Stylized Rendering for Anatomic Visualization

A nonphotorealistic rendering (NPR) technique can help medical practitioners visualize anatomic models and illustrate them with different stylizations. This technique provides potentially useful rendering alternatives to conventional volume or surface rendering in medical visualization. Improved performance allows interactive visualization of anatomic models and adding stroke texture synthesis can enrich medical object illustrations.

Many sophisticated volume and surface rendering and visualization techniques for 3D medical images can help medical practitioners in their clinical practice. Volume rendering techniques directly display surfaces from sampled scalar data of 3D imaging without converting them into geometric primitives. Although volume rendering can produce high-quality rendering and has proved medically useful, however, it's usually much slower than surface rendering. Surface rendering is faster, can be easily accelerated by using conventional graphics cards, and is particularly helpful for real-time medical applications such as surgical planning.

Recently, researchers have proposed a compact multiresolution point rendering technique called QSplat[1] that's computationally efficient for rendering a large number of meshes of small primitives. The marching cube algorithm used in surface rendering always generates a large amount of small size triangular primitives. However, this approach requires a large amount of storage space, and it's unwise to render very small triangles on the screen. To speed up displaying it, we can use level of details to simplify data, but this requires a lot of time. On the other hand, QSplat can render good quality images and store less data than triangular meshes. In this article, we use the QSplat point data structure to represent and render anatomic objects from 3D medical imaging.

Recently, we've proposed a nonphotorealistic rendering (NPR) framework[2] based on QSplat to render point-sampled models with different stylizations. (See the "Nonphotorealistic Rendering" sidebar for more details on this topic.) In this article, we apply this technique to render and visualize anatomic data with NPR stylization. In contrast to conventional volume and surface rendering techniques, we can generate smoother and more suitable point data for displaying large medical imaging data. Our rendering performance is fast enough to allow interactive visualization of anatomic models. We also propose an example-based stroke synthesis framework to enrich the NPR stylizations.

Data Preprocessing

The input data to the point-based NPR rendering algorithm is a point data set that represents the isosurfaces extracted from medical volume data. For the examples in this article, we use vertices of triangular meshes from the marching cube algorithm's output as a point data set. Then, we use the QSplat

1521-9615/07/$20.00 © 2007 IEEE
Copublished by the IEEE CS and the AIP

Tong-Yee Lee, Chung-Ren Yan, and Ming-Te Chi
National Cheng Kung University

Nonphotorealistic Rendering

Traditional photorealistic rendering techniques focus on simulating the real physical illuminant, but nonphotorealistic rendering (NPR) techniques imitate artistic stylizations. (A thorough survey of related NPR techniques appears elsewhere.[1,2]) In contrast to traditional rendering methods, brush strokes used in NPR can illustrate more object information by using the strokes' orientation, thickness, and tone. To illustrate anatomic models, for example, the NPR technique enhances a visualization by using orientations to demonstrate a curvature's variations, displaying the luminance with thickness, and presenting the curvature's changes via variations of tone.

In the past few years, researchers have developed NPR techniques to visualize medical data. Xiaoru Yuan and Baoquan Chen[3] proposed a framework for illustrating surfaces in volume data by drawing feature lines such as silhouettes, valleys, and ridges to enhance the effect of visualization. Femg Dong and his colleagues[4] presented a volumetric hatching technique to generate pen-and-ink illustrations suited for medical illustration. Aidong Lu and her colleagues[5] used a stipple drawing technique to interactively visualize medical data sets.

Our framework for 3D NPR based on point based hierarchy provides efficiently NPR rendering and is flexible for various stroke styles.

References

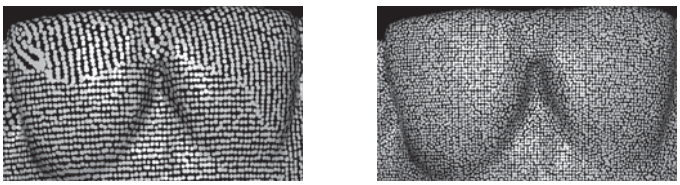
1. B. Gooch and A. Gooch, *Non-Photorealistic Rendering*, AK Peters, 2001.
2. T. Strothotte and S. Schlechtweg, *Non-Photorealistic Computer Graphics (Modeling, Rendering and Animation)*, Morgan Kaufmann, 2002.
3. X. Yuan and B. Chen, "Illustrating Surfaces in Volume," *Proc. VisSym 04: Joint IEEE/EG Symp. Visualization*, 2004, pp. 9–16.
4. F. Dong et al., "Nonphotorealistic Rendering of Medical Volume Data," *IEEE Computer Graphics & Applications*, vol. 23, no. 4, 2003, pp. 44–52.
5. A. Lu et al., "Non-Photorealistic Volume Rendering Using Stippling Techniques," *Proc. IEEE Visualization 2002*, IEEE CS Press, 2002, pp. 211–218.

