Visualizing and Analyzing the *Mona Lisa*

Louis Borgeat, Guy Godin, Philippe Massicotte, Guillaume Poirier, François Blais, and J.-Angelo Beraldin
National Research Council of Canada

Size and scale issues present a complexity problem in visualizing detailed 3D models built from sensor data. A model of Leonardo da Vinci's *Mona Lisa*, with its thin pictorial layer, illustrates the need for intuitive real-time processing tools that are seamlessly integrated with a multiresolution visualization environment.

As technologies for acquiring 3D data and algorithms for constructing integrated models evolve, very large data sets representing objects or environments are emerging in various application areas. As a result, significant research in computer graphics has aimed to interactively render such models on affordable commodity computers. Interest is growing in the possibility of integrating real-time analysis and transformation tools in interactive visualization environments as they become more available.

Large meshes built from measurements are complex not only because of their typically huge size but also because of their multiple scales of information. High-resolution and high-precision 3D scanning provides information not only about overall shape and color, but also about fine surface details and variations that must be accessed within a larger context and can be hidden within global surface features. This problem differs slightly from that of large models composed of complex collections of abstract geometric entities, such as CAD models. But a common task remains: to efficiently access information at a variety of scales, and to present it at different interest levels, without creating visual artifacts or potential misinterpretations. Here, accessing the data means not only displaying it photorealistically, but also transforming it to enhance the viewer's understanding of its contents, and therefore maximize the data set's value (see the "Previous Work in 3D Modeling for Cultural Heritage" sidebar).

This article presents the set of interactive graphic tools that we used to assist experts in analyzing a detailed complete 3D scan of Leonardo Da Vinci's *Mona Lisa* (see the "Scanning the *Mona Lisa*" sidebar). The techniques we used in the application project aimed to both help document this famous artwork and better understand Leonardo's painting technique. Our tools let us interactively explore the data on commodity hardware using various representations and at different scales.

Multiresolution modeling

Figures 1a and 1b show NRC's high-resolution polychromatic laser scanner we used to digitize the *Mona Lisa*. We assembled the *Mona Lisa* laser scans into a 333-million polygon color-per-vertex model (see Figure 1c). The model represents the entire poplar panel on which the painting was executed, with higher resolution on the obverse (front) surface. Such a model is still too large for even the most recent high-end graphics hardware, and this is in no way an exceptional situation. By combining sensors and digital photography, one can produce models of several hundred millions of 3D samples and hundreds of gigabytes of 2D images in a matter of days, if not hours.

Our general technique for interactively displaying such large scanned surface data sets easily handles models composed of hundreds of millions of polygons and tens of gigabytes of associated texture data.[1] The method is an extension of view-dependent hierarchical levels of detail (LODs), where we use geomorphing to ensure temporal and spatial continuity between the different LODs and across the entire model.

Our rendering technique combines several important advantages for this application context.

First, by using static preoptimized geometry units as the primitives for the view-dependent computations, we strike a better balance between GPU and CPU usage. Thus, we can benefit from on-GPU geometry and texture caching and from better use of the GPU vertex cache. On a model such as the *Mona Lisa*, we can render more than 100 million geomorphed polygons per second when computing per-vertex lighting. This performance level lets us render the multiresolution models with subpixel accuracy at interactive frame rates.

We also minimize the visual artifacts associated with the view-dependent transformation of the displayed data through geomorphing. One of the known inconveniences of LOD-based approaches is the visible transitional artifacts associated with changes in the resolution level. When rendering a model such as the *Mona Lisa* at high resolution, no artifacts are visible even if we set a target resolution slightly over one pixel.

In addition, our technique requires little CPU usage because we perform all geomorphing on the GPU. We use

 Published by the IEEE Computer Society 0272-1716/07/$25.00 © 2007 IEEE

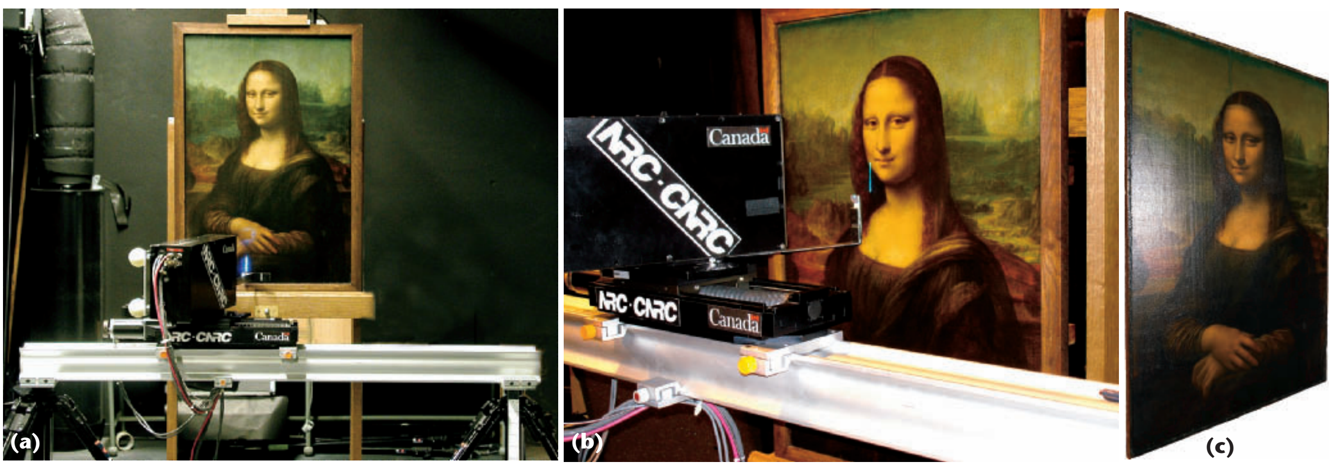
Previous Work in 3D Modeling for Cultural Heritage

