

# Hyper3D: 3D Graphics Software for Examining Cultural Artifacts

MIN H. KIM, KAIST, Korea

HOLLY RUSHMEIER, JOHN FFRENCH, IRMA PASSERI, and DAVID TIDMARSH, Yale University, USA

---

Art conservators now have access to a wide variety of digital imaging techniques to assist in examining and documenting physical works of art. Commonly used techniques include hyperspectral imaging, 3D scanning and medical CT imaging. However, viewing most of this digital image data frequently requires both specialized software, which is often associated with a particular type of acquisition device, and professional knowledge of and experience with each type of data. In addition, many of these software packages are focused on particular applications (such as medicine or remote sensing) and do not permit users to access and fully exploit all the information contained in the data. In this paper, we address two practical barriers to using high-tech digital data in art conservation. First, users must deal with a wide variety of interfaces specialized for applications besides conservation. We provide an open-source software tool with a single intuitive interface consistent with conservators' needs that handles various types of 2D and 3D image data and preserves user-generated metadata and annotations. Second, previous software has largely allowed visualizing a single type or only a few types of data. The software we present is designed and structured to accommodate multiple types of digital imaging data, including as yet unspecified or unimplemented formats, in an integrated environment. This allows conservators to access different forms of information and to view a variety of image types simultaneously.

Categories and Subject Descriptors: I.3.4 [Graphics Utilities]: Software support—*Computer Graphics*

General Terms: Computer Graphics

Additional Key Words and Phrases: open-source software, visualization, cultural heritage

## ACM Reference Format:

Kim, M. H., Rushmeier, H., ffrench, J., Passeri, I., and Tidmarsh, D. 2013. Hyper3D: 3D Graphics Software for Examining Cultural Artifacts. *ACM J. Comput. Cult. Herit.* 0, 0, Article 0 (0), 18 pages.

DOI : <http://dx.doi.org/10.1145/0000000.0000000>

---

## 1. INTRODUCTION

Many digital imaging technologies have become valuable tools in physical art conservation. The nondestructive nature of imaging, and the rapid acquisition of data over large areas, offer great advantages over

---

Min H. Kim acknowledges Korea NRF grants (2013R1A1A1010165; 2013M3A6A6073718), Samsung Electronics (IO121207-05485-01) and MSRA for additional support. Holly Rushmeier acknowledges a Seaver Institute grant. Authors' addresses: Min H. Kim, KAIST, Dept. of Computer Science, 373-1 Guseong-dong, Yuseong-gu, Daejeon, South Korea 305-701; email: [minhkim@kaist.ac.kr](mailto:minhkim@kaist.ac.kr); Holly Rushmeier and David Tidmarsh, Yale University, Dept. of Computer Science, PO Box 208285, New Haven, CT 06520; emails: [holly@acm.org](mailto:holly@acm.org), [tidmarsh@aya.yale.edu](mailto:tidmarsh@aya.yale.edu); John ffrench and Irma Passeri, Yale University Art Gallery, 1111 Chapel Street, New Haven, CT 06510; emails: [john.ffrench@yale.edu](mailto:john.ffrench@yale.edu), [irma.passeri@yale.edu](mailto:irma.passeri@yale.edu). This work was originally presented at the 13th International Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST 2012) and received a Best Paper Award; it was then referred to ACM JOCCH with substantial revision and extension.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or [permissions@acm.org](mailto:permissions@acm.org).

© 0 ACM 1556-4673/0/-ART0 \$15.00

DOI : <http://dx.doi.org/10.1145/0000000.0000000>

other techniques such as taking small physical samples or taking measurements at individual points. However, the full potential of digital imaging has not been realized because of the difficulty of assembling the various image types in a common framework to develop an integrated view of an artifact. In this paper, we present a new open-source tool to facilitate the viewing and navigation of multiple image data types in a form convenient for art conservators.

Various types of imaging techniques, generally targeted at other applications, have been developed and are becoming ever more popular among conservators. Clearly, digital camera images and 2D scans of historic photographs remain useful in documenting works. Beyond these basic tools, however, advanced imaging technologies such as X-rays, hyper- and multispectral imaging, 3D scanning, and computed tomography (CT) imaging are being used with increasing frequency.

Current software environments for advanced imaging techniques are nevertheless generally bound to specific imaging hardware and/or applications. Companies developing the hardware provide system-specific software applications, often at extra cost. In some cases, software is provided by other sources, but for a single end application – e.g. medical applications for CT scans, or mechanical manufacturing applications for 3D scanners. Such systems do not facilitate importing other data types, and include many modules and interface options irrelevant to the needs of art conservation. Worse yet, various software systems use different interface conventions for tasks that should be similar across data types, such as navigation, zooming and lighting. Further compounding the difficulty is the inability of many software applications to store and recall user-generated content about the images that they display. In the event that a particular application does support saving notes or bookmarks about an image or project, the resulting annotations may be incompatible for use in other applications used to visualize additional data formats.

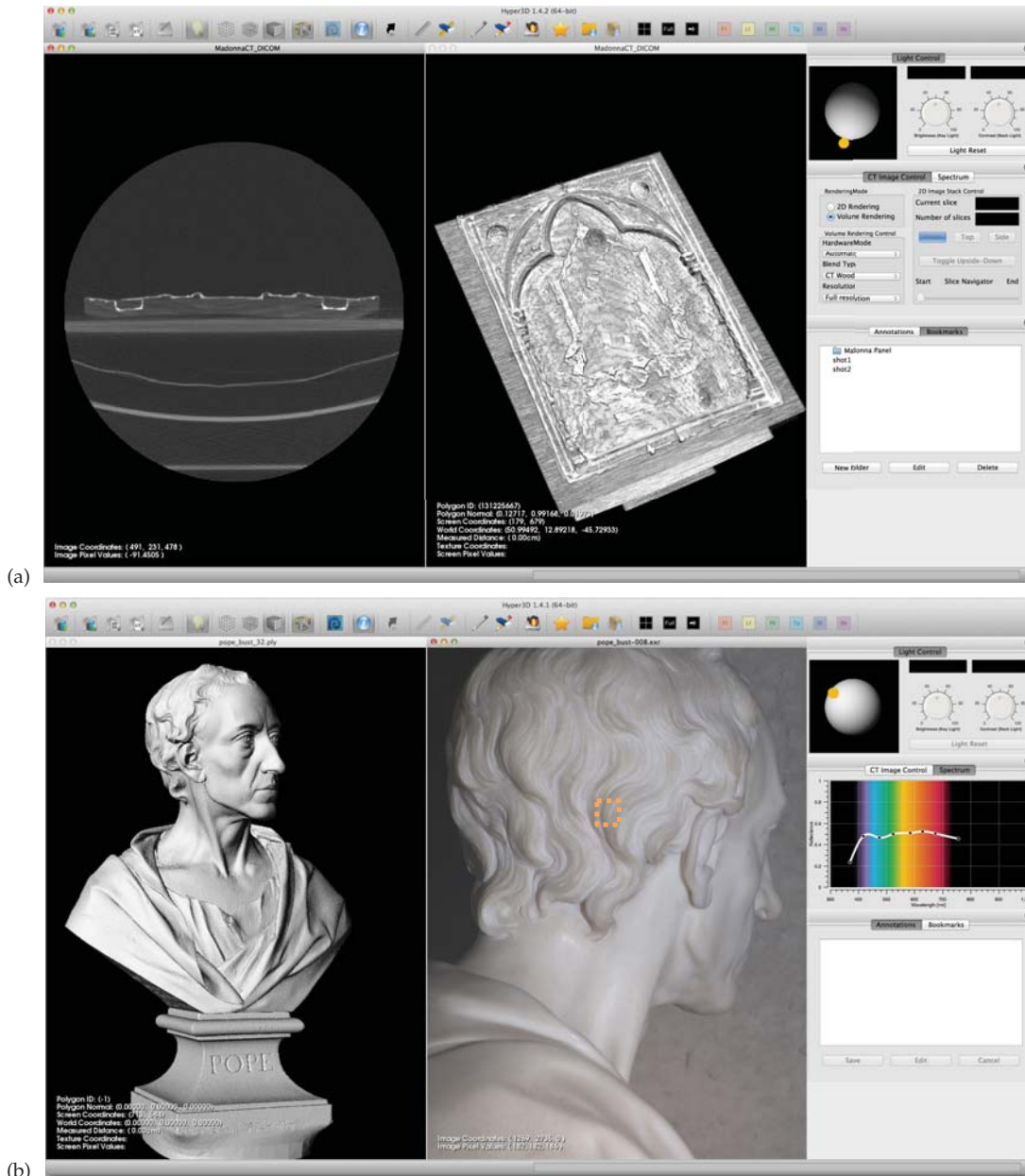
The cost, complexity and interface inconsistency of software for different types of imagery have become obstacles for conservators who try to adopt new imaging technologies in their work. Conservators might have to use multiple workstations to support running multiple software systems simultaneously to examine e.g. a CT scan, a 3D scan, and a stack of hyperspectral images of a single artifact. Furthermore, they must constantly adjust to the different interface and annotation conventions of the software when switching between workstations. A tool is needed that can import data without unnecessary functionalities from other applications and that possesses an intuitive common interface.

