chattanooga, TN; (2) Smithsonian Institution—Museum Conservation Institute; Washington, DC

## Abstract

*Researchers and conservators use a variety of scientific and computational imaging techniques for non-destructive examination and documentation: recording the condition of an object, informing care and treatment, archiving, and contributing to a better understanding of an object. Acquisition of a variety of images creates a need to digitally integrate them in a way that is accessible and useful, allowing dynamic comparison and movement through layers; zooming in and out of features; changing the opacity of layers to compare defects, changes, or materials; rearranging the layers; and measuring distance and areas. The MapArt OpenSource Viewer created by MapArt Systems with testing and input from the Smithsonian's Museum Conservation Institute (MCI) addresses the shortcomings in current methods of viewing image data from multiple techniques. The MapArt OpenSource Viewer uses GIS (geographic information system) based software to overlay scientific images of cultural heritage objects in a dynamic and accessible interface allowing researchers and conservators the ability to virtually "walk through" an object in a multispectral and multidimensional way.*

## Introduction

As stated in *The AIC Guide to Digital Photography and Conservation Documentation*, "Documentation is basic to the ethical practice of conservation" [2]. Conservators have "an obligation to produce and maintain accurate, complete, and permanent records of examination, sampling, scientific investigation, and treatment" with the purpose of this documentation including establishing the condition of cultural property, aiding in the care of cultural property in informing future care and treatment, and "increasing understanding of an object's aesthetic, conceptual, and physical characteristics" [1]. Researchers and conservators use a variety of scientific and computational imaging techniques to accomplish this including reflectance transformation imaging (RTI); high dynamic range imaging (HDRI); gigapixel imaging; microscopy; digital radiography; and multispectral imaging: UV (ultraviolet) induced fluorescence, IR (infrared) reflectography, and reflected IR digital photography. Acquisition of such a variety of images creates a need to digitally layer them in a way that is accessible and useful, allowing dynamic comparison and movement through layers; zooming in and out of features; changing the opacity of layers to compare defects, changes, or materials; rearranging the layers; and measuring distance and areas.

The MapArt-MCI project addresses shortcomings in current methods of viewing image data from multiple techniques. There

are zoomable viewers available and even some options for layering images in zoomable browsers, but there are limited options for viewing several images in a dynamic and accessible way. One of the most feasible options is to open multiple images in different windows, downsizing and reopening those windows, or shuffling through them to compare and contrast images that are not easily aligned. One can then zoom into one image, but the zoom level might not match that of another image in a different window. Another option would be to create a large file using image-processing software, with each acquisition technique included as a layer. The layers can be accurately aligned with options to change layer order and opacity, but the oversized file would be restricted to the proprietary software, with no export option for a layered image, thus limiting the accessibility and viewing capabilities of the composite image. Not everyone has the same image processing software or the same version, and working with large files can be a timely experience depending on the computer's processor.

The priority for this project was to create a viewer with an easy-to-use interface that would be accessible to those acquiring and processing the scientific imagery and also the conservators and researchers using the images to better inform their work. A viewer is needed that can overlay a variety of multi-resolution images with an ability to turn layers on and off and change the opacity and layer order on the fly. The viewer needs to be accessible (cost, non-proprietary, etc.) and maintain full interactivity (not editing capabilities but turning layers on and off, adjusting opacity, etc.) and ideally not include downloading additional plug-ins and software. Priorities were set for using as much screen space to view the images (not just using a fixed or small portion of the browser window) and for measurement capabilities.

## The MapArt OpenSource Viewer

The MapArt OpenSource Viewer created by MapArt Systems with testing and input from the Smithsonian's Museum Conservation Institute (MCI) addresses the shortcomings in current methods of viewing image data from multiple techniques. The MapArt OpenSource Viewer uses GIS (geographic information system) based software to overlay scientific images of cultural heritage objects in a dynamic and accessible interface. This allows researchers and conservators the ability to virtually "walk through" and rearrange image layers of an object in a multispectral and multidimensional way.

The MapArt OpenSource Viewer runs on a local server and allows users to upload up to 10 images of the same extent. The images are "tiled" with the Geospatial Data Abstraction Library (GDAL), and an OpenLayers document is created that treats each image as a layer.