The cultural heritage field has provided a stimulating testing ground for many developments in 3D acquisition, modeling, and display. Elsewhere, we review surface sensor-based data-set-rendering techniques of large models.[1] Large-scale cultural heritage documentation projects include the Digital Michelangelo Project,[2] and the scanning of the *Florentine Pietà*[3] and of the Great Buddha of Kamakura.[4] A description of other projects in cultural heritage and archaeological applications is available elsewhere.[5] Some work has also aimed to enhance the rendering of 3D models for analysis purposes. Anderson and Levoy process and enhance models of cuneiform tablets using curvature and accessibility coloring.[6] Cohen et al. use interactive 2D+ visualization combined with nonphotorealistic lighting to also analyze such tablets.[7] Rusinkiewicz et al. describe a nonphotorealistic rendering technique to highlight shape details of 3D models that's derived from hand-drawing shading techniques.[8]

When sufficient resolution and precision become available, paintings can no longer be considered flat 2D objects. Rather, they enter the realm of 3D imaging applications and become massive data sets comparable to those obtained on sculptures. Our work on modeling the complete *Mona Lisa* from laser scans is one such recent project (see the "Scanning the *Mona Lisa*" sidebar on the next page).

References

1. L. Borgeat et al., "Gold: Interactive Display of Huge Colored and Textured Models," *Proc. Siggraph*, ACM Press, 2005, pp. 869-877.
2. M. Levoy et al., "The Digital Michelangelo Project: 3D Scanning of Large Statues," *Proc. Siggraph*, ACM Press, 2000, pp. 131-144.
3. F. Bernardini et al., "Building a Digital Model of Michelangelo's *Florentine Pietà*," *IEEE Computer Graphics and Applications*, vol. 22, no. 1, 2002, pp. 59-67.
4. D. Miyazaki et al., "The Great Buddha Project: Modeling Cultural Heritage through Observation," *Proc. 6th Int'l Conf. Virtual Systems and Multi-Media* (VSMM), Int'l Soc. on Virtual Systems and Multimedia, 2000, pp. 138-145.
5. G. Godin et al., "Active Optical 3D Imaging for Heritage Applications," *IEEE Computer Graphics and Applications*, vol. 22, no. 5, 2002, pp. 24-36.
6. S.E. Anderson and M. Levoy, "Unwrapping and Visualizing Cuneiform Tablets," *IEEE Computer Graphics and Applications*, vol. 22, no. 6, 2002, pp. 82-88.
7. J. Cohen et al., "iClay: Digitizing Cuneiform," *Proc. Symp. Virtual Reality, Archaeology, and Cultural Heritage* (VAST), A.K. Peters, 2004, pp. 135-143.
8. S. Rusinkiewicz, M. Burns, and D. DeCarlo, "Exaggerated Shading for Depicting Shape and Detail," *ACM Trans. Graphics* (Proc. Siggraph), vol. 25, no. 3, pp. 1199-1205.



1 Creating a model of the *Mona Lisa*: (a and b) We used our high-resolution polychromatic laser scanner to digitize the painting. (c) We can then render the model using the 3D and color data.

the CPU only to cull the coarse hierarchical LOD structure, as with any scene graph, and to prefetch data during navigation. Resources are therefore available for other tasks such as audio/video transcoding for telecollaboration (see the "Displaying the Imagery" sidebar on page 63).

Finally, we display the original full-resolution model at the finest level of detail. This ensures that the user can differentiate between real features present in the data and potential artifacts that could be caused by the processing and rendering steps associated with managing a large data set.

Before we can interactively display a model with this tool, we must precompute the appropriate multiresolution data structure. For a model the size of the *Mona Lisa*, this process will take a few hours.[1] Preprocessing converts the triangular mesh model into a compressed LOD hierarchy of optimized geometry patches and associated texture or attribute data. The first step of this preprocessing is to simplify the entire model into a sequence of discrete LODs using an algorithm based on vertex aggregation. This choice of simplification technique is important because the aggregation paths become the geomorphing paths in the real-time rendering phase. We then decompose the lowest resolution LOD into a set of triangle patches. We partition the next higher resolution level along

Scanning the *Mona Lisa*

At the request of the Louvre's Paintings Department, the French Center for Research and Restoration for Museums (C2RMF) undertook the largest scientific examination ever conducted on the *Mona Lisa*. This study coincided with the painting's move to the newly renovated Salle des États. A team of 39 specialists with backgrounds in art history, conservation, photography, physics, engineering, chemistry, optics, and digital imaging from seven institutions took part in various aspects of the study.

