Uncertainty Visualization Methods in Isosurface Rendering

Philip J. Rhodes,[1] Robert S. Laramee,[2] R. Daniel Bergeron,[1] and Ted M. Sparr[1]

[1]Computer Science Department, University of New Hampshire, Durham, NH, {rhodes,rdb,tms}@cs.unh.edu
VRVis Research Center, Vienna, Austria, Laramee@VRVis.at, www.VRVis.at

Abstract

We describe two techniques for rendering isosurfaces in multiresolution volume data such that the uncertainty (error) in the data is shown in the resulting visualization. In general the visualization of uncertainty in data is difficult, but the nature of isosurface rendering makes it amenable to an effective solution. In addition to showing the error in the data used to generate the isosurface, we also show the value of an additional data variate on the isosurface. The results combine multiresolution and uncertainty visualization techniques into a hybrid approach. Our technique is applied to multiresolution examples from the medical domain.

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Methodology and Techniques I.3.8 [Computer Graphics]: Applications

1. Introduction

With the exception of geographic information systems (see, for example, Hunter and Goodchild [6]), there has not been a large amount of research into identifying and visualizing the uncertainty in data. Recently, however, researchers in other fields have begun to address this issue. For example, Lodha and Pang have experimented with visualizing uncertainty in vector fields [9, 10, 12, 14, 15], and Cignoni et al. [3] have developed a tool, *Metro*, for visualizing mesh surface approximation error. In addition to the very difficult problem of identifying and maintaining the error itself, it is also very difficult to present that error to the user in an effective and meaningful form.

Incorporating uncertainty into any visualization requires rendering at least one additional variate (actually we need to add one new variate for each variate that has its own error measure). Although many innovative multivariate visualization techniques have been developed in recent years and some have proven useful in some situations, this is a challenging problem which is exacerbated by the enormous size of modern scientific data.

In principal, we would like to incorporate the uncertainty information into the data visualization. When the error information is locally defined (i.e., it has about the same resolution as the data), this approach usually results in some form of degradation in the display of the data itself. Wittenbrink et al. [15] call these overloading techniques (as opposed to their glyph-based technique which they call *verity visualization*). Cedilnik and Rheingans [2] use annotations with a visualization in order to reduce the distraction caused by the error visualization. In order to be most effective, it is important that the user have the ability to turn the uncertainty visualization on and off interactively. With uncertainty visualization disabled, the user is likely to have the best chance of understanding the fundamental nature of the data. After enabling the uncertainty visualization, the user can now get an understanding of the error in the data.

1.1. Multiresolution Data

One major reason for the relatively low emphasis placed on uncertainty visualization in the past is that uncertainty information is seldom available except in very abstract forms. With the growing interest in the generation of coarse resolution approximations to a large dataset, explicit error information is more readily available. Creating multiresolution data often introduces additional error into the data, but it is often relatively easy to measure this new error and it is usually significantly greater than the error in the original data. Consequently, we can expect to be able to create and access error information about coarse resolutions of a multiresolution data hierarchy. Furthermore, it is particularly important to incorporate error into the visualization of data that is only a coarse approximation to the real data. A scientist needs to know what portions of a coarse resolution visualization have relatively low error (and therefore are an authentic representation of the data in that area) and which have a relatively high error. The representation of the low error regions is likely to be reasonably authentic, but the scientist is likely to want to visualize areas of high error at a higher resolution, e.g., an adaptive resolution visualization.

1.2. Isosurface Rendering

Isosurface rendering is a very good candidate for adding uncertainty visualization. Rendering an isosurface within a volume of univariate data is a very effective technique for many applications. Since all the data being visualized has the same data value, the particular value does not need to be incorporated into the visualization. Conventional isosurface rendering assigns a constant color to the vertices of the triangles that define the isosurface, and uses standard lighting models and Gouraud interpolation to give a sense of the shape of the isosurface. Consequently the color (and texture) parameter may be available for visualizing the uncertainty. It is important to realize that this approach allows us to visualize the error of the data used to generate the isosurface rather than the error between the low and high resolution versions of the isosurface.