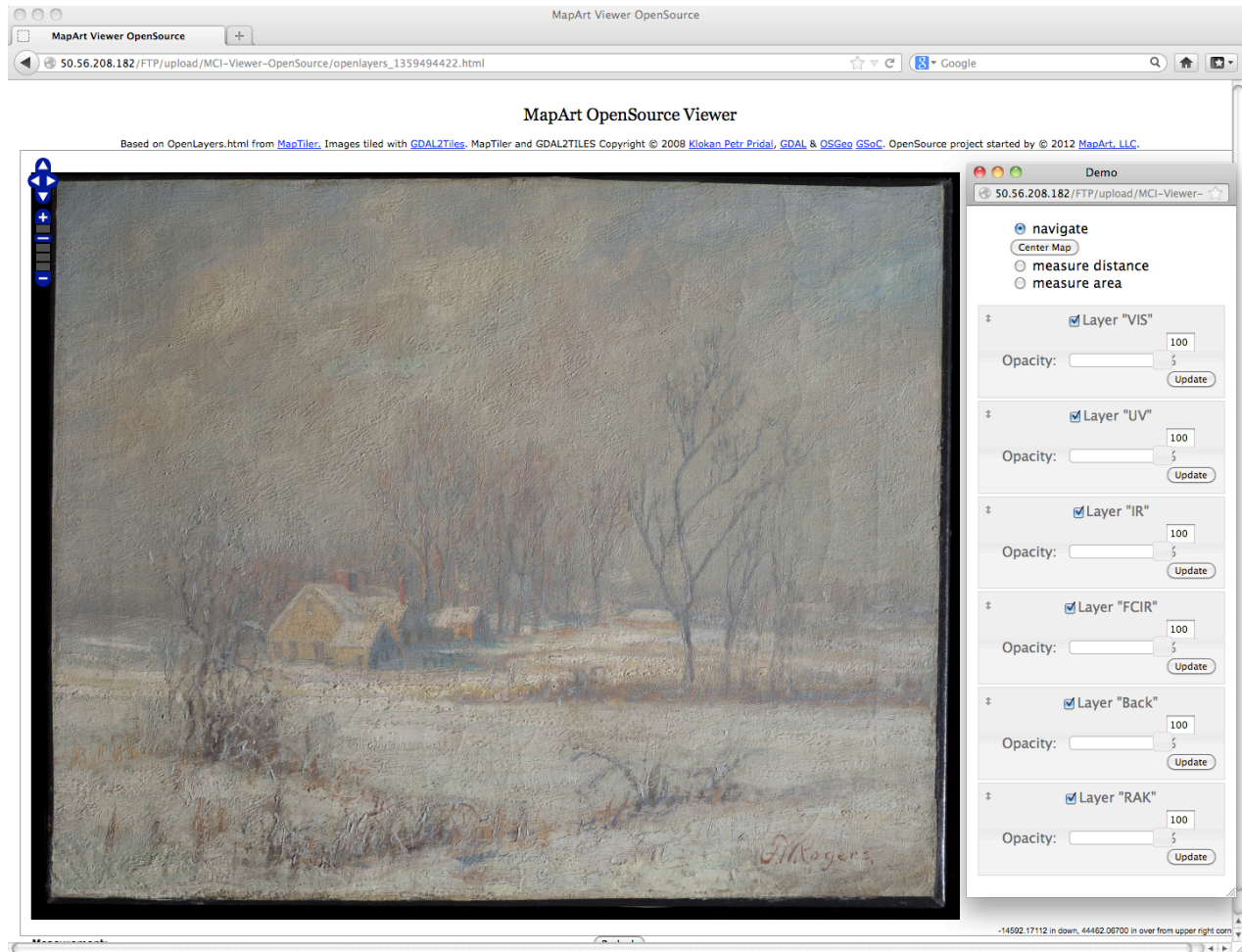


Figure 1. Screenshot of the MapArt OpenSource Viewer with image layers of a painting by FW Rogers.

The features include:

- Controls that allow zooming in and out and panning around layers.
- Layer order can be changed by dragging layer controls up and down in the list at the bottom of the page. This changes the order (z-index) of the layers, so they can be moved up and down.
- Layer opacity can be changed with the slider in each layer control, or by entering a number in the text box and hitting “update.”
- Layers can be turned on and off by toggling the checkbox in each layer control. A layer is “turned off” by switching its opacity to zero.
- Tools that can measure length and area.
- Controls can be “undocked” and will appear in a new window to maximize the image view in the browser.
- The bottom right corner of the viewer includes a read-out of the current mouse position. This read-out is real-world measurements from the nearest corner, based on the dimensions entered on “initiate.php”.