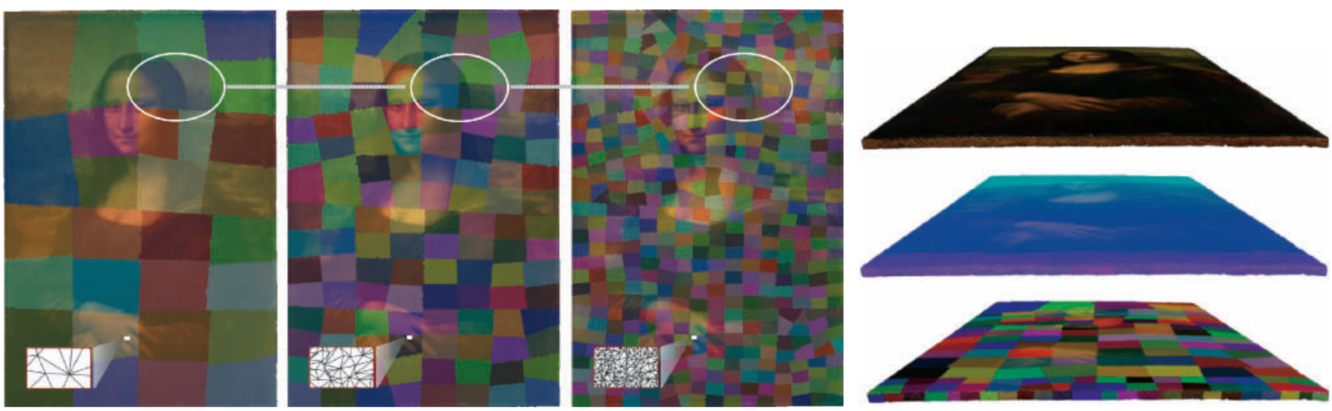
As part of this project, a team of 3D imaging scientists from the National Research Council of Canada (NRC) was invited to Paris in October 2004 to undertake the 3D scanning of Leonardo's most famous painting. The objective was to acquire a high-resolution 3D image data set of the obverse, reverse, and sides of the *Mona Lisa* to build a model of the complete painting. Leonardo painted the *Mona Lisa* on a poplar panel measuring 79.4 × 53.5 cm. The resulting 3D data and model provide a digital record of the painting's condition, and help in the computerized study of Leonardo's structure and technique.

The *Mona Lisa* presented a unique research and development challenge for 3D-imaging technologies, because of the flat pictorial layer and absence of apparent brush strokes, and the surface's translucency. Moreover, the team had access to the painting for a few hours over two nights, thus requiring careful planning of the scanning operations. Restorers performed all manipulations of the *Mona Lisa*, and the painting's environmental conditions were monitored at all times. Preliminary results are published elsewhere.[1,2]

The complete model of the painting contains 333 million triangles, with an average sampling density of 60 μm on the obverse. The NRC's scanning system measures the surface geometry by triangulating a projected low-power laser spot. The laser used is a mixture of wavelengths, which are decomposed at observation to estimate the surface color in perfect registration with the geometric measurements. This measurement process leads naturally to a color-per-vertex representation. We achieved complete surface coverage by multiple overlapping scans that are integrated into a unified mesh. Researchers can then examine the 3D and color model using the techniques we've described, either in a realistic manner or in transformed representations that enhance features of interest.

References

1. J.-P. Mohen, M. Menu, and B. Mottin, eds., *Mona Lisa, Inside the Painting*, Abrams, 2006.
2. F. Blais et al., "More than a Poplar Plank: The Shape and Subtle Colors of the Masterpiece Mona Lisa by Leonardo," *Proc. SPIE*, vol. 6491, SPIE, 2007.



(a) (b)

2 Modeling processes. (a) The recursive group subdivision process across the level-of-detail (LOD) sequence. (b) Three representations of the same view-dependent rendering of the model. The top image shows a simple color rendering. In the middle image, each LOD is associated with a unique color. We color groups by blending the colors of the two levels between which they're morphing, according to their current geomorphing ratio. The bottom image shows the connected assembly of selected patches spanning three LODs.

the same boundaries, and subpartition each group in this level until we reach the desired granularity. We apply this process sequentially to all levels of the model, which gives us a hierarchy of group subdivisions spanning the whole sequence of LODs (see Figure 2a).

We can shape groups according to criteria such as compactness, common orientation, texture/viewpoint association, culling requirements, model structure, existing partitions, and number of primitives per group. These criteria change depending on the model type to be displayed. For example, for a model with much more texture than geometry, we might create patches that segment images in texture space to minimize the size and number of texture units. For a color-per-vertex model such as the *Mona Lisa*, we might prefer to create more compact patches to minimize errors in the view-dependent computations. Finally, we individually convert groups into triangle strips optimized for the GPU vertex cache to maximize rendering speed. These groups or patches constitute the basic units for all the view-dependent computations.

The runtime culling process selects a front in the LOD patch hierarchy (see Figure 2b, bottom).The

Displaying the Imagery

High-resolution models yield high-resolution imagery, which should be appropriately displayed to be fully appreciated. When in a workstation configuration, we can leverage our data sets by using one or two relatively affordable digital 30-inch screens with a $2,560 \times 1,600$-pixel resolution. However, when working collaboratively on larger screens, we're quickly limited by commodity projectors' much lower resolution. Indeed, a $1,024 \times 768$ meeting-room projector is more expensive than our high-resolution digital display, but displays five times fewer pixels. Typical solutions to this problem involve using multiple projectors and image tiling to obtain more resolution. But such setups can become expensive and require large installations, especially in the context of stereoscopic visualization, which requires either doubling the number of projectors for passive techniques, or using time-multiplexed stereo projectors.