1.3. Overview

In this paper we describe some experiments with incorporating an uncertainty variate into isosurface rendering. Our visualization tool is part of a broader research effort to develop a formal model and a support environment for dealing with large multiresolution and adaptive resolution data sets [8, 13]. A fundamental aspect of this model is the incorporation of local error measures into the data representation. Although our uncertainty visualization does not depend on any particular technique for generating the volume data, we start by describing our wavelet-based multiresolution volume data which does incorporate a meaningful error component. We conclude with some specific examples of the visual results of the approach.

2. Multiresolution Volume Data

Our motivation for developing a tool for incorporating uncertainty into isosurface rendering arose from our interest in using multiresolution data representation for large scientific data sets [16, 17]. We are particularly interested in generating coarse approximations to a large dataset that are more tractable in terms of size but still retain sufficient authenticity to be useful. For this approach to be viable, it is critical that we provide an estimate of the error that is introduced into the coarse data on a local basis. In principle, every data point in each level of a multiresolution data hierarchy includes both data and an error measure associated with that data –its uncertainty. In other words, we want to identify the regions of the data where the coarse representation is not an authentic representation of the original data.

3. Isosurface Rendering with Error

We have extended the standard Marching Cubes algorithm [11] to incorporate a measure of the error of the data. Volume data points contain both data values and the error associated with each data point. During the Marching Cubes algorithm, we compute an error associated with each triangle vertex by interpolating between the error values of the associated cube vertex error values. We use the error value for each triangle vertex to modify the appearance of that portion of the isosurface.

3.1. Uncertainty Representation using Hue

The vertices of the triangles that define the isosurface are assigned a color based on hue, saturation and brightness and the triangles are then rendered using an external light source and Gouraud shading. For basic isosurface rendering (without uncertainty enabled), all triangle vertices have the same color. The user may choose to map the uncertainty to any of the three color parameters (hue, saturation and brightness), while leaving the other two parameters fixed. In addition, the user can interactively select what constant values should be used for the other two color parameters. Since hue is specified as an angle between 0 and 360, it is clearly not desirable that the full range be used. If it were, the largest and smallest error values would have the same hue. Consequently, we allow the user to select the range of hues that should be used for the uncertainty.

Although we allow users to assign the uncertainty rendering to any of the three color parameters (hue, saturation, brightness), we recognize that the most reasonable mapping for this particular problem is to map the error to the hue. The brightness component is needed in order to effectively represent the shape of the isosurface and it is generally agreed that we are more sensitive to hue changes than saturation changes. In general, the perceptual issues associated with color usage are beyond the scope of this paper.

3.2. Uncertainty Rendering using Texture

We have developed a second error visualization method that uses texture to show regions of the isosurface with high uncertainty. Textures and texture hardware have been used by various researchers as an aid to data visualization [1, 3, 7]. These approaches either use texture hardware to accelerate visualization, or rely heavily on the color component of the textures for their visual effects. Our approach, on the other hand, does not use hue as part of the texture, so that it is available for visualizing another variate on the isosurface.

Our implementation uses a second texture surface which envelops the original isosurface, but is slightly offset from it.

A stipple texture is mapped to this surface, and the opacity is varied according to the uncertainty of the data. That is, the texture will be most visible in areas with high uncertainty, but absent or faint where uncertainty is low.

Figure 4 *left*, shows the relationship between color and texture visualization. The top row simply shows a set of typical hues. The middle row shows a texture imposed over a green surface. The texture becomes increasingly visible as the tiles progress to the right. Notice that the underlying green can still be seen, even in the rightmost tile. In the bottom row, the hue of each tile varies as in the top row, but now we have imposed the texture as well. The texture becomes increasingly visible as we progress to the right. In either case, both the texture and underlying hue are suitable for visualizing distinct variates. For example, with CFD simulation data, we might use the pressure variate to compute the isosurface, map the error of the pressure to the texture surface and render the temperature variate to the surface hue.