## Software Methodology

### PHP Upload

The first page of the viewer accessed in an Internet browser, *upload.php*, “uploads” up to 10 images into the “uploading/” directory. On a local server, this has the net result of copying those files to the “uploading/” directory, but the transaction is handled by the local server in the form of a POST request. Transactions are therefore subject to the limitations of the local server, such as upload size limits, maximum execution time, and other parameters. On submitting the upload form, the data is sent back to *upload.php* (“action=?”). This same file handles the actual upload of the files and on completion the browser is redirected to *initiate.php* with GET parameters specifying file names.

*Initiate.php* works by creating an AJAX call to *process.php*, and then repeatedly reading the contents of “infile.txt” on the local server. *Process.php* calls various GDAL commands to process the input images, and directs all output from those commands to “infile.txt”. The GDAL methodology is as follows for each image.

### Find Image Size

The size of the input image is called by “gdalinfo”. For example: *gdalinfo file.jpg*

The output from gdalinfo is parsed to get the image size. From this output, IMAGE\_HEIGHT (referred to as “lines”) and IMAGE\_WIDTH (a.k.a. “pixels”) are acquired.

### Tile the Image

The image is now “tiled,” or broken down, into 256x256 pixels squares. This is done for various “zoom levels,” or image resolutions.

```
gdal2tiles.py -p 'raster' -z 0-5 IMAGE_NAME >> infile.txt
```

The list of arguments passed to the *gdal2tiles.py* program specifies how the program will run. The first argument, “-p”, specifies the “tile cutting profile” as “mercator,” “geodetic” or “raster.” The *gdal2tiles.py* program is designed to tile geographically registered images. By passing the “raster” value for this argument, the program is told that the image contains no geographic information and will instead be treated as a normal rectangular image.



Figure 2. Image of painting at zoom level 0, a single image made up of a 1 tile x 1 tile with a total resolution of 256 x 256 pixels.

The second argument, “-z”, specifies the “zoom levels” at which the image will be tiled. At each level, the image is broken down into a certain number of 256 x 256 pixel “tiles.” At level 0, this is a single image, a 1 tile x 1 tile, with a total resolution of 256 x 256 pixels, or 65,536 pixels (65 kilopixels or kpx). Since the image does not perfectly fit into a 256 x 256 tile, notice the blank space at the top of the image (see Fig. 2). This space has opacity set to zero (transparent). These are PNG files, so the alpha layer for these pixels is set to zero.

At level 1, the image is broken into 4 tiles. The total resolution is now (2x2)x(256x256)=262,144 pixels or 262 kpx.



Figure 3. Image of painting at zoom level 1 where the image is broken up into four tiles.

At each consecutive level, the vertical and horizontal resolutions are both doubled. Therefore, each level has four times the resolution of the level before.

Level	Resolution	Tiling time*
0	65kpx	0.730s
1	262kpx	0.748s
2	1.0mpx	1.247s
3	4.2mpx	3.097s
4	16.8mpx	8.461s
5	67mpx	29.02s
6	268mpx	103.8s
7	1.07gpx	400.5s

\*All times were recorded on a machine with a 2-core AMD Turion 64-bit processor with 747MiB RAM, using a 425 kilopixel image.

In this application, images are tiled from zoom level 0 (resolution less than 65 kpx) to zoom level 5 (resolution 67 megapixels or mpx). Ideally, this should go from level 0 to a resolution that is just beyond the resolution of the input image. For instance, an input image that is 200mpx should get tiled from level 0 (effective resolution 65 kpx) to level 6 (effective resolution of 268 mpx).

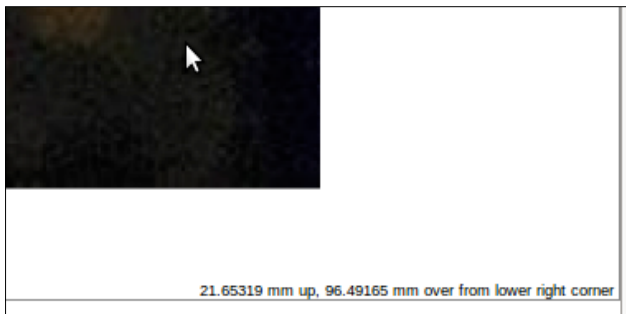
Note that the effective resolutions listed above are maximums that would only apply to perfectly square images. If an image is not perfectly square, there is some “dead space” to the top or right



of the image in the level zero 256 x 256 image, which is written as simple transparency. This “dead space” cannot be counted in the effective resolution of the image. At high zoom levels where a tile may consist of only pixels in this “dead space,” such tiles are not even created. Finding the true resolution of the image at each zoom level is further complicated by the *gdal2tiles.py* program. For a more detailed discussion of tiling and its effects on measurement tools, consult the technical documentation that accompanies the open-source software package.