We've developed an alternative technique that combines stereoscopic visualization with a focus+context, or foveated approach (see Figure A). On a large display wall, we project a high-resolution, high-brightness stereo image within a larger, lower-resolution one. A pair of commodity PCs and projectors produces each image. Such a setup combines the advantages of accessing a high-resolution area of interest while keeping a larger context within sight. Additional techniques ensure that the inset's presence doesn't affect the stereoscopic perception along its boundary, and that aligning the projectors doesn't require special manual efforts.[1]

We're also developing an application that integrates our rendering and display tools with other components useful in processing and analyzing 3D data sets. We've already included modalities to support collaborative work on our display wall.[2] Figure B illustrates a collaborative session in which participants can share a virtual world based on a multiresolution model, and interact by inserting X3D avatars with video insets and audio links in the environment. Participants can annotate the model using 3D drawing and guide the viewpoint of other participants to collaborate. We designed the application with large data sets in mind, so we've implemented many features to optimize resource usage. For example, the GPU partially encodes and decodes video as part of the 3D rendering, and all the 3D annotations are done using information from the depth buffer so there's no need to compute ray intersection directly on data sets. We also update the scene graph using efficient differential encoding.

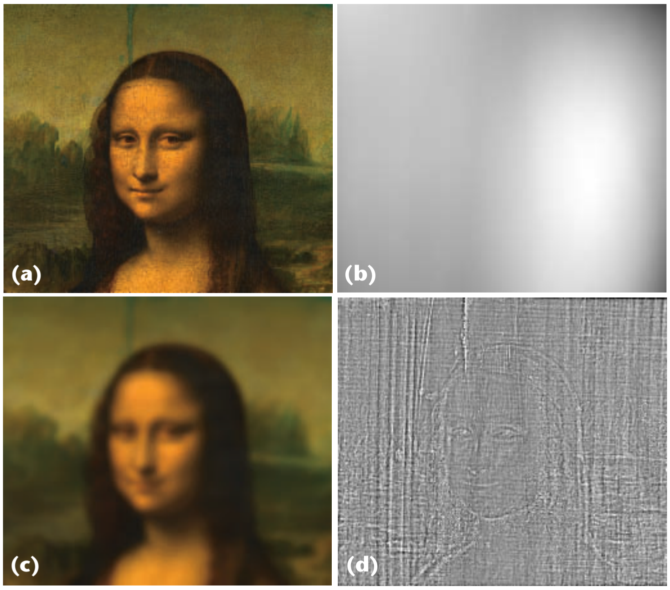
A **A foveated display used to collaboratively analyze high-resolution 3D models (shown here in monoscopic mode for clarity).**

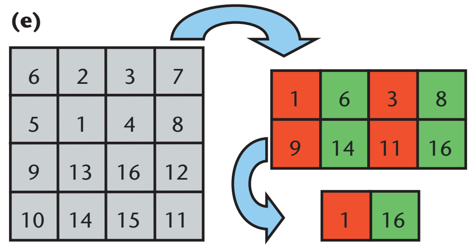
B **A collaborative environment for remote interaction with large 3D data sets.**

References

1. G. Godin, P. Massicotte, and L. Borgeat, "Foveated Stereoscopic Display for the Visualization of Detailed Virtual Environments," *Proc. 10th Eurographics Symp. Virtual Environments*, ACM Press, 2004, pp. 7-16.
2. L. Borgeat et al., "Collaborative Visualization and Interaction for Detailed Environment Models," *Proc. 10th Int'l Conf. Virtual Systems and Multimedia*, OCSL Press, 2004, pp. 1204-1213.

algorithm selects groups based on a chosen maximum screen pixel error and a unique worst-case precomputed error measurement associated with each group. We compute geomorphing ratios between selected groups and their lower resolution parent for each frame. Geomorphing boundary points between groups maintain seamless continuity between neighboring LOD groups at all times according to a specific set of rules. A vertex program on the GPU morphs the other points using a uniform ratio per individual patch, maintaining a more uniform on-screen resolution (see Figure 2b, middle). Because this program is on the GPU, this process requires almost no CPU resources. We apply geomorphing to spatial and texture coordinates, normals, and colors, and perform out-of-core prefetching asynchronously on a different thread to ensure smooth navigation and interaction. We assign groups for prefetching during culling by selecting another front in the LOD hierarchy, this time using a lower pixel-size error.





3 **The rendering process: (a) a rendering of the *Mona Lisa* model, (b) a rendering of the depth from the same viewpoint, (c) the same rendering of the color with a Gaussian filter applied, and (d) the result obtained after applying our detail highlight operator. (e) The reduction process involves aggregating neighboring pixels in successive rendering passes into smaller and smaller buffers. Minimum values are accumulated in the red channel, maximum values in the green channel.**

Different representations from a single viewpoint

Current GPUs provide much more control of the image-generation pipeline than their early counterparts. They're also significantly more powerful and can perform more real-time computations. We can harness this additional power to further process the rendered multiresolution model by implementing transformations useful as analytical tools. Such hardware will let us implement simple techniques such as rendering enhanced depth information instead of color to visualize other aspects of the data sets. We can also perform, in real time, complex multistep filtering and image-composition techniques that would require significant effort and manipulation in conventional 2D image-processing tools such as Adobe Photoshop. Recent GPUs also provide 32-bit float buffers as rendering targets, so we can execute high-precision computations and measurements directly on the GPU.