algorithm to build a multiresolution point hierarchy. In this hierarchy, each point generally contains position, radius, and normal attribute information. To make these points better suited for medical illustration, we must perform some preprocessing.

Smoothing and Relaxation

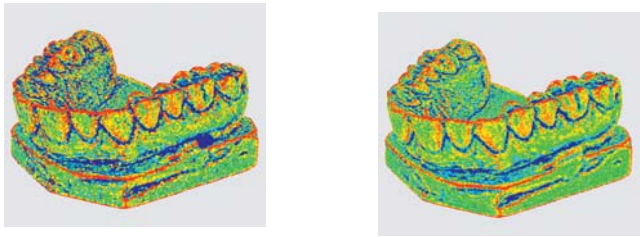
Our initial point data set from the marching cube might contain noise and errors generated while we extract and sample isosurfaces from volume data. To reduce this problem, we first apply the Multilevel Partition of Unity (MPU) Implicits technique[3] to our data set. This technique can compute several smooth implicit surfaces to fit a given point data set well. We call this *smooth filtering* our initial point data.

Next, we regularly sample these implicit surfaces and get a new data set, as Figure 1a shows. The user can control the sampling density to determine the number of point data. However, the regular sampling doesn't guarantee an even point distribution in 3D spatial space. Therefore, we perform a relaxation operation using repulsion force[4] to distribute the points evenly. Figure 1b shows the point distribution after relaxation.



(a) (b)

Figure 1. Model of a tooth (incisor). We can clearly see the number of data points distributed (a) before and (b) after relaxation.



(a) (b)

Figure 2. Curvature visualization. For this point data set, we can determine curvature information (a) without and (b) with smoothing and relaxation operations.

Directions and Curvature

When visualizing a 3D anatomic object with brush strokes, the brush strokes' distribution and orientation play an important role in determining the illustrative effects. Therefore, based on the surface variation of the point-sampled surface that Mark Pauly and his colleagues proposed,[4] we use principle component analysis (PCA) to estimate the surface curvature (see Figure 2) and the initial curvature directions for a point-sampled, anatomic surface. In Figure 2, we visualize curvature infor-

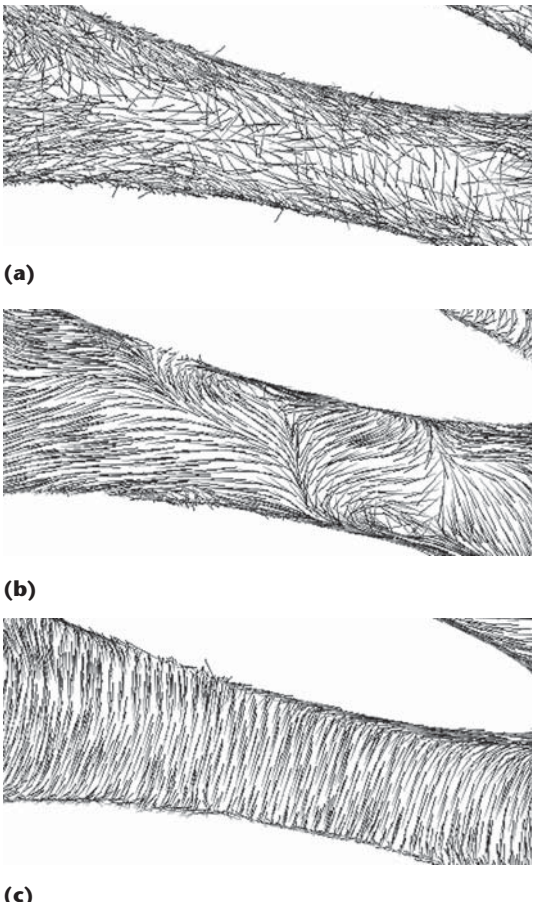


Figure 3. Visualization of curvature direction field. (a) The initial curvature directions we obtained using principle component analysis (PCA). The results after (b) RBF smoothing by randomly choosing feature points and (c) smoothing by high-curvature feature points.