4. Experimental Results

Our isosurface software is implemented in Java and is built on the VisAD system [4, 5] which uses Java3D for rendering. Figures 1 through 3 were rendered directly in this system. The remaining figures were rendered in a separate program using *gl4java* (http://www.jausoft.com/gl4java.html) since we needed a lower-level API to implement the texture-based error visualization.

For these tests we used a CAT scan of a cadaver head courtesy of North Carolina Memorial Hospital and Siemens Medical Systems, Inc., Iselin, NJ. The original data is $113 \times 256 \times 256$. For the convenience of the wavelet transform, we appended 15 slices of padding to get a $128 \times 256 \times 256$ dataset. We then applied a 2D Haar wavelet to each slice and three successive 3D Haar wavelets to get a 4 level hierarchy. Figure 1 *left*, shows an isosurface rendering of the 128^3 representation for the isovalue 0.185 (a skin value). The next two coarser resolutions of the skin isosurface are shown in Figure 1 *middle*, (64^3) and Figure 1 *right*, (32^3). It is clear that the surface shown in Figure 1 *middle*, is coarser than that shown in Figure 1 *left*, but the overall impressions of the two surfaces are very similar. Figure 1 *right*, however, shows a substantial loss of accuracy of the surface. Hence these three resolutions serve as a useful basis for comparison.

4.1. Uncertainty Mapped to Hue

Figure 2 *left*, shows the skin value isosurface of the 128^3 resolution representation with constant saturation and brightness and uncertainty mapped to the range of hue from 144 degrees (green) down to 0 degrees (red). In other words, green represents low uncertainty and red high uncertainty. The error associated with the 128^3 dataset is very low and this is reflected in the visualization. At normal scale no high error areas are visible although at very high magnification it is possible to see some very light pink areas around the mouth region. Figure 2 *middle*, shows the uncertainty visualization of the 64^3 dataset using the same visualization parameters as Figure 2 *left*. Here more error is readily discernible as reddish areas around the mouth, eyes, and forehead. Figure 2 *right*, shows the 32^3 resolution data with the same visualization parameters. As we would expect, there is obviously increased error in many areas of the isosurface.

It is not clear what range of error might be expected for different kinds of input data and so it is also unlikely that there is a single ideal mapping of error to color. Figure 3 *middle*, and Figure 3 *right*, show the 64^3 and 32^3 data sets with a narrower hue range (108 to 0) intended to accentuate the error while maintaining red as the color of the highest error.

In addition to the MR isosurface renderer we have shown here, we have incorporated this technique into a system for creating and rendering adaptive resolution representations [8]. Figure 3 *left*, shows a result of that work.

4.2. Uncertainty Mapped to Texture Opacity

The last several figures show our texture based error visualization method. Figure 4 *middle*, shows the skull data with error mapped to texture transparency. Regions of high error can be seen above the ear and proceeding left towards the forehead. Figure 4 *right*, demonstrates the use of texture visualization of error while hue is mapped to another variate. For this example, we generated a synthetic variate based on polygon normals to demonstrate the technique. It is difficult to see the texture in a small picture, but we provide an enlargement of the forehead area in Figure 5. The texture visualized error can be clearly seen even though it interferes minimally with the accurate visualization of the second synthetic variate.

4.3. Error Visualization Performance

We ran our tool on a dual processor 800MHz Mac OS X machine using a GeForce 3 graphics card that has 64MB of memory. This tool assembles the isosurface polygons into an OpenGL display list, which is then used for rendering. Therefore, the computations for error visualization occur only once. Such an approach works well for static data and takes advantage of the graphics card, which also impacts performance. Table 1 shows the performance in FPS of both hue and texture based error visualization methods for different data set sizes. Since the texture-based technique doubles the number of polygons rendered, speed is roughly halved when compared to the hue based method. On modest hardware, we achieve marginally interactive speeds even with the 128^3 data set using the texture-based method.