When rendering large data sets, implementing such multipass filters on the GPU doesn't necessarily result in a linear increase of the rendering time with the number of required passes. Passes in which we actually render the multiresolution mesh are costly because they use a large amount of graphics memory and involve processing millions of primitives. But we can often implement complex rendering stages by processing the mesh once and working iteratively on intermediate results. Those results are only pixel data buffers, not 3D models, located in fast graphics memory that we can rapidly transform and combine. We'll describe, as an example, a real-time filter that we implemented so we could separate small detail variations in an object's local shape at a given resolution from its global structure, acting as a high-pass filter for the data at the current LOD. In this representation of the *Mona Lisa*, we in effect separate the surface variations of the wood grain structure and paint layer from the more important warping of the poplar panel. Figure 3 illustrates some steps in this process.

First, we render the multiresolution model from the chosen viewpoint. This process is similar to that used for the photorealistic mode. We use the same interpolation process between LODs as when processing the vertices. However, in this case the target buffer is a single-channel 32-bit float buffer. At the end of the first pass, we obtain a 32-bit depth image of the 3D data in eye space. Figures 3a and 3b show the results of rendering both color and depth from the same viewpoint. We rescaled the depth in the figure across the entire grayscale spectrum for illustrative purposes: this is analogous to the technique used to produce Figure 4b and the depth component of Figure 4c.

The problem with such an image is that the human eye can only distinguish a limited number of shades of gray. Even if small shape details are present in this 32-bit high-precision image, they aren't visible because they're hidden in the panel's global warping, which consumes the dynamic range's most significant bits. Only the warping is apparent in the figure. We therefore need a way to highlight these details. The simplest solution is to virtually unwarp the poplar panel so other details can emerge.

We achieve this by first creating a representation of the overall shape in the form of a second float image that we obtained by convolving the depth image with a large Gaussian blur kernel. We can efficiently implement convolution on a GPU[2] by filtering sequentially with a unidimensional kernel along both image axes, resulting in fewer texture fetches. At the scale at which we apply the filter, we effectively filter out all the fine details we seek to recover, while preserving the panel's global warping. Because the details we want to filter out aren't visible on the gray image, we also apply the filter to the corresponding color image for illustrative purposes (see Figure 3c).

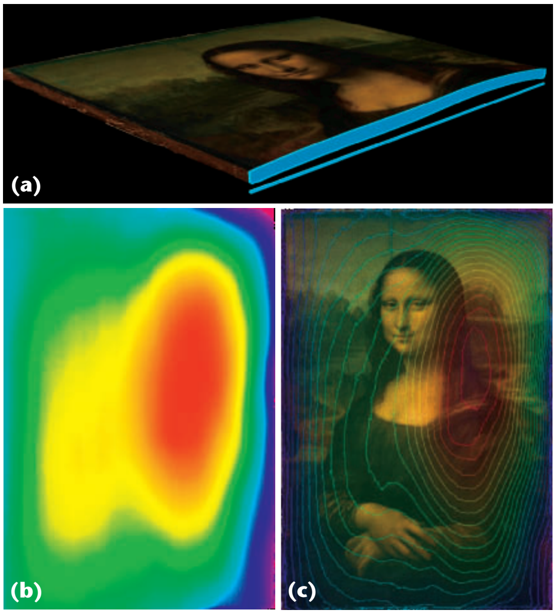
Next, we subtract the original image from the filtered one in a new rendering pass. This process results in an image containing only the local variations that were contained in the original buffer. We now simply need to transform those small variations into color values for display. Because the GPU must rescale values between zero and one to produce color values, we need to find the minimum and maximum values in our depth image. The fastest way to achieve this on a GPU is to hierarchically combine neighboring pixel values by rendering into smaller buffers until we're left with a one-by-one image containing our result. This technique is called *parallel reduction*.[2] We use two color channels to find the minimum and maximum at the same time. Figure 3e illustrates this reduction process. We can map the final result to a color or gray-scale spectrum (see Figure 3d). In practice, for a typical rendering at 2,560 × 1,600 pixels with subpixel geometry resolution, we observe a reduction in rendering speed by a factor between 2 and 3, even if we perform six full passes (plus the smaller ones for the reduction) instead of a single pass.

This example emphasizes the strong advantage we gain from working in our viewer's 3D multiresolution space instead of working with a 2D image-processing tool such as Photoshop or the GNU Image Manipulation Program. Indeed, to achieve the same image processing as described, we would have needed to first render the 3D model's depth into a 2D image, then crop it and choose appropriate filter sizes, create and combine layers, and apply histogram manipulations. In 3D, we specify the kernel filter's resolution in terms of screen pixels. Thus, by zooming into the model, we naturally crop the view and select the filter size in model units. We also ensure that the amount of data to be processed is constant without having to manipulate multiple stored intermediate representations, or to crop and explicitly work on multiple versions on the image in parallel. We can modify the viewpoint and instantly switch from one representation to the other while still navigating in 3D. Filter parameters are interactively modifiable within the navigation application. This greatly increases our ability to efficiently analyze the models. These advantages also apply to all filters used to produce the results in the following sections.

Many faces of the *Mona Lisa*

To people who approach heritage preservation from the computer graphics or remote-sensing world, removing an object's color to explore its shape using synthetic shading is a given, if not the default representation most often associated with laser scanning of objects and scenes. But for curators used to working with photographs or directly with the original artifacts, this simple representation often comes as a revelation. Indeed, to them, a mode in which there are no colors or ambiguous shadows can be of even greater interest than a photorealistic rendering. Raking light is a common technique for observing and analyzing a painting's surface shape. This technique is more often than not limited by ambiguities produced by cast shadows and color variations, or difficulties posed, for example, by global warping in the surface. Applying synthetic shading only on the model geometry conceptually corresponds to raking light without cast shadows (unless they're explicitly computed) on a uniformly colored surface, thus revealing surface details without being affected by actual surface color variations.