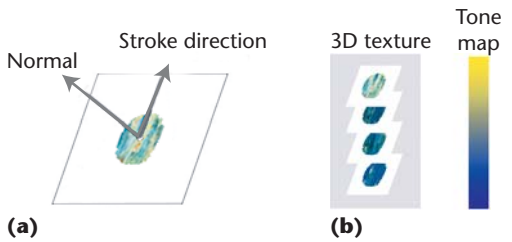


Figure 4. Concept diagram for rendering a point splat as a single texture-mapped quad. (a) Geometry setup for drawing a quad. (b) A series of stroke textures as a 3D texture map and 1D tone map.

mation for point data with and without smoothing and relaxation operations. With these two operations, curvature information becomes less noisy and better suited for further visualization.

The initial curvature directions obtained from PCA are often too random for illustration, as Figure 3a shows. Therefore, we chose the curvature directions of several feature points and then use a radial basis function (RBF) to interpolate these directions for nonfeature points. After this RBF smooth interpolation, curvature directions become smoothly distributed across the model. Figures 3b and 3c show the difference between randomly selecting feature points and selecting feature points with a higher curvature, respectively. Our experimental results suggest that choosing high curvature points helps better illustrate curvature direction information.

NPR QSplat Point-Sampled Hierarchy

As we mentioned earlier, we use the QSplat[1] technique to build a point-based hierarchy structure. When displaying a model with hundreds of millions of polygons, traditional algorithms for displaying, simplifying, and storing polygon meshes are impractical. Each primitive is a data point in the QSplat technique. Therefore, it's much easier to build a point-based hierarchy to simplify models and display visualization faster than traditional polygonal algorithms.

The QSplat hierarchy is built by recursively splitting along the longest axis of the input points' bounding box. This hierarchy is, in particular, suited for adopting culling (such as frustum and back-face culling) and level of detail (LOD) techniques for fast rendering. For data sets of this size, polygons are small, and their projected sizes on the image screen are always a few pixels or even less than one pixel. In this situation, it's more efficient to render a point primitive than a polygonal mesh. Because the point primitives don't require us to store the meshes' connectivity, the storage requirements for the entire QSplat point-based hierarchy are less than those of polygonal meshes.

Stroke Drawing with Stylization

In this article, we want to interactively illustrate an anatomic object with NPR strokes. For this purpose, we apply an NPR QSplat rendering technique[2] to render point-sampled anatomic models. In a previous study,[2] from the leaf to root nodes in the NPR QSplat hierarchy, we rendered points of various sizes into multiple-size strokes on the canvas. By selecting suitable abstractions, brush strokes, and lighting parameters, we can use this technique to interactively generate various NPR stylizations. (Related research with more technical details appears elsewhere.[2])

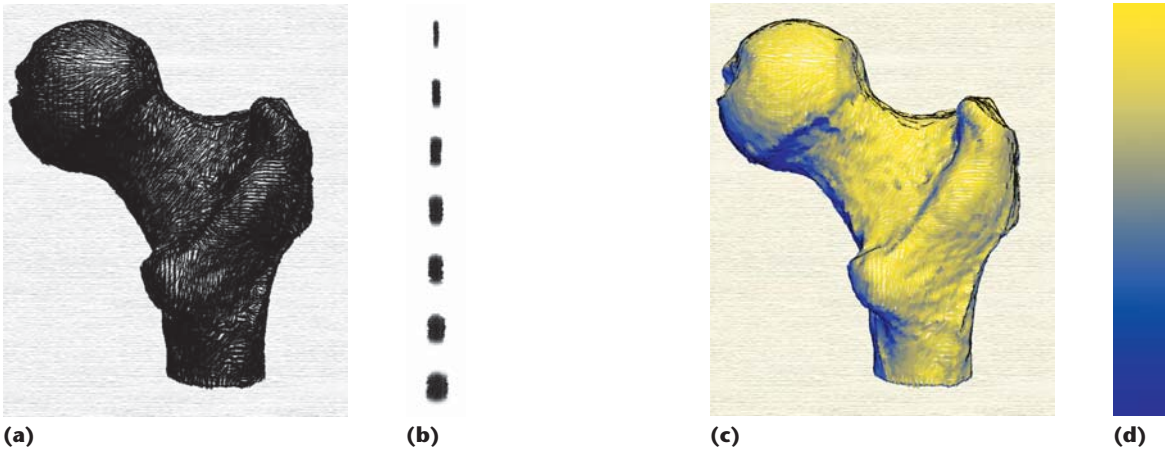


Figure 5. NPR rendering for anatomic objects. (a) Rendered results using (b) a series of stroke maps. (c) Combiming the stroke maps with (d) a 1D tone map helps illustrate anatomic objects.