5. Conclusions and Future Research

Isosurface rendering of multiresolution data is an ideal candidate for including uncertainty visualization. The incorpo-

original data size	Err Vis with Hue	Err Vis with Texture
32^3	136.1	65.9
64^3	31.5	14.9
128^3	6.5	1.7

Table 1: *Sample frame rates for the error visualization.*

ration of the uncertainty into the visualization using color is relatively straightforward and provides effective feedback about where the visualization is unreliable without detracting significantly from the data visualization, especially in areas of low uncertainty. Texture based visualization of uncertainty has the additional benefit of making the surface color available for the visualization of another variate. In the future we intend to incorporate uncertainty into more complex visualization techniques, such as direct volume rendering (DVR) and flow visualization.

Acknowledgements

The research reported here is supported in part by the National Science Foundation, grants IIS-9871859 and IIS-0082577. We also thank the Austrian KPlus program (www.kplus.at) for their financial support.

References

1. I. Boada, I. Navazo, and R. Scopigno. Multiresolution Volume Visualization with a Texture-Based Octree. In *The Visual Computer*, volume 17(3), pages 185–197. Springer, 2001. 2
2. A. Cedilnik and P. Rheingans. Procedural annotation of uncertain information. In *Proceedings Visualization 2000*, pages 77–84. IEEE Computer Society Technical Committee on Computer Graphics, 2000. 1
3. P. Cignoni, C. Rocchini, and R. Scopigno. Metro: Measuring Error on Simplified Surfaces. *Computer Graphics Forum*, 17(2):167–174, 1998. 1, 2
4. W. Hibbard. Connecting People to Computations and People to People. *Computer Graphics*, 32(3):10–12, 1998. 3
5. W. Hibbard and D. Santek. Interactivity is the Key. In *Proceedings of the Chapel Hill Workshop on Volume Visualization*, pages 39–43, May 1989. 3
6. G. J. Hunter and M. F. Goodchild. Managing Uncertainty in Spatial Databases: Putting Theory into Practice. In *URISA Proceedings*, volume 1, pages 1–14, Atlanta, Georgia, 1993. 1
7. E. C. LaMar, B. Hamann, and K. I. Joy. Multiresolution Techniques for Interactive Texture-based Volume Visualization. In David Ebert, Markus Gross, and Bernd Hamann, editors, *IEEE Visualization '99*, pages 355–362, San Francisco, 1999. IEEE. 2
8. R. S. Laramee and R. D. Bergeron. An Isosurface Continuity Algorithm for Super Adaptive Resolution Data. In John Vince and Rae Earnshaw, editors, *Advances in Modelling, Animation, and Rendering: Computer Graphics International (CGI 2002)*, pages 215–237, Bradford, UK, July 1-5 2002. Computer Graphics Society, Springer. 2, 3
9. S. K. Lodha, A. Pang, R. E. Sheehan, and C. M. Wittenbrink. UFLOW: Visualizing Uncertainty in Fluid Flow. In *IEEE Proceedings of the Conference on Visualization*, pages 249–254, Los Alamitos, October 27–November 1 1996. 1
10. S. K. Lodha, C. M. Wilson, and R. E. Sheehan. LISTEN: Sounding Uncertainty Visualization. In Roni Yagel and Gregory M. Nielson, editors, *IEEE Visualization '96*, pages 189–196. IEEE, 1996. 1
11. W. E. Lorensen and H. E. Cline. Marching Cubes: a High Resolution 3D Surface Construction Algorithm. In *SIGGRAPH '87 Conference Proceedings (Anaheim, CA, July 27–31, 1987)*, pages 163–170. Computer Graphics, Volume 21, Number 4, July 1987. 2
12. A. Pang and J. Furman. Data Quality Issues in Visualization. In *SPIE proceedings on Visual Data Exploration and Analysis*, volume 2178, pages 12–23, 1994. 1
13. P. J. Rhodes, R. D. Bergeron, and T. M. Sparr. A Data Model for Distributed Multiresolution Multisource Scientific Data. In G. Farin, H. Hagen, and B. Hamann, editors, *Hierarchical and Geometrical Methods in Scientific Visualization*, Heidelberg, Germany, 2002. 2
14. Q. Shen, A. Pang, and S. Uselton. Data Level Comparison of Wind Tunnel and Computational Fluid Data Dynamics Data. In *Proceedings of the IEEE Conference on Visualization*, pages 415–418, 1998. 1
15. C. M. Wittenbrink, A. T. Pang, and S. K. Lodha. Glyphs for Visualizing Uncertainty in Vector Fields. *IEEE Transactions on Visualization and Computer Graphics*, 2(3):266–279, September 1996. 1
16. P. C. Wong and R. D. Bergeron. Authenticity Analysis of Wavelet Approximations in Visualization. In *Proceedings of the IEEE Conference on Visualization*. IEEE, 1995. 2
17. P. C. Wong and R. D. Bergeron. Performance Evaluation of Multiresolution Isosurface Rendering. In *Proceedings, Dagstuhl '97 Scientific Visualization*, pages 332–341. IEEE Computer Society Press, 2000. 2