Using various representations, we illustrate some interesting technical aspects of Leonardo's famous painting. A more in-depth analysis and historical interpretations of sensor imagery of the *Mona Lisa* are available elsewhere.[3,4] All images in this section are snapshots from Atelier3D, our prototype interactive visualization and analysis tool.



4 Global shape of the poplar panel: (a) cut of the painting model with a virtual reference plane, (b) shape of the panel with color depth-coding after aligning the viewpoint using principal components analysis, and (c) contour lines overlaid on a depth and color blend.

The wood panel

One of the 3D scan's key goals was to precisely measure the shape of the poplar panel on which the *Mona Lisa* is painted. This would let conservators document the painting's current condition and help to assess its conservation state. Figure 4 shows different representations for visualizing the panel's global shape. Figures 4b and 4c are based on encodings of the model's depth relative to the observer. In Figure 4c, we blended the depth information with color data to help correlate the shape and appearance. To facilitate navigation, the Atelier3D interface provides tools to perform principal components analysis on the model's visible part to align the observer with the inspected data. This process is equivalent to fitting a plane within the depth buffer and realigning the viewing axis perpendicularly to it. This alignment tool is important for observing details because, as we stated earlier, we can only observe a limited number of color differences. The interface lets users adjust the histogram to obtain more detail in certain areas, similar to what's done in 2D tools. We implement the 3D cutting tool (see Figure 4a) at minimal cost on the GPU using either clipping planes or explicit geometry computations in the fragment program. Those dynamic cut tools are an intuitive way of understanding large-scale shape variations, especially when adding a reference frame to the scene. GPUs now support full floating-point buffers, so we can easily extract precise

5 **Local depth variations of the painting surface computed at a global scale. Wood grain and paint layer thickness var iations are clearly visible.**

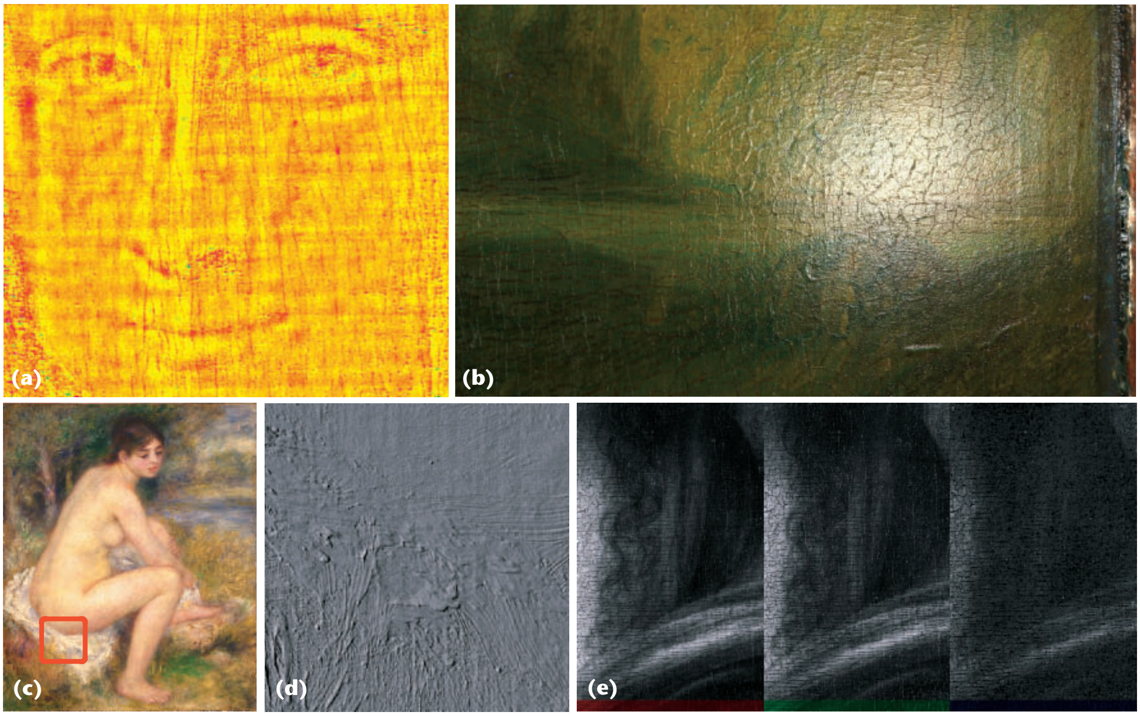
measurements or profiles in real time from the buffers used to render those images.

Figure 5 shows the result of applying the local variation filter. This image shows the panel's wood grain structure in a less ambiguous way than when using raking light techniques. Because we apply the filter at the pixel resolution chosen by navigating the 3D space, we can easily process the image in real time while navigating. The kernel size (in model units) adapts naturally as we zoom to areas of interest. Again, performing the equivalent task in a 2D tool from an initial full-size depth rendering would involve selecting and cropping areas, applying the filter to data that's at an excessive resolution, and waiting for the result. We'd then have to figure out numerical values for the filters, and so on, not to mention that many of the common image editing/processing software tools don't operate on 32-bit floating-point images.