Rendering with Strokes

After the preprocessing steps we discussed earlier, we know several items of information for each point in the NPR QSplat hierarchy: position, radius, normal vector, and direction vector field (such as curvature direction after RBF smoothing). We can render a point splat as a single texture-mapped quad sprite, as Figure 4 shows. For each point splat, the quad is located at the tangent plane of its normal and the center of a quad is located at its position (That is, for each 3D point splat P, it has a 3D position (x,y,z). We want to draw a quad to represent a tiny surface for it. We must first find the plane for this quad, so we choose the tangent plane of P's normal. The quad's center is located at $P(x,y,z)$.) A quad's size is determined by a point's radius, and its orientation is determined by the direction vector field.

To draw a texture-mapped quad,[2] we use a series of stroke maps as a 3D texture map such as the Tonal Art Map (TAM).[5] According to different lighting conditions, we can compute a continuous stroke texture map for a quad at rendering time.[2]

Figure 5a shows the rendered result of using the series of stroke maps in Figure 5b. In addition, we can replace Figure 5b with a 1D tone map (Figure 5d) to produce toon and Gooch shading (see Figure 5c)[6] effects to illustrate anatomic objects.

Example-Based Stroke Synthesis

In the past, most researchers acquired brush-stroke maps by scanning images or using image editing tools to make a series of strokes from bright to dark. Figure 6a shows an oil painting that consists of several stokes with different luminance and stroke directions. To enrich the variety of illustration, we can paint different stroke texture maps we

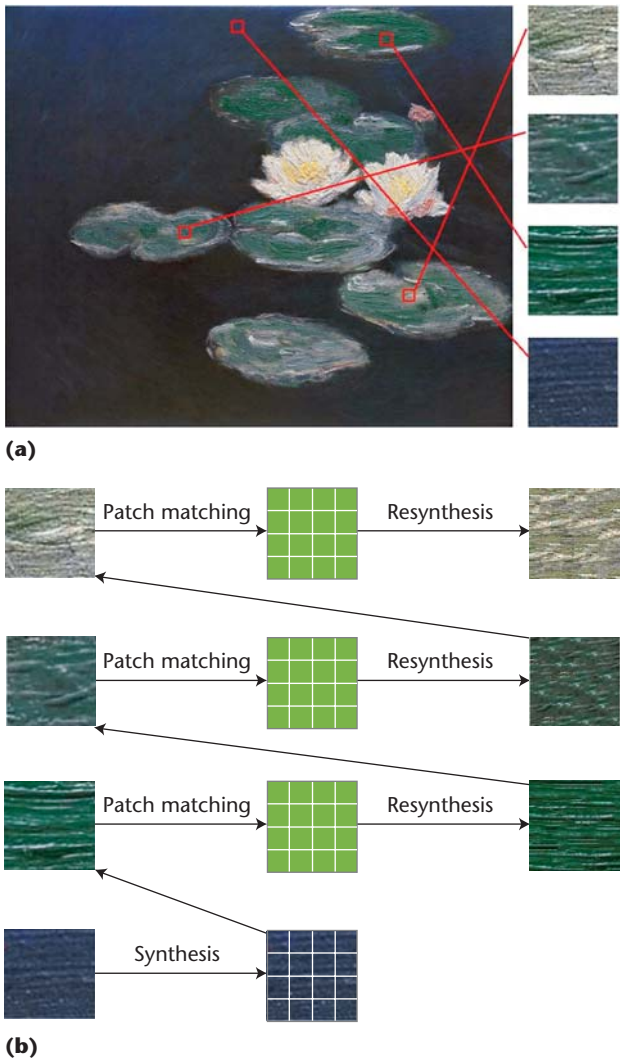
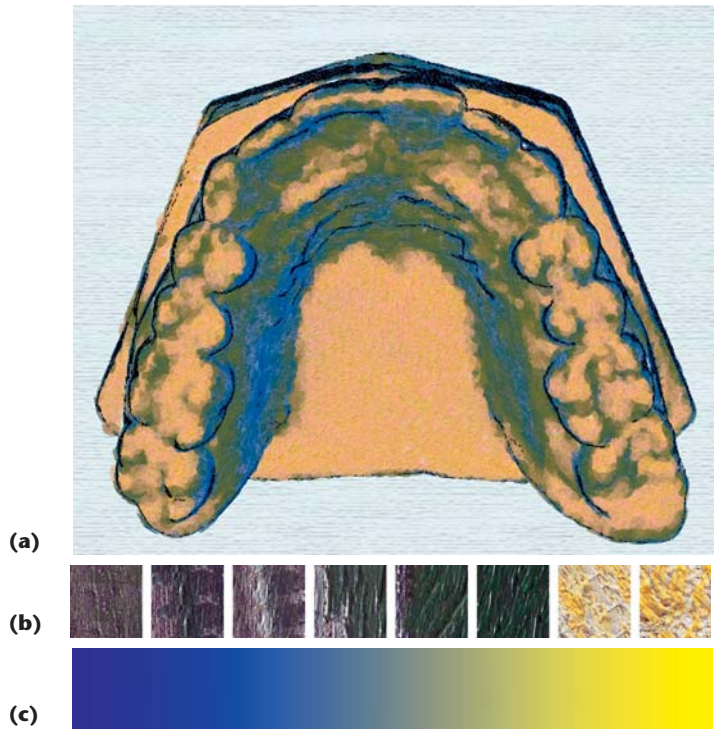