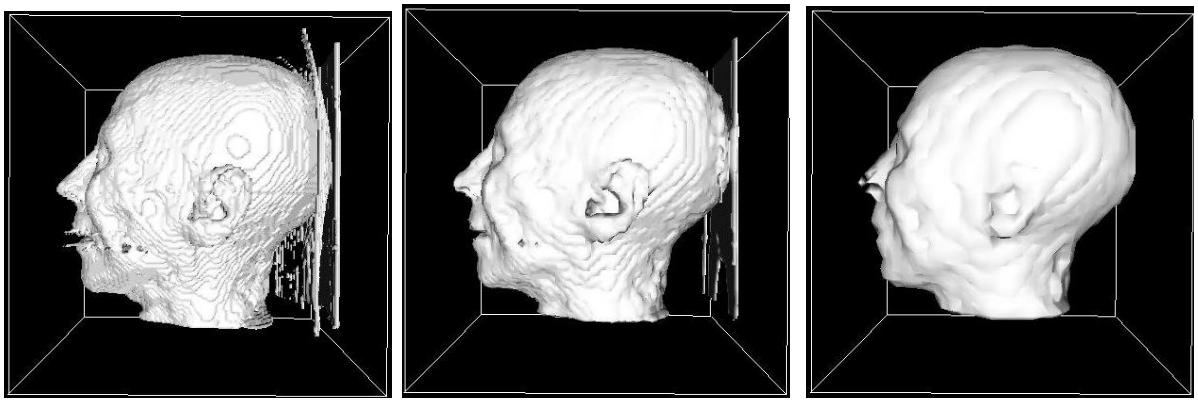

Figure 1: *Three isosurfaces corresponding to skin, isovalue 0.185, with uncertainty disabled: (left)* 128^3*, (middle)* 64^3*, (right)* 32^3