Beyond importing various known imaging formats and saving notes and metadata about such images, conservators need an extensible framework that allows the existing interface to be used with future types of data. For instance, if a conservator receives a 3D model with hyperspectral textures, it should be possible to use the software to investigate the data with the same navigation and lighting controls as used with previous 3D models, and with the same spectral displays as used with 2D spectral images.

The field of computer graphics has developed many open-source software frameworks for processing 3D geometric models [Cignoni et al. 2008; Lévy 2008; Möbius and Kobbelt 2010]. These projects have been successful in their own right and have also had a broad impact on other fields. Nevertheless, these efforts remain focused on 3D geometric processing, although some have been extended to support additional operations.

In this paper, we address the lack of equivalent software packages that permit art conservators to easily use digital data for art restoration. We provide an open-source imaging software tool, shown in Fig. 1, that reduces the complexity in using data from high-end imaging technology. In contrast to previous open-source geometric frameworks, the tool can display multiple types of data rather than just 3D models. We avoid replicating this previous work by focusing on navigating and viewing images, rather than offering processing capability.

We describe the development of this software designed to give art conservators easy access to high-end imaging. In the spirit of previous software projects in computational geometry, we provide this software as open-source to enable future development by other users.



**Fig. 1:** Two screenshots of the software displaying different cultural artifacts. (a) *Tempera on panel with gilding, The Coronation of the Virgin, Italy ca. 1325, by Jacopo del Casentino*. On the left, a CT 2D stack visualization; on the right, a volume rendering of the polychrome panel. The column on the right-hand side shows the light controls for the 3D model/volume rendering visualization (key/fill light), the CT stack data navigator, and the volume rendering options. The widget at the bottom right presents the interface for loading and storing image bookmarks. (b) *Marble bust, Alexander Pope, 1741, by Louis François Roubiliac*. At left, a 3D laser scanning model; at right, a hyperspectral image. At the middle of the column to the right, the plot widget illustrates the spectral reading of the region enclosed by the orange dotted line.

Our contributions are:

- The integrated visualization of various types of imaging data for art restoration in a single software package.
- A coherent system for saving and retrieving user-generated metadata and annotations alongside the images displayed.
- A publicly accessible open-source framework for extensive future modification.

## 2. BUILDING ON PREVIOUS WORK

In this section we review the types of imaging data commonly in use by art conservators and the background for the selection of our software’s basic framework.

### 2.1 Data Types in Art Conservation

Various imaging technologies have been used in the field of art conservation for many decades [Fischer and Kakoulli 2006]. Infrared reflectography (IRR) with narrow band-pass filters has been used in the study of paintings [Delaney et al. 1993]. Mansfield et al. [2002] and Attas et al. [2003] applied multi-spectral imaging to examine drawing and oil painting. Spectral imaging can, for example, detect variations in material indicating repairs that have been made to the original works.

Full volumetric scanning, in particular CT scanning, is used to investigate the structure of artifacts, especially those made of wood and bone [Ghysels 2005]. For example, CT scanning was used to obtain a full volumetric map of a Fang reliquary head from Gabon to answer questions such as the internal morphology of and the degree to which oils had permeated the wooden object [Kaehr et al. 2007].

Conservators can now make simultaneous use of multiple types of digital information in understanding, documenting and conserving works of art. In studying the work of Raphael, Roy et al. [2007] demonstrated the use of infrared and ultraviolet imaging and 3D profilometry. The work of Arbace et al. [2012] shows how scanned historic photographs, 3D shape scans, and color data can be brought to bear in the reconstruction of a work. Mohen et al. [2006] describe a remarkable set of imaging technologies as well as other physical measurements all used in a definitive study of the state of Leonardo da Vinci’s *Mona Lisa*. In general, conservation projects dealing with multiple imaging modalities have focused on the study of a particular object, rather than a tool that can be used to study many objects. In most cases technical experts in different disciplines, and in the different imaging modalities, were needed to acquire and interpret the data.

While conservators rely on a variety of digital image formats to visualize works of art, they also need to generate data about the objects that they visualize to assist them in formulating conclusions and to share with colleagues, researchers, and the public. For example, conservators may wish to take a screenshot of the object currently being visualized, to save (and later retrieve) a “bookmark” of an object from a particular camera angle with certain display and lighting conditions, or to “tag” parts of the object and associate them with annotations elaborating upon noteworthy features. In addition to free-form annotations, conservators may also wish to catalog information about the object using controlled metadata systems. Popular standards for describing works of art and visual culture include the Getty Research Institute’s Categories for the Description of Works of Art [Baca and Harpring 2009] and the Visual Resources Association’s VRA Core [VRA 2013].

In this project we seek to provide art conservators access to different types of digital imaging data and metadata in an integrated environment. Experts might still be needed to acquire the data (as with CT scans) or to process data into a useful form (as with the conversion of range images into 3D models). However, given the processed data, our tool should allow conservators to inspect and annotate the results in their day-to-day work, without the assistance of technical experts. Working as a team including a museum conservator (I. Passeri) and a museum director of visual resources (J. French), we compiled this list of digital

imaging types found useful in practice: conventional 2D (RGB) digital photography, 2D high-dynamic-range (HDR) imaging [Debevec and Malik 1997], reflectance transformation imaging (RTI) [Palma et al. 2010], 2D X-ray imaging [Tsui 1994], 2D hyperspectral imaging (including UV and IR channels), 3D scanning with texture [Bernardini et al. 2001], 3D imaging spectroscopy (including hyperspectral textures) [Kim et al. 2012], and 3D volumetric computed tomography [Ney et al. 1990]. Our framework is flexible enough to handle all of these, and with the exception of RTI, all have been implemented in our viewer. The structure and interface we have developed are designed to accommodate the future addition of RTI.

## 2.2 Previous Software Frameworks

The need for inspection tools for cultural heritage professionals has been a topic of discussion for some years. Previous work on cultural heritage specific software, however, has focused on the flexible and intuitive manipulation of a single type of data. One example is the Virtual Inspector [Callieri et al. 2008], which focuses on the easy setup of annotated 3D models for viewing by museum visitors, rather than on using 3D models as part of a technical examination by a conservator.

Widely used open-source image processing programs include ImageJ [Abràmoff et al. 2004] and the GNU Image Manipulation Program (GIMP) [Kylander and Kylander 1999]. ImageJ, developed originally for and still skewed toward medical imaging applications, allows the importing of multiple types of 2D images and provides many image processing operations such as edge detection, cropping, and cut-and-paste. ImageJ does not deal with 3D data, in particular with 3D mesh objects. GIMP also provides multiple image processing applications, but is more focused on the artistic editing of images rather than the analysis of images that ImageJ provides. Like ImageJ, GIMP does not deal with 3D data.

The scope of our work includes the visualization of 3D geometry with texture, and there are several useful open-source frameworks in the field of geometry processing. MeshLab [Cignoni et al. 2008] is a mesh processing software framework that can handle various types of 3D mesh data and enables the use of various geometric operations. This framework also provides the independent VCG library. MeshLab has proved extremely useful to the field of cultural heritage in producing 3D models from scan data. Graphite [Lévy 2008] is also a complete software package designed for research in geometric processing that can handle various types of 3D data and includes implementations of various geometry processing algorithms. OpenFlipper [Möbius and Kobbelt 2010] is a more commercially oriented open-source project. Like other software, OpenFlipper is designed as a geometric processing toolkit. The software is commercially available because the core OpenMesh engine was published under the GNU Lesser General Public License (LGPL) license, as opposed to other frameworks which use the GNU General Public License (GPL). These open-source frameworks are very well written and well maintained. However, they cannot readily be expanded to viewing types of data such as hyperspectral images and volumetric rendering of CT data.

There have also been numerous efforts to provide viewers for 3D volumetric scan data, typically provided in the Digital Imaging and Communications in Medicine (DICOM) format. One prominent example is the popular OsiriX software [Rosset et al. 2004], which has become both a widely used open-source project and a commercial software package with clearance from the United States Food and Drug Administration. While OsiriX imports data from multiple image modalities (e.g., magnetic resonance and computer tomography), it focuses on volumetric scan data, and provides functionalities specific to medical applications.