Figure 6. Stroke texture synthesis. (a) Stroke maps extracted from an input image. (b) Stroke synthesized results.

(a) (b)

Figure 7. Rendered results using synthesized stroke textures in Figure 6b. (a) The rendered result with the color of the synthesized stroke texture, and (b) the rendered result of combining a specific color (in this instance, gray) with the intensity variance of the synthesized stroke texture.



(a)

(b)

(c)

Figure 8. Model of teeth (incisors). (a) This is the rendered results using (b) synthesized stroke textures with (c) tone mapping.

obtained from Figure 6a onto the 3D anatomic model. However, these selected strokes might have different stroke directions. If we directly paint these strokes onto a model without any modification, the rendered illustrations' stroke directions might look inconsistent (see Figure 6b). We propose using a texture synthesis algorithm to produce more consistent stroke textures and present the variety between brightness and darkness.

Figure 6 shows the basic idea of this method. Here are the major steps:

1. Select example stroke textures from the input image and sort them by their average luminance.
2. Align example textures in the same direction as much as possible.
3. Start synthesizing a new texture from the example texture with the darkest luminance using Tong-Yee Lee and Chung-Ren Yan's resynthesis algorithm.[7]
4. Before continuing to synthesize the next example texture *T*' with the brighter luminance, split the current synthesized texture image *T* into several regular patches (see Figure 6b).
5. Use the patches from Step 4 to find the most similar/matched patches in the next example texture and paste the most similar patch to the corresponding position on *T*'.
6. Among the adjacent patch boundaries, use the resynthesis algorithm[7] to reduce the discontinuity in the synthesized texture.
7. Repeat Steps 4 through 6 until all example textures are synthesized.

In these steps, the user can select several regions as stroke maps (Step 1) characterized with different luminance and stroke directions. For each region, we can convert its RGB color into the YUV color space. Then, we sort all example stroke maps by luminance (Y channel) from brightest to darkest. For example, Figure 6b contains four selected, sorted stroke maps. However, the selected stroke maps might have significant differences in their stroke directions. To deal with problem, we first manually rotate these maps to roughly align their directions as much as possible in Step 2. Then, we use the texture synthesis technique[7] to generate a series of maps with more coherence in their stroke directions. The patch-based method[7] can run quickly and is helpful for maintaining stroke consistency.

We begin texture synthesis from the example texture with the darkest luminance (Step 3). We repeat this synthesis process for the textures from the darkest to the brightest luminance. In this repeated loop, after we finish texture synthesis for the current example texture, we uniformly split this newly generated texture image *T* and continue synthesizing another texture image *T*' with the brighter luminance based on *T*.

In Step 5, we use luminance for the patch matching metric. In the right side of Figure 6b, we show the synthesized stroke maps using the proposed framework. We can see that these generated stroke maps have more consistent stroke directions.