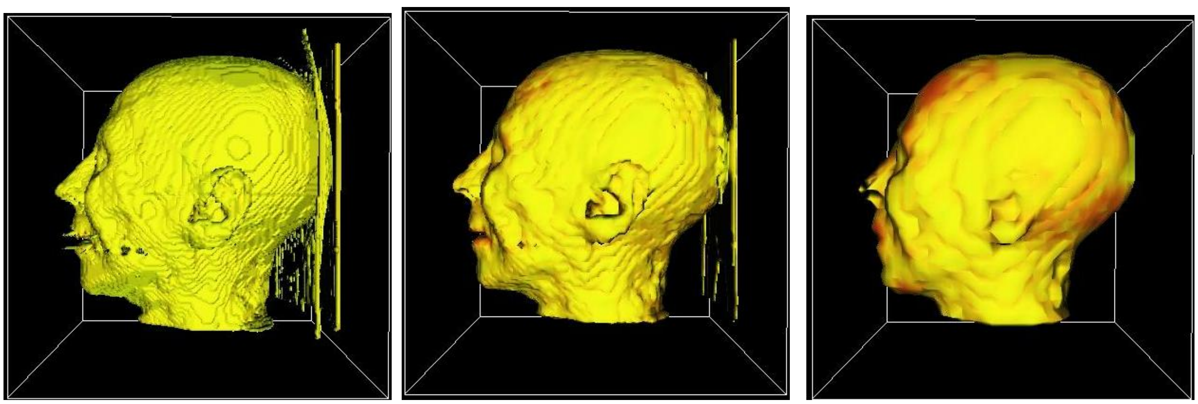

Figure 2: *The three isosurfaces from Figure 1 with uncertainty mapped to hue in the range (144, 0): (left)* 128^3*, (middle)* 64^3*, (right)* 32^3

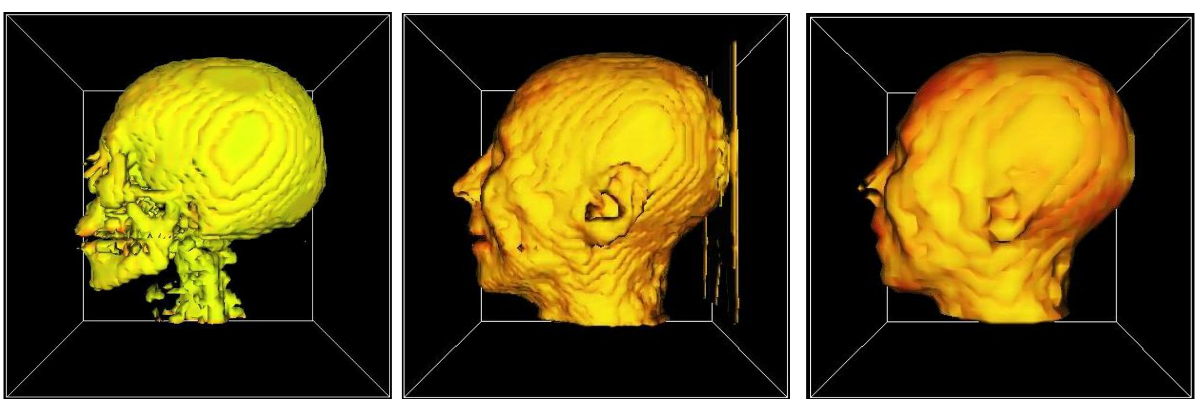

Figure 3: *(left) An adaptive resolution isosurface corresponding to bone, isovalue 0.378, with uncertainty mapped to hue in the range (144, 0), (middle) uncertainty mapped to hue in the range (108, 0),* 64^3*, (right) uncertainty mapped to hue in the range (108, 0),* 32^3