Some processing and visualization systems chronicle the sequence of actions that users take to produce an image or model. Tools such as the VisTrails workflow management system [Callahan et al. 2006] and the PTMbuilder software for automating the RTI file generation process [Mudge et al. 2008] track the provenance of the resulting data in order for users to validate its accuracy and to reproduce scientific conclusions. Unlike these projects, we do not seek to record the entire history of user interaction with the data indiscriminately, but permit the user to decide which information and viewpoints are semantically meaningful and worthy of preservation.

Other visualization and modeling systems, such as the Mental Canvas sketch-based modeling tool [Dorsey et al. 2007], have the capacity to create “bookmarks” that allow the user to recall a particular perspective or insight by recording the current state of the system. To implement our own bookmarking feature, we rely on the concept of the “visualization snapshot” from Mindek et al. [2013], a set of values observed and transcribed at a given point in time that documents the entire state of the program. Finally, we also draw inspiration from systems to create and share annotations between users, such as the ImaNote open-source Web-based viewing and annotation tool [Díaz et al. 2011] and the redesign of the Arrigo Showcase virtual exhibit to add standards-compliant semantic annotations to 3D models [Havemann et al. 2009].

It is important to note that we seek to integrate the *viewing* capabilities of previous projects for 2D and 3D images and 3D geometric objects. However, we do not seek to unify all of the *processing* capabilities of these programs. In our system, we assume that the image or geometric processing needed for each type of data is performed in specialized software for that type.

### 2.3 Basic Building Blocks

We are interested in building a complete software framework that can handle the visualization of various 2D, 3D, and volumetric data types. We are careful not to rewrite or reinvent existing processing software such as Photoshop and MeshLab that can be used to prepare the image files before their visualization.

All of the frameworks cited in the previous section for image, geometry and volumetric data are themselves built on standard libraries. We follow the lead of these successful projects by adopting the libraries used by these projects as our basic building blocks.

To build a complete software package, we include three main application programming interfaces (APIs): graphics/visualization, user interface, and linear algebra (see Fig. 2). First, we have adopted two graphics libraries, the Visualization Toolkit (VTK) [Schroeder et al. 2006] and the Insight Segmentation and Registration Toolkit (ITK) [Taka and Srinivasan 2011]. VTK is an object-oriented wrapper for the OpenGL pipeline, and we therefore implement OpenGL operations via the VTK API. We selected these two libraries as the software’s graphics engine because VTK supports volumetric rendering and ITK provides basic image processing modules necessary for handling CT image data, including input and output for Digital Imaging and Communications in Medicine (DICOM) files. The medical imaging application OsiriX is also based on these two graphics libraries. By using these two APIs, we can save our time and effort for implementing other functionalities, such as integrating multispectral imaging into 3D scanned data.

Second, we employ the Qt API [Dalheimer 2002] for our software’s user interface, which we want to make as intuitive as possible (see Fig. 1 for screenshots of the final product). The Qt API has not only the benefit of being cross-platform but also uses the signal-slot mechanism to handle interactions between user

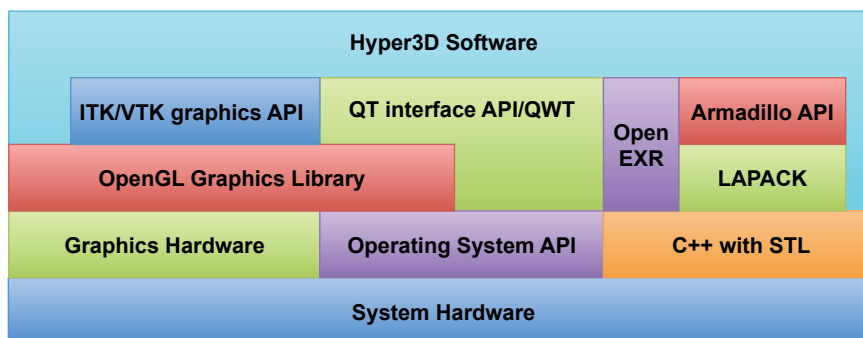


Fig. 2: This diagram illustrates how we configure the building blocks of our software architecture (described in Section 2.3).

interface objects. Qt has been commonly used in previous geometric processing software [Cignoni et al. 2008; Lévy 2008; Möbius and Kobbelt 2010]. However, the VTK libraries do not fully support the multi-document framework of Qt, and some engineering was required to accommodate both APIs within the same software. For further details, please refer to the software’s source code [Hyper3D 2013].

Finally, our software includes the Armadillo linear algebra API (that wraps BLAS and LAPACK operations) for high-performance calculations [Sanderson 2010] and the OpenEXR API [Lucas Digital Ltd. 2009] for handling HDR image data and multispectral channels by using the multi-layer feature in the OpenEXR library. This allows us to read and visualize multispectral images and textures on a hyperspectral 3D model [Kim et al. 2012].

Our software application currently supports 32- and 64-bit Windows and 64-bit Mac OS X. These two target operating systems were chosen according to the preferences of the art conservators we consulted.

## 2.4 Basic Functions and Interfaces

In addition to benefiting from the experience of previous projects, we want to follow the standards and conventions of successful user interfaces. We structure our software according to the multi-document framework, permitting users to open multiple images of different types (see Fig. 1 for a view of the interface with several documents opened). While multiple images can be treated as independent objects, they can also be collated within a single project (reflecting the project feature in software such as MeshLab), which allows the user to easily load the current state of the program at a later date and to record metadata about the images’ nature and relationship.

Following standard software conventions, our software has both a menu bar and a toolbar through which most of its functionalities are accessible (see Fig. 4(a)). The software can read 2D and 3D images and the contents of a CT directory (since DICOM medical image data is contained in a directory rather than in individual files). It provides three different types of common rendering modes (points, wireframe, and surface) and the ability to toggle the texture (if any) on a 3D model. We also allow users to illuminate objects with or without a directional light. Our software delivers conventional information about the model or image loaded, such as the polygon count, current polygon index, surface normal of the selected polygon, world coordinates, texture file information, etc. It also includes the ability to zoom in and out and to select orthogonal view options, similar to 3D geometric software. In addition, we implemented a measuring tool for 3D models (both scanned surfaces and CT volumes) that displays the length between two selected points, represented in the image panel as a straight orange line.

## 3. SOFTWARE – IMPLEMENTATION AND CURRENT FUNCTIONALITY

Recent art conservation research has been extended to the realms of hyperspectral 2D and 3D imaging and volumetric rendering, as evidenced in Roy et al. [2007]. We recognize that the needs of art conservation are different from the functionalities of existing geometry processing and medical imaging software. Our current software tries to fill in the gap between these packages in terms of software implementation and functionality. Most geometry processing applications focus on the algorithms and techniques to deal with geometric operations and support only the rendering of RGB textures, if any. These applications therefore lack the capability to visualize hyperspectral textures or to display volumetric data. Meanwhile, medical imaging applications have concentrated on volumetric rendering to depict and understand the inner structure of the human body, with no support for hyperspectral texture visualization.

### 3.1 System Design Decisions

The feedback from the art conservators drove our system design process from the choice of the fundamental APIs to the interface of design and functionalities, e.g., hyperspectral 3D model inspection, recording bookmarks, annotations and metadata on various data types. In this section, we outline our key design de-

cisions that we made through the development of our system in interaction with the professional end users and reasoning process behind them.

In order to overcome the practical barriers for art conservators to using high-tech digital data and fully exploiting all the information contained in the data, we carefully chose basic building blocks upon their requirements to inspect and explore digital data from hyperspectral imaging, 3D scanning and medical CT imaging.

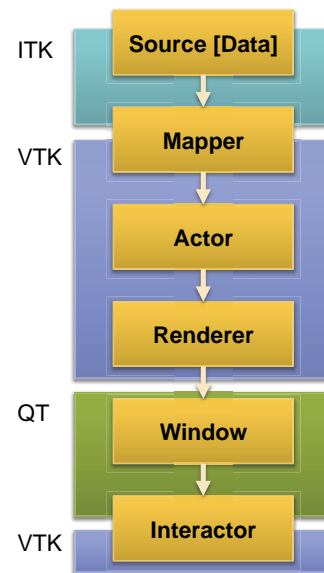
**Handling Various Data Types.** As described in Section 2.3, we first focus on visualizing two different main data types: medical CT data and hyperspectral 3D scanning. Since the art conservators have commonly used medical CT imaging scanner for geometric inspection on the surface of artifacts, our system needs to be capable of reading various types of DICOM medical data (with various compression types) and visualizing them through 3D volume rendering. In this scope, we decided to import the ITK library that includes the DICOM I/O support (see Section 2.3). It allows us to decompress various types of 2D CT image stack and convert them to 3D voxel values, which can be rendered with volume rendering in the OpenGL environment.