Results

We implemented our system using OpenGL on the Nvidia GeForce 6800 graphic card. The computing platform is a 2.8-GHz Intel Pentium IV with 2 Gbytes of RAM. We can render more than 25 frames per second when drawing several tens of thousands of display splat nodes on the canvas. The number of leaf nodes in point hierarchy for anatomic models in this article varied from 200,000 to 400,000.

Figure 7 shows the rendered result using the synthesized stroke textures in Figure 6b. We rendered Figure 7a with the original color of example stroke textures. We then modulated the stroke texture's color with the user-specified color (which is gray in this example) to get Figure 7b.

Figure 8 shows another interesting example. In addition to synthesized stroke textures, we modulated the result by tone mapping, similar to Figure 5c.

Figure 9 shows the result of rendering with different stroke alpha textures. Figure 10 illustrates models with *stippling*[2] (an NPR technique that uses small dots to simulate varying degrees of shading) by rendering splat primitives with a different radius according to the illumination condition.

In a previous work,[2] our NPR QSplat system visualized models with multipass rendering. Figure 11 shows the result with two-pass rendering. In this example, we rendered the first pass using the curvature information encoded in color and the second pass using the stroke textures. In this manner, we can better visualize the geometric property of models using curvature information.

In the near future, we plan to collaborate with colleagues in our medical school to investigate the usefulness of NPR rendering in medical applications. In time, as NPR techniques mature, they will prove useful in future medical visualization. Using NPR, we're investigating better medical visualization tools that might help medical practitioners plan and simulate their surgery operations. Furthermore, for a given set of features specified by users, we want to develop techniques to automatically determine the best illustrative view using NPR for medical visualization, Finally, we also plan to consider better camera-sampling analysis[8] to improve point-based rendering quality. CiSE

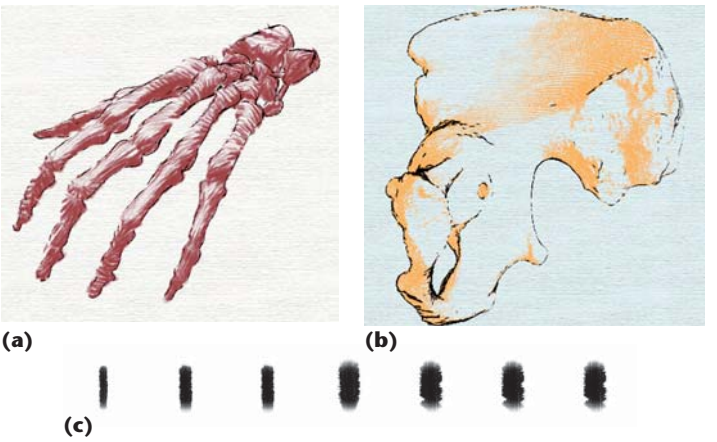


Figure 9. Rendered results with different stroke textures. (a) The hand and (b) hip images are rendered using (c) the stroke textures. Each stroke texture corresponds to a illumination condition varying from the bright to dark.

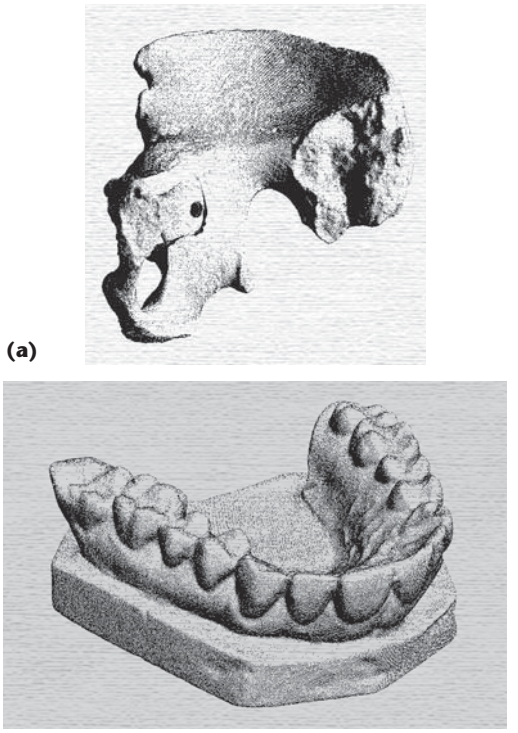


Figure 10. Rendered results for (a) a hip and (b) teeth with stippling.