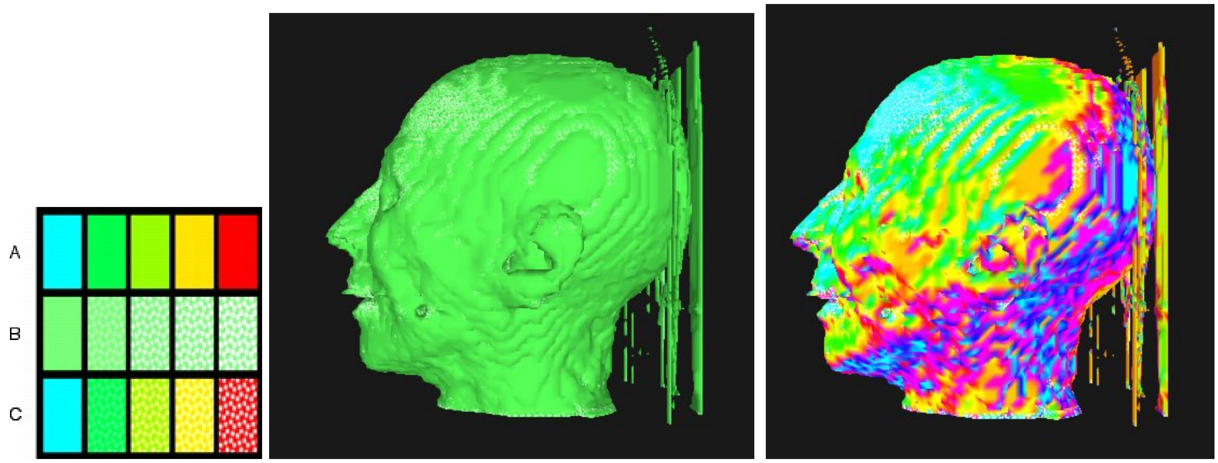

Figure 4: *(left) A) Varying hue only, B) Texture of increasing opacity over constant hue, C) Increasing opacity over varying hue (middle) Error mapped to texture opacity over a constant hue, (right) Error mapped opacity over a hue mapped to a synthetic variate*

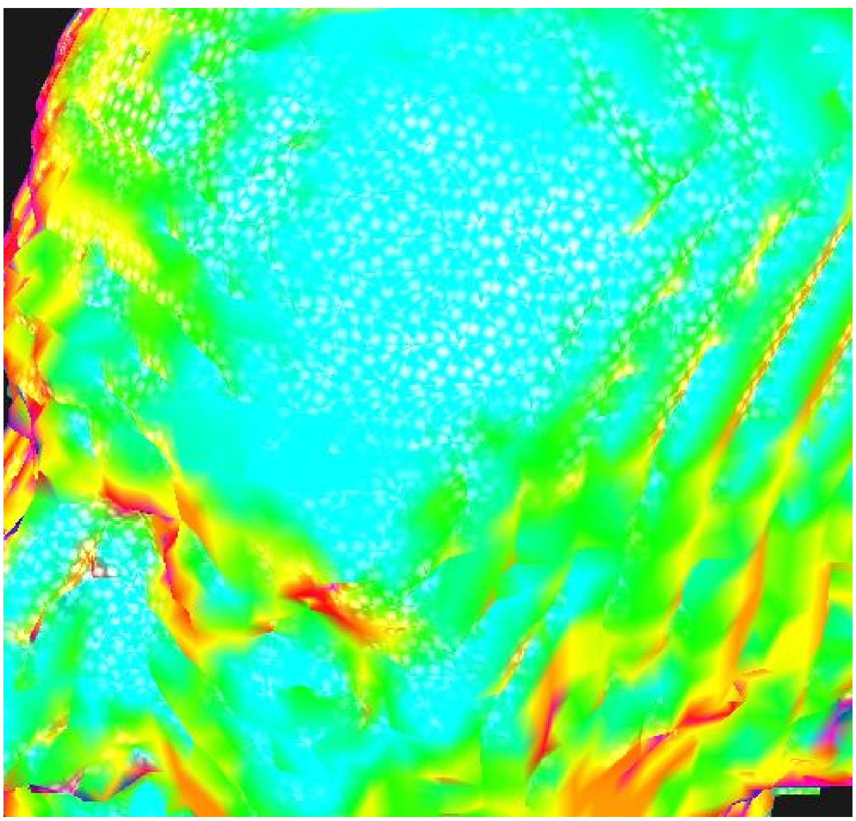

Figure 5: *A close up of the forehead region from Figure 4, right. The texture can be clearly seen, but interferes minimally with the underlying hue.*