As described in [Kim et al. 2012], hyperspectral imaging data include more spectral channels than three RGB channels, and such data are stored in the multi-channel OpenEXR format. We additionally plug in the OpenEXR library, with which we implemented hyperspectral texture visualization across the 3D rendering pipeline. Note that this hyperspectral rendering process is lacking in previous medical imaging applications. We also implemented color mapping from hyperspectral data into human visible colors with linear algebra optimization APIs (LAPACK and Armadillo). This allows the conservators to inspect general appearance of visible colors along with hyperspectral data.

**Integrating User Interface.** The other requirement from the professional conservators is the single intuitive user interface consistent with their needs to explore and generate metadata and annotations in their inspection. Previous imaging software has allowed visualizing or exploring a specific data type or a few other types only. Its interface stands modal without allowing integration with other data types visualized.

We first chose the Qt API to accomplish an intuitive graphics user interface and the VTK library to render the 3D voxel data converted from CT medical scan. This enables the conservators to navigate the three-dimensional topology of artifact intuitively in the OpenGL rendering environment. The window objects are mainly rendered by the Qt API, and the mouse interaction and events are controlled by both Qt and VTK. Fig. 3 summarizes how we organize the three different APIs (ITK, VTK and Qt) to render 3D objects and medical data in the integrated interface. In addition, the Qt library includes the support of the XML data structure. We implement the digital annotation and metadata system with the Qt API.

**Control Widgets.** Mindful of the needs of art conservators, we include customized widgets for various types of data in the software. We provide a light widget tool to assist in the qualitative visual inspection of 3D geometry (see Fig. 4(b)). Conservators are accustomed to visual inspection of objects by varying the direction of illumination, and many are also familiar with RTI imaging



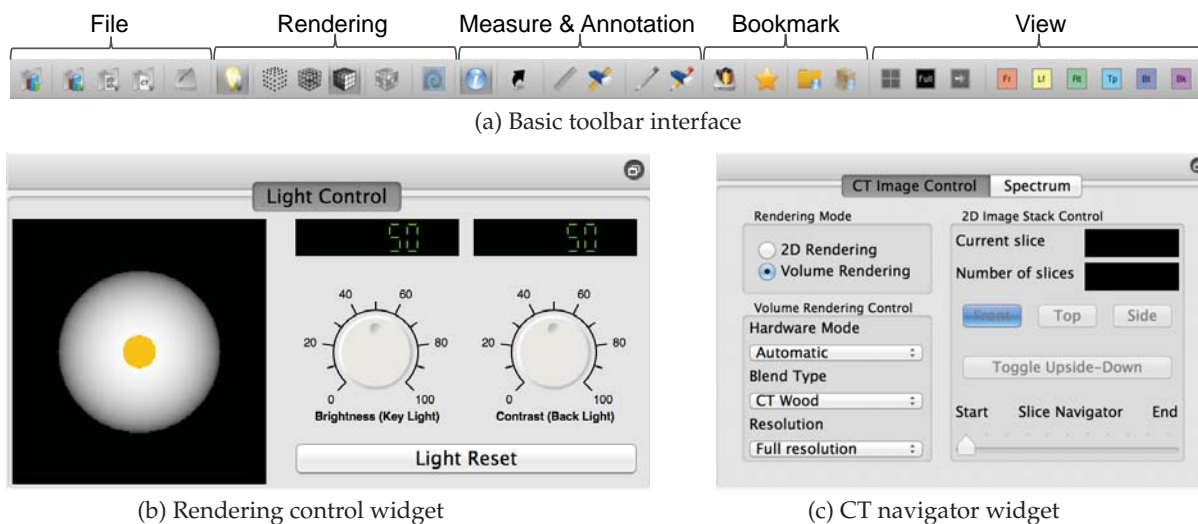
**Fig. 3:** Our rendering and user interaction pipeline using ITK/VTK/Qt.



to examine artifacts under a variety of lighting conditions. Our current software therefore includes a light control widget for controlling the direction vector of the virtual light source over 3D objects. A subtle point is the coordinate system of the light source specification. Various objects may have different local coordinate systems based on how they were acquired. The conservator positions the objects in the system windows, and the only coordinate system that is clear to them is the camera system. We therefore locate and control the light based on the camera's point of view. Following the control point of the trackball widget, the virtual light orbits the object in context. To do so, we obtain the orientation vector from the trackball interface and then determine the updated light position by applying the trackball transformation to the current viewing vector of the OpenGL camera. One consequence of this adjustment is that the conservator can treat multiple objects in adjacent windows as if they were on a virtual table. Adjusting the lighting control changes the light simultaneously in all windows, just as a physical light would affect all items on the same table.

The software also includes a CT data visualization widget. CT data consists of a stack of 2D X-ray images forming a 3D cube of rasterized images. Our current software provides two different types of visualization. One is to navigate the image stack in three different orientations (top, front and side) with or without interpolation. The other is to render it as a volume to help users understand the objects' internal structure (see Fig. 8(b)(1) and 1 for examples). We visualize the volumetric rendering of the CT data with ray casting techniques enabled using the VTK libraries [Schroeder et al. 2006]. Users are offered options for CPU or GPU rendering, image resolution, and blend type (e.g. maximum intensity projection (MIP), isosurface, etc.). See Fig. 4(c) for the CT widget.

**Iterative Testing and Refinement.** While we made design decisions at the start of software development, the interface layout and handling of data types were continuously refined with conservator input throughout development. Hyperspectral, volumetric and 3D surface scan data was acquired for objects actively being conserved to exercise the software in use. The iterative refinement resulted in modification in the original interface, and motivated the types of annotations needed for conservators to record observations.



**Fig. 4:** (a) The basic toolbar interface for convenient access to the software's functionalities. (b) Widgets for controlling the intensity of the virtual key/fill lights. (c) CT medical imaging data navigation options for stack and volume rendering.

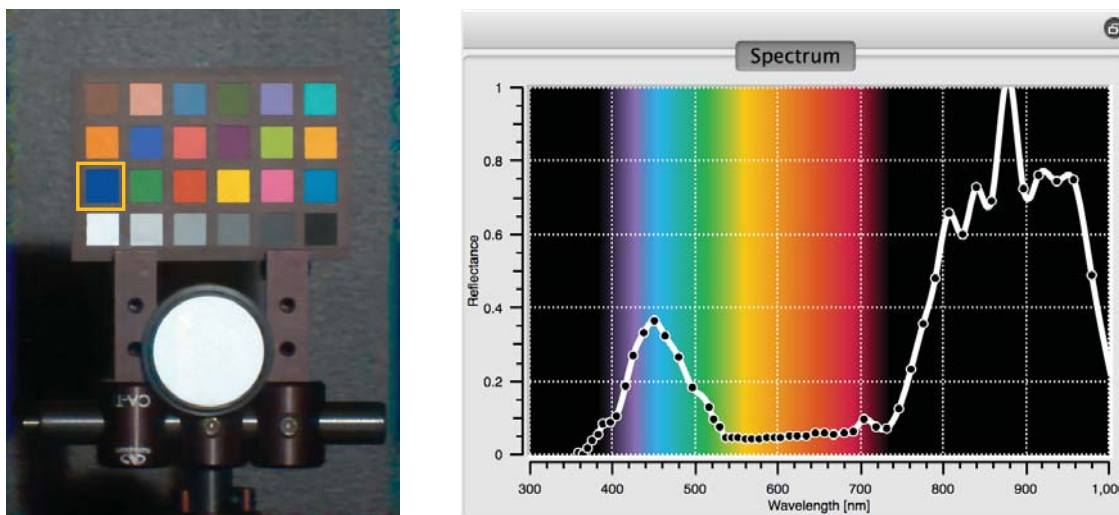
### 3.2 Hyperspectral Imaging Software

Among the major functionalities for art conservation software that existing software libraries do not accommodate are color conversion and visualization techniques to deal with hyperspectral and multispectral data. Hyperspectral imaging allows conservators to examine high-dimensional spectra qualitatively and quantitatively. Visual inspection of high-dimensional data is a powerful and successful tool for museum and gallery professionals. However, one problem with hyperspectral imaging is that spectral data is very dense and beyond human visual capabilities. One naive approach is to visualize the data in each individual spectral band as a grayscale image. This allows radiometric inspection, but it does not present the true color information needed by users to relate the data to their direct visual impression of the artifact.