The paint layer

The paint layer's most striking conservation aspect is the craquelure visible over the entire painting. Craquelure can indicate many phenomena of interest to conservators, such as panel deformation, improper painting technique, or normal paint aging. We can easily extract the 3D craquelure patterns using edge detection or other filters, or highlight the patterns using specularity to show both color and shape at the same time, as in Figure 6b.

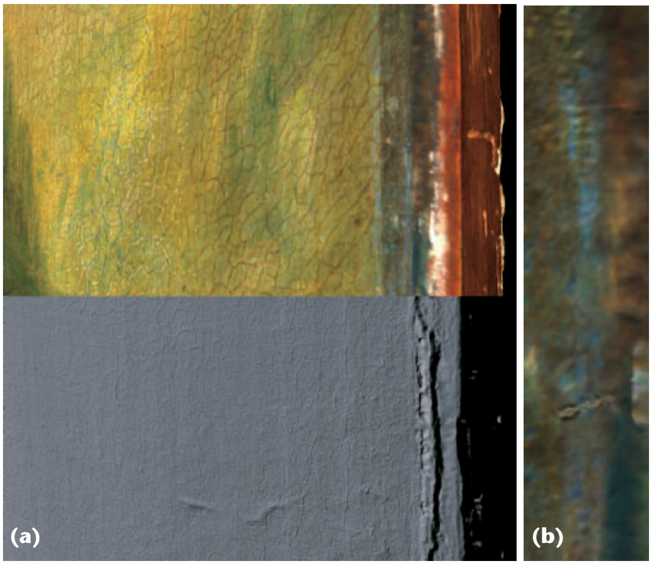
Art historians and curators have expressed interest in understanding Leonardo's technique in painting the *Mona Lisa*. Figure 6a is a local-variation depth rendering similar to Figure 5, but at a finer scale. The goal is to provide some indication concerning local variations in the paint layer thickness. We can clearly see that higher areas generally correspond to darker sections of the face's actual color. One theory is that Leonardo applied extremely thin successive layers of tinted glaze to produce those fine lighting and shading effects. This technique is called *sfumato*, which means "vanished" in Italian. The image tends to confirm this hypothesis. One visible result of Leonardo's subtle painting tech-



6 **Paint layer views. (a) Local elevation variations in the face area. (b) Using virtual specularity on the painting model is a good way to display information about the shape and color at the same time. The actual painting is also specular. (c and d) Detail of the pictural layer of a Renoir painting. Contrary to the *Mona Lisa*, brushstroke details are easily visible in the 3D data. (e) Separation of the color channels. Some details of the composition are enhanced at selected wavelengths. In this case, details of the hair are more visible in longer wavelengths.**

nique is that no brush strokes are apparent on the paint layer. To demonstrate the relevance of using laser scanning for painting surface analysis, we compare this surface with the work of another renowned painter, Pierre-Auguste Renoir. Figure 6d shows an artificially shaded area of Renoir's *Femme nue dans un paysage* (Nude woman in a landscape). We can compare this shading to the bottom part of Figure 7a, similarly produced from the *Mona Lisa* model.

The 3D laser-scanning system we designed and used simultaneously provides color measurement in registration with the geometric information. The reflectivity at different wavelengths and different scattering levels influenced these readings. We can gather interesting information by looking separately at the painting's three color channels. Figure 6e shows the three color channels for a single viewpoint. The red image gives a much better view of the fine hair details, which are much more difficult to see in the composite color version.



7 **Paint layer views: (a) paint layer border, and (b) magnified view of the bluish pigment. Paint accumulation is on the side, confirming that the painting was never cut. The blue area might be much closer to the original sky color.**

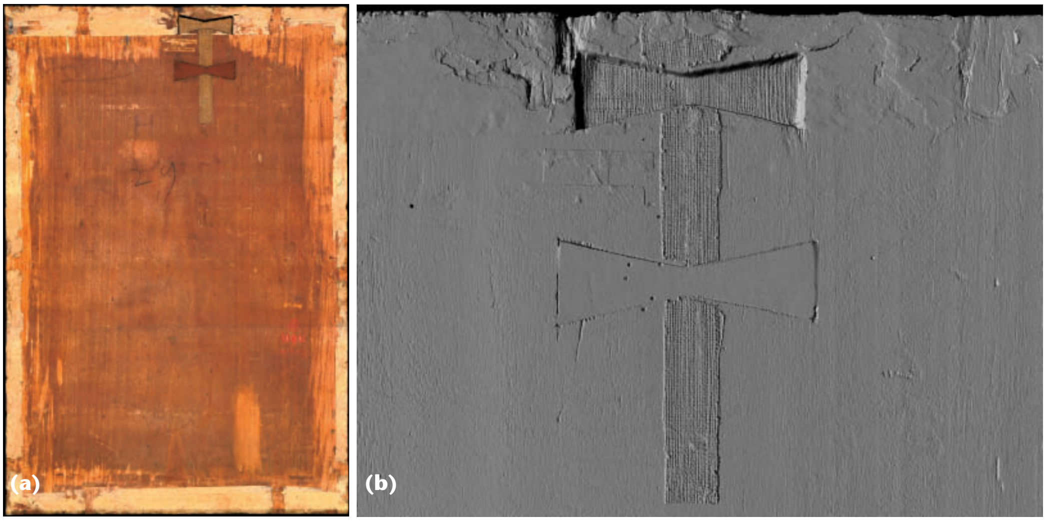
Traces of history

In addition to the craquelure pattern and the varnish's yellowing, time has left us with several interesting features to visualize on the *Mona Lisa*. The most obvious one is an 12-cm split on the top of the wood panel for which many restorations have been done over the years. Those are mostly visible on the back of the panel (see Figure 8). Conservationists have inserted butterfly-shaped wooden pieces transversally to stabilize the painting. Worm holes are visible on sides of the wood panel (Figures 7a and 7b). Many other smaller restorations are also apparent on the surface. To find these restorations, we could look in independent color channels. Even if the paint used has the same color as the painting under normal lighting, it might have been executed using pigments different from the original, and therefore exhibit a different spectral response. We can observe these differences when using different lighting—a phenomenon called *metamerism*. These color differences where enhanced much further using other imaging modalities as part of the overall documentation project.[3]