### OpenLayers

Once the images are tiled, an OpenLayers-based HTML file is written to the disk. This file uses a local version of the OpenLayers library, which includes minor changes in the file `OpenLayers-2.11/lib/OpenLayers/Control/MousePosition.js`. These changes override the read-out of the mouse position in the bottom right corner of the map, changing the output from coordinates to units (inches, mm, etc) from the nearest corner.



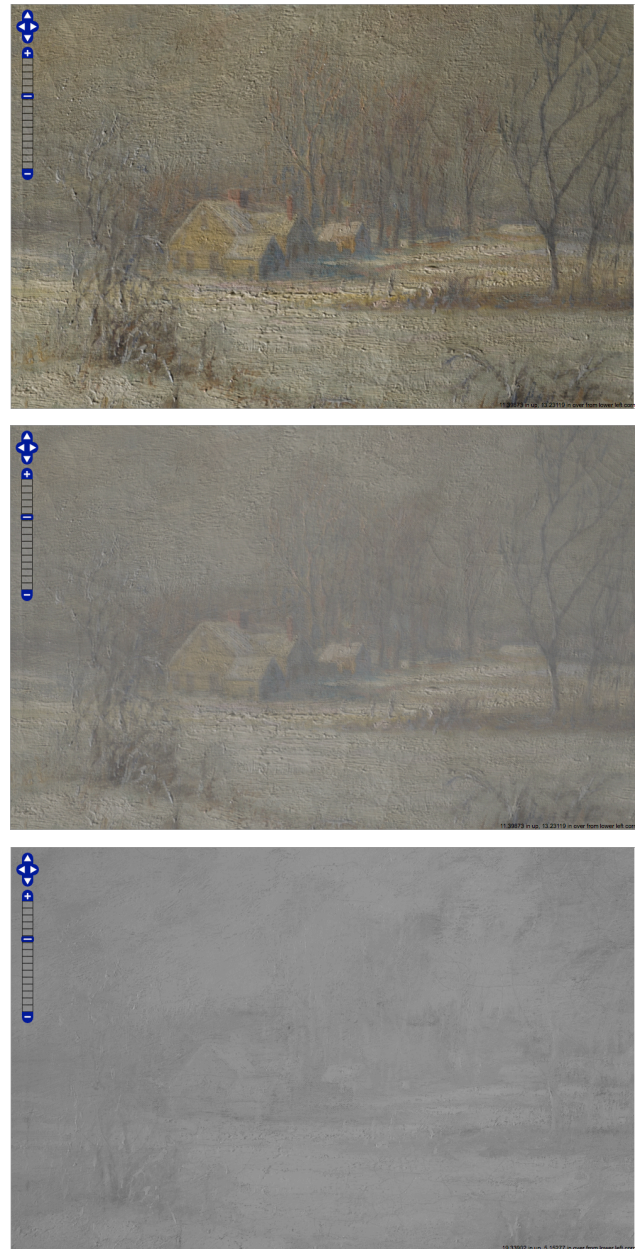
**Figure 4.** A screenshot showing the read-out of the mouse position found in the bottom right corner of the Viewer.

### Applications

The MapArt OpenSource Viewer allows the user the ability to virtually “walk through” a museum object in a multispectral and multidimensional way. For example, the various image layers of an easel painting could be compared and examined—the structure with digital radiography; under-drawing with IR imaging; surface geometry with raking light; and changes with UV-induced fluorescence—all coming from a web-based file made up of a number of overlaid scientific images. With a visual integration of these various scientific images, there is more potential information to be gained by the researcher and conservator. For example, the crack patterns seen in the raking light view can be investigated in context with the assembly of the stretcher bars, tacks and overall structure of the painting seen with digital radiography. Different narrow bandwidths within the infrared can be compared looking at under-drawings (IR imaging), base paintings (X-ray and/or IR imaging) and the final painting (visible light photography) to learn about the artist’s workflow and process.