We therefore include two different types of visualization of hyperspectral 2/3D data. We first map the high-dimensional spectral information to the visible range by converting the original data to the trichromatic constraints of human vision.

However, the spectral range of human vision is only a subset of the electromagnetic radiation that recent hyperspectral imagers can detect [Fischer and Kakoulli 2006]. We therefore provide another widget for viewing hyperspectral spectrum information simultaneously as the artifact is displayed in trichromatic color. Users can probe the color representation of a hyperspectral image or textured object and obtain detailed spectral measurements on a point-by-point basis.

Our current software reads hyperspectral information in multi-layers supported by the OpenEXR file format. The spectral data from a 2D image or 3D texture map is read using the OpenEXR API. Next, the spectral channels  $s$  are first projected to the tristimulus values by CIE 2-degree color matching functions  $M_{XYZ}$  [CIE 1986]. The tristimulus values in XYZ are then transformed by  $M_{sRGB}$  into sRGB color primary values  $c$  by [Nielsen and Stokes 1998], where the rendered image is white-balanced by the gray-world algorithm [Buchsbaum 1980]:  $c = M_{sRGB} \cdot M_{XYZ} \cdot s$ . The linear sRGB color  $c$  is then rendered via gamma correction [Poynton 2008] to display the signal depths. We then compute the histogram of the converted image



**Fig. 5:** The left image shows an example of color visualization of a hyperspectral image. The right image shows the software's spectral plot widget visualizing reflectance from the blue patch. The background of the plot widget is color-indexed. The black region to the left corresponds to invisible ultraviolet wavelengths (under 380 nm). The black region to the right shows the invisible infrared spectral reading. The colors in the middle represent the visible spectrum. Data courtesy of Kim et al. [2012].

and stretch the pixel levels between 1% and 99% to the range of display signals by effectively clamping the values.

When the software loads the hyperspectral 3D data, the data is also provided to the plot widget (see Fig. 5). For a 3D object, the screen coordinates of the mouse pointer are converted to world coordinates. After the coordinates in the texture map are determined from the corresponding world coordinates on the 3D object, the software visualizes the appropriate spectral information on the plot widget. This allows users to inspect not only the color at the point of interest but also the corresponding spectrum quantitatively.

### 3.3 Annotations and Metadata

Although the software can be used to visualize and browse separate objects in isolation, we also provide a number of related features that permit users to collect and maintain information about the objects.

Simplest among these is a screenshot tool that can capture the visualization of the current window and save it in one of three common graphics formats for raster images: JPEG, BMP, and PNG. The ubiquitous status of these file formats means that users can easily insert these images into other files, such as a word processing document or slide show, or display them in a Web browser to share with other users.

**Bookmarks.** The software contains a “bookmarking” feature that allows the user to document and restore the visualization of a single object at a particular point in time. The user may also create folders to organize and rearrange the bookmarks. All bookmarks contain information about the lighting and camera position and zoom in order to duplicate the user’s previous viewing experience and some additional elements corresponding to the specific filetype of the object. For example, a bookmark made of a 3D mesh might indicate that the object was visualized with a wireframe rendering, but would not contain notes about the current blend type or resolution, as this would be semantically meaningless information when not displaying CT scan data.

Bookmarks are file-specific and can be accessed at any time that the image is being visualized in the active window; changes made to a file’s bookmarks while working in one project will be apparent when working with the same file in a separate project or when viewing the image individually. We implement this functionality in a simple tree hierarchy widget that contains only those bookmarks belonging to the image currently being visualized (see Fig. 6).

**Metadata.** We provide a metadata system in our software that permits users to record structured information about the current image or project. Our metadata schema is based on the Getty Research Institute’s Categories for the Description of Works of Art, a popular standard for recording commentary about works of art and visual culture [Baca and Harpring 2009]. We selected the CDWA as the foundation for our metadata system largely because of its appeal to art conservators, with categories describing an object’s history of conservation and treatment, history of ownership and collecting, history of exhibitions and loans, and other salient information. In addition, the CDWA has a well-defined schema crosswalk that maps its categories to other common metadata standards, including the VRA Core, LIDO (Lightweight Information Describing Objects), and the Dublin Core. Because our software has seen interest from zoologists and other scientists, we also include several metadata categories from the Canadian Heritage Information Network’s Natural Sciences Data Dictionary to catalog objects of natural history, including notes about the object’s taxonomy, physical appearance, and chemical composition [CHIN 2013].

Although the CDWA defines several “core” elements that must be completed to constitute a valid metadata entry, we enforce no such circumscription in our software; instead, we allow users to add categories and elements as needed to describe the object, possibly including multiple instances of the same category or element. We are amenable to adding additional categories to the system in the future with sufficient user feedback; individual users may also wish to take advantage of our open-source framework by adding new categories themselves as necessary.

**Annotations.** We recognize that users may wish to record information not only within a structured meta-data system but also via free-form annotations about an image’s noteworthy or distinctive features. Annotations are represented as colored marks (seed widgets in the VTK library) representing a point on a 2D image or 3D model. Upon selecting the point to place the annotation, the user is prompted to enter the text that will be associated with that point. The annotation is then visualized on the object within the current render window. When the user interacts with an existing annotation, the corresponding text is displayed in the information widget, which allows users to edit and save the current contents of the text box.

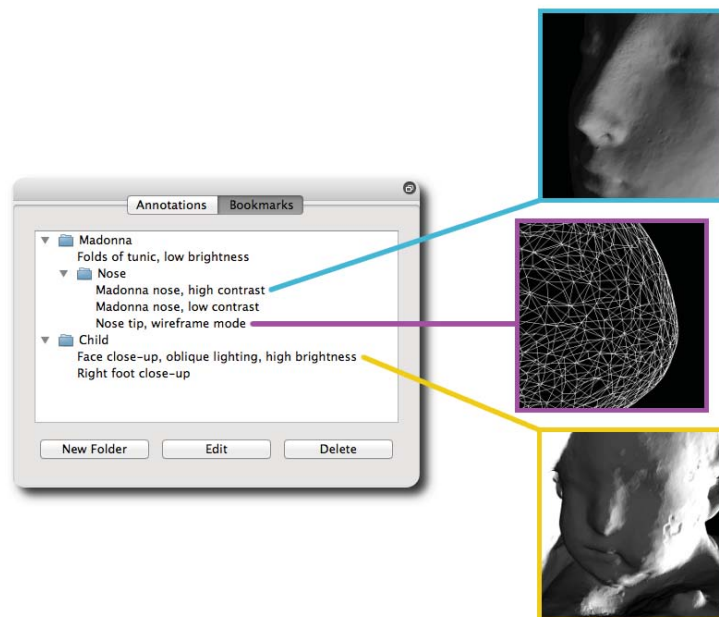
To store information about bookmarks, annotations, and metadata, we use the popular and powerful XML format. Because Qt contains multiple classes for XML file manipulation and I/O, parsing files and adding, editing, or deleting elements becomes a relatively simple task. Furthermore, the XML format and the associated Qt API are flexible enough to handle any supplementary elements that may be necessary to record as new software functionalities are added.

## 4. APPLICATIONS

### 4.1 Existing Functionality

We illustrate the use of the software in analysis by considering three objects from the Yale art museums.

The first object studied is the sculptural polychrome stucco relief *Virgin and Child*, created in the 15th century by Italian artist Vittorio Ghiberti. The object is 89 cm high and 61 cm wide, made of stucco plaster. It is believed that several copies of the sculpture were made from an original mold and now reside in collections in the U.S. and Europe. The structural support of the *Virgin and Child* is in a weakened state due to its age and the nature of the medium. The painted pigment surface layer has also deteriorated over