One enduring myth is that thieves cut the *Mona Lisa* wooden panel when they stole it in 1911. The 3D data in Figure 7a can easily debunk this myth, which probably originates from the existence of copies of the painting that included wider columns. We can see the accumulation of paint that formed along the original support frame. If the painting had been cut, that crest along the original frame's border would no longer be present. We can also see in that image a few blue specks, highlighted in Figure 7b. Those are believed to be pigments that were protected from aging by another frame, and could indicate the sky's original color, which turned from blue to green as the varnish yellowed and the painting became much darker.

Conclusion

3D imaging technologies play an increasingly important role in numerous application domains, such as manufacturing, architecture and infrastructure, and security. 3D sensors related to the one used for the *Mona Lisa* project are now used in space to inspect the shuttle



8 **Restoration of the 12-cm split in the poplar panel. (a) Rendering of the back of the painting. (b) Virtual raking light on a close-up of the split area, showing the consolidation work.**

Video Web Extra

View the author's video supplement to this article at http://doi.ieeecomputersociety.org/10.1109/MCG.2007.162

before re-entry. The need to visualize and interactively analyze very large data sets is expanding with such data sets' size and complexity. The software that we developed in the specific context of the *Mona Lisa* project provides us with a generic framework that can be applied to those other fields.

One key aspect of 3D data analysis not discussed in this article is the sequence of modeling steps that transforms raw sensor data into a complete and valid model ready to be processed for visualization and analysis. This modeling process is often long and tedious, and most commercial software currently fails to handle very large or overly complex models. Our future work will aim to provide tools that scale better with data size and complexity, and minimize the amount of human intervention required in the modeling phase. ■

Acknowledgments

We thank our colleagues Luc Cournoyer, Michel Picard, Louis-Guy Dicaire, Marc Rioux, John Taylor, and Daniel Gamache. We also thank Christian Lahanier and his colleagues at the French Center for Research and Restoration for Museums (C2RMF) in Paris, as well as the Paintings Department of the Louvre Museum.

References

1. L. Borgeat et al., "GolD: Interactive Display of Huge Colored and Textured Models," *ACM Trans. Graphics* (Proc. Siggraph) vol 24, no. 3, ACM Press, 2005, pp. 869-877.
2. R. Fernando, ed., *GPU Gems: Programming Techniques, Tips, and Tricks for Real-Time Graphics*, Addison-Wesley, 2004.
3. J.-P. Mohen, M. Menu, and B. Mottin, eds., *Mona Lisa, Inside the Painting*, Abrams, 2006.
4. F. Blais et al., "More than a Poplar Plank: The Shape and Subtle Colors of the Masterpiece Mona Lisa by Leonardo," *Proc. SPIE*, SPIE, vol. 6491, 2007.

Louis Borgeat *is an associate research officer with the Visual Information Technology Group at the National Research Council of Canada. His current research interests include modeling and visualization of large sensor data sets, telecollaboration, real-time computer graphics, and 3D computer vision. He is completing is PhD in electrical engineering and has an MS in computer engineering from Laval University. Contact him at louis.borgeat@nrc-cnrc.gc.ca.*

Guy Godin *is a senior research officer with the Visual Information Technology Group at the National Research Council of Canada. His research interests include 3D computer vision and image analysis, shape and appearance modeling, and interactive visualization. He has a MEng in electrical engineering from McGill University. He is a member of the IEEE and ACM. Contact him at guy.godin@nrc-cnrc.gc.ca.*

Philippe Massicotte *is a computer scientist with the Visual Information Technology Group at the National Research Council of Canada. His research interests include modeling and interactive visualization of large data sets, multiresolution displays, and virtual environments. He holds a BSc in computer science from the Université du Québec à Hull. Contact him at philippe.massicotte@nrc-cnrc.gc.ca.*

Guillaume Poirier *is a computer scientist with the Visual Information Technology Group at the National Research Council of Canada. His research interests include real-time computer graphics, nonphotorealistic rendering and perception, and human–computer interaction. Poirier has a MMath degree from the University of Waterloo. Contact him at guillaume.poirier@nrc-cnrc.gc.ca.*

François Blais *is a principal research officer with the National Research Council of Canada. His research interests cover various fields in digital and image processing, control, 3D, vision systems, and their applications. He has an MSc in electrical engineering from Laval University, Quebec City. Contact him at francois.blais@nrc-cnrc.gc.ca.*

J.-Angelo Beraldin *is a senior research officer at the Visual Information Technology Group in the Institute of Information Technology at the National Research Council of Canada. His research interests include sensor systems as well as engineering and signal processing for 3D vision systems and their application to the documentation of cultural heritage. He has an MS in electrical engineering from the University of Ottawa. Contact him at angelo.beraldin@nrc-cnrc.gc.ca.*