Acknowledgments

The National Science Council, Taiwan, Republic of China, supported this project under contracts NSC-93-2213-E-006-060, NSC-94-2213-E-006-063, NSC-95-2221-E-006-368, and NSC-95-2422-H-018-001.

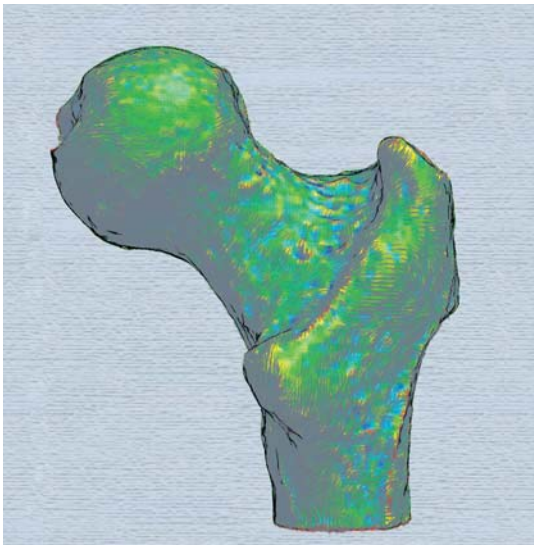


Figure 11. Rendered results with two-pass rendering. This result combines the curvature information encoded in color (first pass) with stroke textures (second pass) to help better visualize a model's geometry.

References

1. S. Rusinkiewicz and M. Levoy, "QSplat: A Multiresolution Point Rendering System for Large Meshes," *Proc. SIGGRAPH 00*, ACM Press, 2000, pp. 343–352.
2. M.-T. Chi and T.-Y. Lee, "Stylized and Abstract Painterly Rendering System Using a Multi-scale Segmented Sphere Hierarchy," *IEEE Trans. Visualization and Computer Graphics*, vol. 12, no. 1, 2006, pp. 61–72.
3. Y. Ohtake et al., "Multi-Level Partition of Unity Implicits," *ACM Trans. Computer Graphics*, vol. 22, no. 3, 2003, pp. 463–470.
4. M. Pauly, R. Keiser, and M. Gross, "Multi-Scale Feature Extraction on Point-Sampled Models," *Proc. EUROGRAPHICS 2003*, Blackwell, 2003, pp. 281–289.
5. E. Praun et al., "Real-Time Hatching," *Proc. ACM SIGGRAPH 2001*, ACM Press, 2001, pp. 581–586.
6. A. Gooch et al., "A Non-Photorealistic Lighting Model for Automatic Technical Illustration," *Proc. ACM SIGGRAPH 1998*, ACM Press, 1998, pp. 447–452.
7. T.-Y. Lee and C.-R. Yan, "Feature-Based Texture Synthesis," *Proc. Int'l Conf. Computational Science and its Applications* (ICCSA 05), LNCS 3482, Springer-Verlag, 2005, pp. 1043–1049.
8. P.-H. Lin and T.-Y Lee, "Camera- Sampling Field and Its Applications," *IEEE Trans. Visualization and Computer Graphics*, vol. 10, no. 3, 2004, pp. 241–251.

Tong-Yee Lee *is a professor in the Department of Computer Science and Information Engineering at National Cheng-Kung University in Taiwan and leads the Computer Graphics Group/Visual System Lab at National Cheng-Kung University (http://couger.csie.ncku.edu.tw/~vr). His current research interests include computer graphics, nonphotorealistic rendering, image-based rendering, visualization, virtual reality, surgical simulation, and distributed and collaborative virtual environments. Lee has a PhD in computer engineering from Washington State University, Pullman. He is also an associate editor of* IEEE Transactions on Information Technology in Biomedicine. *He is a member of the IEEE, ACM, and IEICE. Contact him at tonylee@mail.ncku.edu.tw.*

Chung-Ren Yan *is a PhD student in the Department of Computer Science and Information Engineering at National Cheng-Kung University, Taiwan. His research interests include computer graphics and texture synthesis. Yan has an MS in computer science and information engineering from Chung-Hua University, Taiwan. Contact him at chongren@vision.csie.ncku.edu.tw.*

Ming-Te Chi *is a PhD student in the Department of Computer Science and Information Engineering at National Cheng-Kung University, Taiwan. His research interests include computer graphics and nonphotorealistic rendering. Chi has an MS in computer science and information engineering from the National Cheng-Kung University, Taiwan. Contact him at dodowell@csie.ncku.edu.tw.*