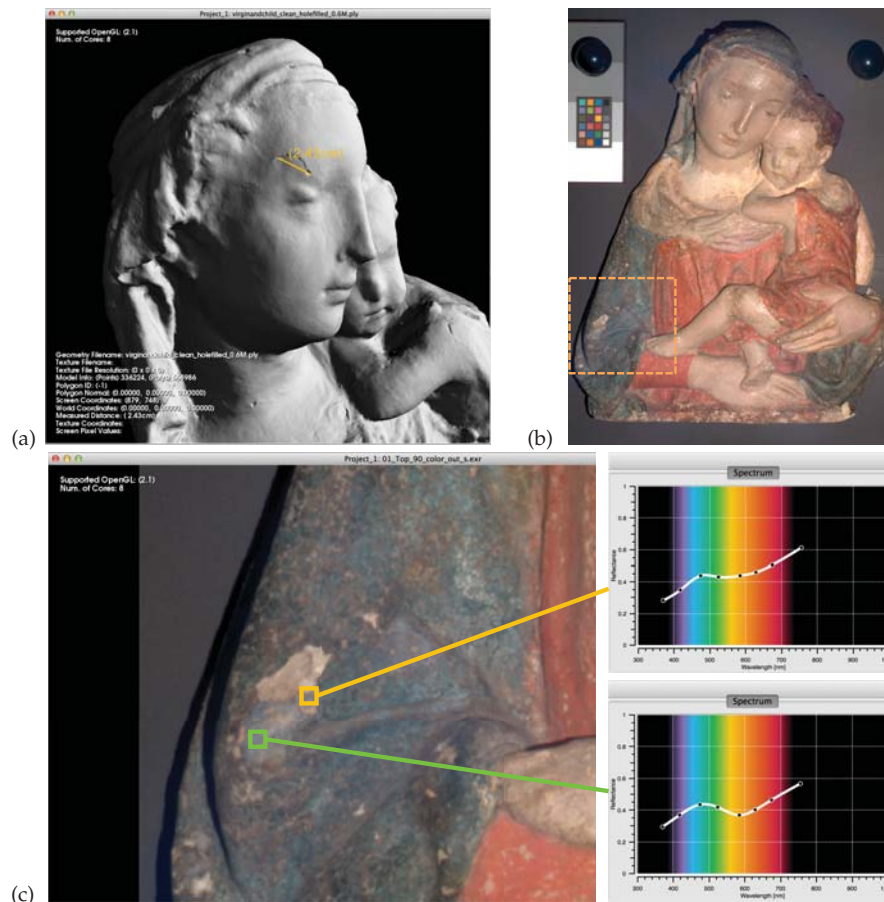


**Fig. 6:** Our software represents bookmarks as items in a tree hierarchy widget. Double-clicking on an item will restore the camera perspective, lighting, and other features associated with the given bookmark. The user may drag and drop items to reorganize them, rename a bookmark or folder, or delete the currently selected item.

the years due to wear and age; in addition, over the years the object has undergone several treatments of restoration or alteration which have altered the original appearance of the relief.

In these given conditions, a museum conservator, starting with various types of non-destructive visual inspection techniques, has undertaken a new conservation treatment. The imaging data types available for this object were hyperspectral imaging data of the ultraviolet, visible, and infrared spectra and 3D scanning data as well as more traditional digital image file types. 3D scanning was performed in order to help examine the dimensions and patterns of cracks on the plaster surface for future comparison against copies of the artwork. See Fig. 7(a) for a screenshot using the geometry inspection tool. Hyperspectral imaging was conducted for non-destructive investigation of pigment deterioration over the 3D polychrome surface. Fig. 7(b) shows a spectral comparison of two different areas on the sculpture in our software.

The second object we consider is *The Coronation of the Virgin*, a panel painting gilt and painted with egg tempera, by Jacopo del Casentino, ca. 1325. The panel is 31 cm high, 24 cm wide, and 2.5 cm thick. The panel support is of poplar and contains sculptured gesso sections that are gilded and painted with tempera.



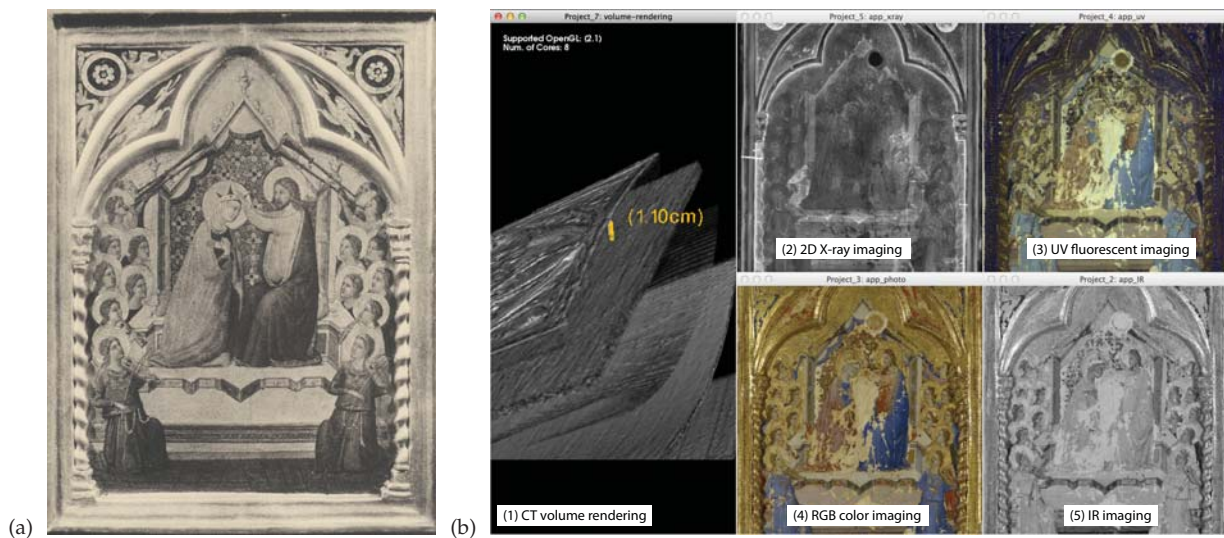
**Fig. 7:** Polychrome stucco relief, *Virgin and Child*, Italy 15th century, by Vittorio Ghiberti. (a) Measuring the length of the plaster crack in the relief. (b) A hyperspectral image of the object. The rectangle with dotted lines indicates the region shown close-up in the next image. (c) Reflectance spectra of pigments on the hyperspectral image (b) are depicted on the spectral plot widget. The two plots represent data from two different points of interest (the green and yellow squares).

Like the relief, this panel painting has been restored many times over the years. As shown in Fig. 8(a), the decorations on the top section of the panel and the crack on the top-middle part were restored in the 1930s. These decorations are visually distinct from the state of the current panel, which underwent a restoration campaign in the 1970s.

In preparation for new treatment, a museum conservator has performed various types of digital imaging analysis including 2D X-ray imaging, UV fluorescent imaging, infrared imaging, CT scanning, and traditional digital imaging file types. Conservators have heretofore focused primarily on better understanding the support layers and construction of the panel. In particular, there are two circular voids in the upper corners of the panel whose original purpose is lost. This, coupled with an interest in the nails which were used to construct the panel, led to a desire for an inspection tool that could allow users to pinpoint the location of such items and study them further.

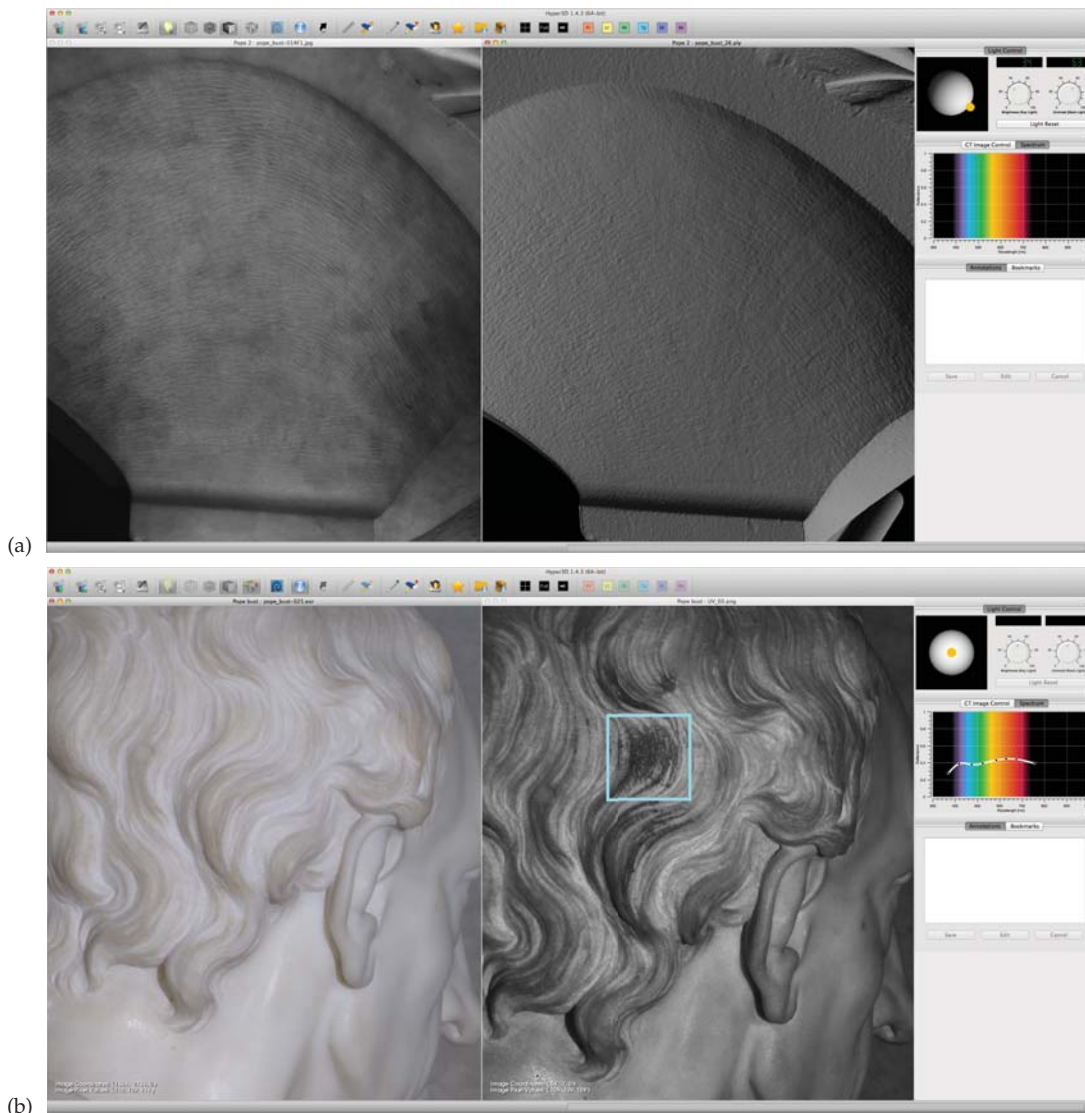
Our third object of study is *Alexander Pope*, a marble bust depicting the eponymous English poet, created by the French sculptor Louis François Roubiliac in 1741. The bust has been digitized as part of an ongoing project to compare the multiple versions of Roubiliac's marble busts of Pope sculpted between 1738 and 1760. We anticipate that the software will prove useful in comparing the relationship between these busts and their copies and adaptations.