The MapArt OpenSource Viewer is ideal for integrating multiple imaging techniques for flat objects including but not limited to paintings, paper, photographic materials including daguerreotypes, etc. Aside from documenting an object and assessing its current condition by compiling the comprehensive imaging techniques, the MapArt OpenSource Viewer also has potential to be used to compare before and after treatment images

of an object or compare multiple copies of an object to note the differences.



**Figure 5, 6 & 7.** Top: A screenshot of a raking light image of a painting in the MapArt OpenSource Viewer. Middle: A screenshot of 50% opacity of raking light layer over the infrared image. Bottom: A screenshot of 0% opacity of raking light layer over infrared image.

The MapArt OpenSource Viewer will not change image acquisition or imaging techniques used, but instead creates a tool for the end user to better utilize and view the acquired data sets. Changing the order and opacity of these stacked layers will allow dynamic and detailed comparison across wavelengths or imaging techniques. In addition, the software is based on open source GIS

code, which will increase accessibility for the client, in this case the researcher and conservator, in a way that is currently not available to the cultural heritage imaging world. Passing along the URL of the project allows easy sharing of the assembled files.

## Conclusions

The MapArt OpenSource Viewer is a layered solution for researchers and conservators to view a variety of scientific and computational image data from multiple techniques. MapArt can help to document and establish the condition of an object, aid in the care of the object informing conservation treatment, and increase the understanding of an object. The viewer addresses some of the shortcomings of current methods of viewing image data from multiple techniques in a dynamic and accessible way. Due to the nature of open-source software, there will continue to be some limitations and known issues, but there is also the opportunity for others to aid in the continued development of the viewer.

MapArt Systems has used a similar platform as the OpenSource Viewer, combined with a cutting-edge, cloud-based back-end, to bring advanced imaging tools, intuitive notations, and instantaneous communication in a proprietary, full feature version for the art conservation world. This viewer's feature set is aimed at simplifying processes that are currently difficult or time-consuming for imaging specialists and art conservators, such as creating text, polygon or mixed media annotation overlays, quickly comparing images from different sources, and rectifying images of different scales or orientations into one view. Multiple users can "share" projects, and see updates, notations and questions from other users, allowing for instant communication and collaboration from anywhere in the world. While the full version MapArt Viewer was created for art conservators, its applications extend into all aspects of imaging science. The MapArt Viewer is currently in private beta testing.



**Figure 8.** A screenshot of the full feature version of the MapArt Viewer looking at a daguerreotype.

## References

- [1] "AIC Code of Ethics and Guidelines for Practice." American Institute for Conservation of Historic and Artistic Works. AIC, Aug. 1994. Web. 15 Jan. 2013. <<http://www.conservation-us.org/index.cfm?fuseaction=page.viewpage>>.
- [2] Warda, Jeffrey, Franziska Frey, Dawn Heller, Dan Kushel, Timothy Vitale, and Gawain Weaver. *The AIC Guide to Digital Photography and Conservation Documentation*. 2nd ed. Washington: American Institute for Conservation of Historic and Artistic Works, 2011. Print.

## Author Biographies

*Matthew Carroll is a Co-Founder of MapArt Systems LLC. His primary focus is directed towards product development and creative problem solving in the field of augmented reality. Prior to MapArt Matthew worked for the Knoxville Museum of Art as the Registrar, and has a long background in electronic controls and fabrication as the Design Manager for A. Carroll Company. In 2004 Matthew earned a B.F.A. in Painting and Drawing from the University of Tennessee at Chattanooga.*

*Reed Tomlinson received his BA in Computer Science and English at Sewanee: The University of the South (2010), where he focused on geographic information systems (GIS) programming and artificial intelligence. He has worked for MapArt Systems LLC for the past year, where he has focused on augmented reality (AR) solutions and web applications.*

*Melvin J. Wachowiak has been at the Smithsonian since 1989. He received his M.S. from Winterthur Museum, Art Conservation Program, University of Delaware and a B.S. in studio art from Springfield College. Projects have included fine arts, anthropology and archeology, and history of technology. Since 2004, he has lead MCI's use of new and advanced spatial and spectral imaging techniques such as 3D scanning, 3D microscopy, and other computational and digital imaging.*

*E. Keats Webb is a digital imaging specialist at the Smithsonian's Museum Conservation Institute. She received her BFA in photography from the University of North Carolina at Chapel Hill. Her work focuses on scientific and computational imaging for the research and conservation of cultural heritage objects at the Smithsonian.*