A first step in the study is to assess the quality of the digital data being acquired. The sculpture has been captured via both 2D hyperspectral imaging and 3D laser scanning. Fig. 9(a) shows both types of images displayed within a single project. Comparing the appropriately lit 3D model with a gray scale photograph allow the user to assess whether details on the surface of the 3D model are due to measurement noise or to real features on the sculpture. While there is some noise apparent in the model where the photograph is smooth, the 3D model does appear to accurately represent small-scale tool marks on the lower back of the piece.



**Fig. 8:** *Tempera on panel with gilding, The Coronation of the Virgin, Italy ca. 1325, by Jacopo del Casentino.* (a) An archival photograph of the panel painting taken in the 1930s. (b) A screenshot of our software with various types of imaging data open, allowing users to compare hyperspectral 2D imaging and 3D CT volume rendering data simultaneously. Image (b) shows (1) CT volume rendering with geometric inspection (measuring the size of a pin inside the wooden panel), (2) 2D X-ray imaging, (3) UV fluorescent imaging, (4) high-resolution RGB color imaging, and (5) IR imaging.

Another step is to consider the surface condition of the piece. As is evident in Fig. 9(b), the bust contains variations in surface coloration that range from a milky white to a light tan color consistent with the accumulation of dust. Images from the ultraviolet channel reveal a discoloration that is not immediately apparent in the photographs taken under visible light.



**Fig. 9:** Marble bust, Alexander Pope, 1741, by Louis François Roubiliac. (a) View of a project with two images loaded. At left, a 2D image captured under ultraviolet light; at right, a 3D model. Arranging the model into roughly the same position as the image allows the user to compare the two representations and thereby assess the quality of the 3D scan. (b) The 2D hyperspectral image on the left combines information from 8 channels, while the 2D image on the right was captured under only ultraviolet light. Comparing the two images reveals a previously concealed variation in surface material indicated by the light blue box.

## 4.2 User Experience and Feedback

The museum staff have found that the various modes of volume rendering and isosurface rendering are beneficial in studying a work of art. The conservator is interested in geometric inspection using CT scan data, along with various high-end imaging data to better understand the support layers of the panel and carved relief, while comparing those data results with reflective surface data. With this software interface, museum staff can better compare the different types of imaging results from the same image coordinates by placing images side-by-side, allowing them to better compare data in one interface.

The program is particularly helpful with hyperspectral visual inspection. Repairs on an artifact are sometimes evident from small changes in color. The system helps the conservator examine and quantify such small color change, and correlate the change with geometric variations. We have found that users beyond conservators find the spectral visualization useful. There is no open source software that allows scholars to examine multispectral images of manuscript pages that allows multiple views of the same data. A paleographer at Yale examining recent multispectral scans of works of Chaucer found the system useful for comparing pigments on a page, and annotating the pages with observations.

We are working to expand the user base and obtain more feedback by presenting talks to museum professionals, including a recent discussion and demonstration at the Museum Computer Network 2013 conference. In general, the conservation community that uses digital imaging technologies has been enthusiastic about an integrated tool. It will take a substantial amount of time however to receive feedback on usage in day to day workflow.

## 5. CONCLUSIONS

We provide an open-source software tool with an intuitive and consistent interface that can handle various types of 2D and 3D imaging data, including hyperspectral images and CT scans, and permits users to record multiple forms of annotations and metadata, in accordance with the needs of art conservators.

Users have requested the ability to read more file types, such as RTI data sets, and the ability to print out files for sharing and archiving of treatments. We may also incorporate the ability to locally flatten 3D meshes, as Pal et al. [2013] do in their viewer for deformed historical documents, into the software. For future development, we would also like to concentrate on more efficiently sharing users' observations and notes. For example, one possible feature is the ability to export the metadata created within our software to other metadata formats, such as LIDO, CIDOC-CRM, and Dublin Core, to use in other applications. Another idea is an import/export project functionality that would encapsulate the files of a project and their associated bookmarks, metadata, and annotations for users to easily share with one another. Finally, creating hyperlinks between the bookmarks, metadata, and annotations within a single project and across projects – e.g., a metadata entry referring to a perspective captured by a particular bookmark, or to the text of an annotation – is another feature that we would like to add in the future.

The Hyper3D software code is available under the GNU General Public License v3.0 via SourceForge at <http://sourceforge.net/projects/hyper3d/> [Hyper3D 2013]. We invite all readers to get involved in the project by developing various improvements and extensions to the software.

## REFERENCES

- M.D. Abràmoff, P.J. Magalhães, and S.J. Ram. 2004. Image processing with ImageJ. *Biophotonics international* 11, 7 (2004), 36–42.
- L. Arbace, E. Sonnino, M. Callieri, M. Dellepiane, M. Fabbri, A. Iaccarino Idelson, and R. Scopigno. 2012. Innovative uses of 3D digital technologies to assist the restoration of a fragmented terracotta statue. *J. Cultural Heritage* (2012).
- Michael Attas, Edward Cloutis, Catherine Collins, Douglas Goltz, Claudine Majzels, James R. Mansfield, and Henry H. Mantsch. 2003. Near-infrared spectroscopic imaging in art conservation: investigation of drawing constituents. *Journal of Cultural Heritage* 4, 2 (2003), 127 – 136. DOI : [http://dx.doi.org/10.1016/S1296-2074\(03\)00024-4](http://dx.doi.org/10.1016/S1296-2074(03)00024-4)
- Murtha Baca and Patricia Harpring (Eds.). 2000. Revised in 2009. *Categories for the Description of Works of Art (CDWA)*. J. Paul Getty Trust and the College Art Association, Los Angeles.



- Fausto Bernardini, Ioana M. Martin, and Holly E. Rushmeier. 2001. High-Quality Texture Reconstruction from Multiple Scans. *IEEE Trans. Vis. Comput. Graph* 7, 4 (2001), 318–332. <http://doi.ieeecomputersociety.org/10.1109/2945.965346>
- G. Buchsbaum. 1980. A Spatial Processor Model for Object Colour Perception. *J. the Franklin Institute* 310, 1 (1980), 1–26.
- Steven P. Callahan, Juliana Freire, Emanuele Santos, Carlos E. Scheidegger, Cláudio T. Silva, and Huy T. Vo. 2006. VisTrails: visualization meets data management. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data (SIGMOD '06)*. ACM, New York, NY, USA, 745–747. DOI : <http://dx.doi.org/10.1145/1142473.1142574>
- M. Callieri, F. Ponchio, P. Cignoni, and R. Scopigno. 2008. Virtual Inspector: a flexible visualizer for dense 3D scanned models. *Computer Graphics and Applications, IEEE* 28, 1 (2008), 44–54.
- CHIN. 2013. CHIN Data Dictionary - Natural Sciences. (2013). [http://www.pro.rcip-chin.gc.ca/bd-dl/ddrcip-sn-chindd\\_ns-eng.jsp](http://www.pro.rcip-chin.gc.ca/bd-dl/ddrcip-sn-chindd_ns-eng.jsp)
- CIE. 1986. *Colorimetry* (2nd ed.). CIE Pub. 15.2. Commission Internationale de l'Éclairage (CIE), Vienna.
- Paolo Cignoni, Marco Callieri, Massimiliano Corsini, Matteo Dellepiane, Fabio Ganovelli, and Guido Ranzuglia. 2008. MeshLab: an Open-Source Mesh Processing System. In *Eurographics Italian Chapter Conference*. Eurographics, 129–136.
- Matthias Kalle Dalheimer. 2002. *Programming with Qt: Writing Portable GUI applications on Unix and Win32* (second ed.). O'Reilly & Associates, Inc., pub-ORA:adr. xviii + 499 pages. <http://www.oreilly.com/catalog/prowqt2>
- Paul E. Debevec and Jitendra Malik. 1997. Recovering High Dynamic Range Radiance Maps from Photographs. In *Proc. the ACM SIGGRAPH Conference*. 369–378.
- J.K. Delaney, C. Metzger, E. Walmsley, and C. Fletcher. 1993. "Examination of the visibility of underdrawing lines as a function of wavelength". In *Proceedings of the 10th Triennial ICOM-CC Meeting*, Vol. 1. 15–19.
- Lily Díaz, Markku Reunanen, Blanca Acuña, and Atte Timonen. 2011. ImaNote: A web-based multi-user image map viewing and annotation tool. *J. Comput. Cult. Herit.* 3, 4, Article 13 (April 2011), 11 pages. DOI : <http://dx.doi.org/10.1145/1957825.1957826>
- Julie Dorsey, Songhua Xu, Gabe Smedresman, Holly Rushmeier, and Leonard McMillan. 2007. The Mental Canvas: A Tool for Conceptual Architectural Design and Analysis. In *Proceedings of the 15th Pacific Conference on Computer Graphics and Applications (PG '07)*. IEEE Computer Society, Washington, DC, USA, 201–210. DOI : <http://dx.doi.org/10.1109/PG.2007.62>
- C Fischer and I Kakouli. 2006. Multispectral and hyperspectral imaging technologies in conservation: current research and potential applications. *Reviews in conservation* (2006).
- Marc Ghysels. 2005. CT Scans in Art Work Appraisal. Online journal asianart.com. (2005). <http://www.asianart.com/articles/ghysels/index.html>
- Sven Havemann, Volker Settgast, René Berndt, Oyvind Eide, and Dieter W. Fellner. 2009. The Arrigo Showcase reloaded towards a sustainable link between 3D and semantics. *J. Comput. Cult. Herit.* 2, 1, Article 4 (July 2009), 13 pages. DOI : <http://dx.doi.org/10.1145/1551676.1551680>
- Hyper3D. 2013. An open-source project in SourceForge.net. <http://sourceforge.net/projects/hyper3d/>. (2013). <http://sourceforge.net/projects/hyper3d/>
- R. Kaehr, L. Perrois, and M. Ghysels. 2007. A Masterwork that Sheds Tears and Light: A Complementary Study of a Fang Ancestral Head. *African Arts* 40, 4 (2007), 44–57.
- Min H. Kim, Todd Alan Harvey, David S. Kittle, Holly Rushmeier, Julie Dorsey, Richard O. Prum, and David J. Brady. 2012. 3D Imaging Spectroscopy for Measuring Hyperspectral Patterns on Solid Objects. *ACM Transactions on Graphics (Proc. SIGGRAPH 2012)* 31, 4 (2012), 38:1–11. DOI : <http://dx.doi.org/10.1145/2185520.2185534>
- K. Kylander and O.S. Kylander. 1999. *Gimp*. Coriolis Group.
- Bruno Lévy. 2008. <http://alice.loria.fr/index.php/software.html>. (2008). <http://alice.loria.fr/index.php/software.html>
- Lucas Digital Ltd. 2009. OpenEXR. <http://www.openexr.com/>. (2009). <http://www.openexr.com/>
- James R. Mansfield, Michael Attas, Claudine Majzels, Edward Cloutis, Cathy Collins, and Henry H. Mantsch. 2002. Near infrared spectroscopic reflectance imaging: a new tool in art conservation. *Vibrational Spectroscopy* 28, 1 (2002), 59 – 66. DOI : [http://dx.doi.org/10.1016/S0924-2031\(01\)00145-X](http://dx.doi.org/10.1016/S0924-2031(01)00145-X)
- Peter Mindek, Stefan Bruckner, and M Eduard Gröller. 2013. Contextual Snapshots: Enriched Visualization with Interactive Spatial Annotations. In *Proceedings of the 2013 Spring Conference on Computer Graphics*. 59–66.
- Jan Möbius and Leif Kobbelt. 2010. OpenFlipper: An Open Source Geometry Processing and Rendering Framework. In *Curves and Surfaces (Lecture Notes in Computer Science)*, Vol. 6920. Springer, 488–500. <http://dx.doi.org/10.1007/978-3-642-27413-8>
- J.P. Mohen, M. Menu, B. Mottin, L. Nash, and G.R. Edwards. 2006. *Mona Lisa: inside the painting*. Abrams.
- Mark Mudge, Tom Malzbender, Alan Chalmers, Roberto Scopigno, James Davis, Oliver Wang, Prabath Gunawardane, Michael Ashley, Martin Doerr, Alberto Proenca, and Joao Barbosa. 2008. Image-Based Empirical Information Acquisition, Scientific Reliability, and Long-Term Digital Preservation for the Natural Sciences and Cultural Heritage , Maria Roussou and Jason Leigh (Eds.). Eurographics Association, Crete, Greece, -. <http://www.eg.org/EG/DL/conf/EG2008/tutorials/T2.pdf>

- Derek R. Ney, Elliot K. Fishman, Donna Magid, and Robert A. Drebin. 1990. Volumetric Rendering of Computed Tomography Data: Principles and Techniques. *IEEE Computer Graphics and Applications* 10, 2 (March 1990), 24–32.
- Mary Nielsen and Michael Stokes. 1998. The Creation of the sRGB ICC Profile. In *Proc. Color Imaging Conf. IS&T*, 253–257.
- Kazim Pal, Melissa Terras, and Tim Weyrich. 2013. Interactive Exploration and Flattening of Deformed Historical Documents. *To appear in Computer Graphics Forum (Proc. Eurographics)* 33, 2 (2013).
- Gianpaolo Palma, Massimiliano Corsini, Paolo Cignoni, Roberto Scopigno, and Mark Mudge. 2010. Dynamic shading enhancement for reflectance transformation imaging. *JOCCH* 3, 2 (2010), 6. <http://doi.acm.org/10.1145/1841317.1841321>
- C. Poynton. 2008. Frequently Asked Questions about Gamma. <http://www.poynton.com/PDFs/TIDV/Gamma.pdf>. (2008). <http://www.poynton.com/PDFs/TIDV/Gamma.pdf>
- A. Rosset, L. Spadola, and O. Ratib. 2004. OsiriX: an open-source software for navigating in multidimensional DICOM images. *Journal of Digital Imaging* 17, 3 (2004), 205–216.
- A. Roy, M. Spring, B.H. Berrie, and E. Walmsley. 2007. *Raphael's painting technique: working practices before Rome: proceedings of the Eu-ARTECH workshop*. Nardini.
- Conrad Sanderson. 2010. *Armadillo: An Open Source C++ Linear Algebra Library for Fast Prototyping and Computationally Intensive Experiments*. Technical Report. NICTA, Australia.
- William J. Schroeder, Martin Kenneth, and William E. Lorensen. 2006. *The Visualization Toolkit* (4th ed.). Prentice Hall.
- Senate Johannes Taka and Subhadra Srinivasan. 2011. NIRViz: 3D Visualization Software for Multimodality Optical Imaging Using Visualization Toolkit (VTK) and Insight Segmentation Toolkit (ITK). *J. Digital Imaging* 24, 6 (2011), 1103–1111. <http://dx.doi.org/10.1007/s10278-011-9362-5>
- Benjamin M. W. Tsui. 1994. Single-Photon Emission Computed Tomography (Spect) Imaging. In *ICIP* (3). 1031–1035.
- VRA. 2013. VRA Core 4.0. (2013). <http://www.vraweb.org/projects/vracore4/index.html>

Received July 2013; revised December 2013; accepted December 2013