Cartographic Rendering of Elevation Surfaces using Procedural Contour Lines

Neha Manya
Louisiana State University
Center for Computation & Technology
Department of Computer Science and Engineering Division
3127 P. F Taylor Hall
USA, LA-70803, Louisiana
nmanya1@tigers.lsu.edu

Isaac Ayyala
Louisiana State University
Center for Computation & Technology
Department of Electrical and Computer Engineering
3101 P. F. Taylor Hall
USA, LA-70803, Louisiana
iayyal1@tigers.lsu.edu

Werner Benger
Louisiana State University
Center for Computation & Technology
213 Johnston Hall
USA, LA-70803, Louisiana
werner@cct.lsu.edu

ABSTRACT

We describe a method to render a height field given on a triangular mesh in a manner that is similar to well-known cartographic map views. This approach utilizes hypsometric tints and creates topographic contour lines procedurally as part of the rendering process. We discuss essential graphical enhancement techniques such as line smoothing, gradient compensation, contour line visibility fading, nonlinear elevation mapping, and shininess variation. The method is exemplified upon various numerical data sets from application sciences, demonstrating its ability to visually allow quantitative and qualitative precise assessment of the data values.

Keywords

Contour lines, hypsometric tints, scientific visualization, bathymetry, coastal modeling, terrain visualization.

1. INTRODUCTION

The challenge of rendering bathymetric data is emphasizing features over a large scale while sustaining accurate data representation. Cartography is an ancient field of interest to draw upon for this purpose. The use of colors to represent ranges of elevation, known as hypsometric tints, is said to have been invented by Leonardo da Vinci [Vin03]. Contour lines complement hypsometric tints to depict more detailed information quantitatively. Both methods are familiar and intuitive. A similar approach is therefore highly desirable for application upon numerical datasets where interactive exploration is mandatory for data analysis, especially when exploring complex geomorphology.

Deltaic estuaries in the northern Gulf of Mexico are good examples of complex geomorphological systems. These estuaries are characterized by numerous channels and wetland networks on the order of 20m – 100m in size, and have a highly complex bathymetry within a fairly narrow depth range of 0m – 5m. Implementation of 3-dimensional hydrodynamic models is particularly challenging in these systems because numerical grids need to resolve both the complex bathymetry and intricate wetland features. Consequently, bathymetric rendering and visualization can provide invaluable information for the generation of 3D numerical grids.

Related Work

The Collaborative Ocean Visualization Environment (COVE [Gro10]) has been designed with oceanographers to provide a set of elaborated tools for bathymetric data exploration, including hypsometric tinting and display of simple contour lines. In our approach we seek to achieve even higher accuracy in emphasis of finer variations in elevation, limited only by numerical precision. In [Gul10] an algorithm is presented to extract contour lines from scanned color topographic maps. Topography maps already have the contour lines in them, but they might be broken or overlapping. They provide techniques to filter noise, remove bifurcations and connect lines properly. In contrast, here we intend to create contour lines from the data itself. [Du04] discusses the limitations of contour line generation. They found that useful contour lines need to be joined, but should only be extracted if they were not too fragmented and enough spatial information was available. The quality of procedural contour lines such as in our approach will of course depend on the

 ISBN 978-80-86943-74-9

quality of available spatial information as well, but joining and fragmentation of lines is not a hindrance.

2. RENDERING TECHNIQUE

We intend to render a height field, such as given in the form of a triangular surface mesh, similar to the well-known cartographic views known from topographic maps. Firstly, this implies using a colorization scheme widely known as hypsometric tints: it renders bathymetric regions (below 'sea level') using bluish shades and hypsometric regions (above sea level) in shades from green via yellow to brown to red and finally gray – white colors, in approximation of terrestrial average colors. Secondly, we like to draw contour lines to allow quantitative reading of height levels. Because explicit calculation of contour lines as required for numerically precise analysis is computationally expensive (even when optimized using marching squares as 2D version of marching cubes [Lor87]), here we want to provide a quick preview method by implementing an approximation of contour lines as part of the surface shading process. A lookup table (LUT) containing ten entries for bathymetric and ten entries for hypsometric color values is sufficient to provide a coarse overview of the data set. The intended purpose of contour lines is to complement the colorization by providing more detailed elevation information. While they may be implemented via a very large, high-resolution LUT, the availability of programmable GPUs allows implementing them procedurally. Thus, the achievable resolution is only limited by numerical precision. The basic idea is to make use of the modulo function $h \bmod \lambda := h - floor(h/\lambda)$ for a given height h. Depending on a level height (e.g. 1m, 10m, 100m) and a given finite thickness δ of the contour line, it determines whether the hypsometric tint from the LUT or a constant color for the contour line, e.g. black or red, is used:

$$color(h) = \begin{cases} Contour, & h \bmod \lambda < \delta \\ LUT(h), & h \bmod \lambda \geq \delta \end{cases}$$

Using this modulo function it is straightforward to implement multiple such contour lines at the same time using different parameters. A set of five types of contour lines is easy to interpret visually as it is familiar from cartographic displays. These may be placed as red thick lines at levels of 10 (0m, 10m, 20m, …), thick black lines shifted by 5 (5m, 15m, 25m, …), red thin lines at levels of 1 (1m, 2m, 3m, 4m, 6m, 7m, ..), black thin lines shifted by 0.5 (0.5m, 1.5m, 2.5m, 3.5m, 4.5m, …) and finally gray lines at distance levels of 0.1 (0.1m, 0.2m, 0.3m, 0.4m, …). This scheme allows reading off a height range of 1:100 quickly and even 1:1000 if we consider a line width of 1cm, as demonstrated in Figure 1. While this approach is pretty straightforward, there are several aspects that can be, or need to be improved.

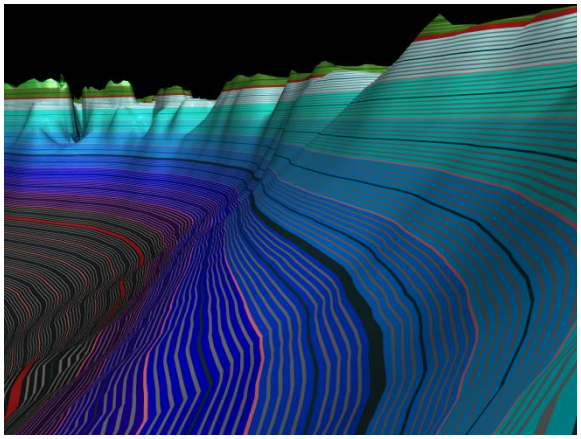


Figure 1: Contour lines as integrated part of surface colorization, utilizing five categories of contour levels.

Line Smoothing

A direct implementation of the modulo function leads to strong aliasing (moiré) effects, which disturbs the visual appearance since for each pixel there will be a binary yes/no decision on whether it belongs to a contour line or not. We can address this problem by replacing the binary decision *color(h)* with a smooth transition between the contour line and the ground color by using a hermite spline interpolation as in

$$S(x_0,\ x_1\ , x) = \begin{cases} 0, & x \leq x_0 \\ t^2(3-2t), & x_0 < x < x_1 \\ 1, & x \geq x_1 \end{cases}$$

where $t = (x - x_0)\ /\ (x_1 - x_0)$. Such a function is provided by the GLSL smoothstep() call.

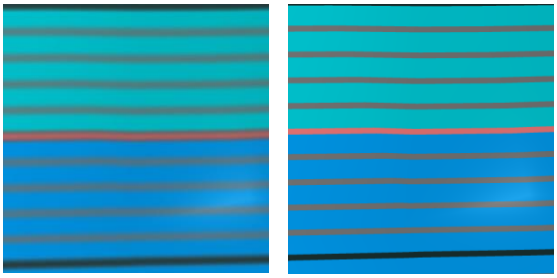


Figure 2: Line smoothing to address aliasing.

We then introduce a smoothing distance ε, which can be set to be $\varepsilon = \delta/4$, for instance, and compute a blending weight $W=B_s\ T_s$ as product of a bottom weight B_s and top weight T_s. The bottom part of the contour line is calculated as

$$B_s = S(0,\ \varepsilon,\ (h+\delta)\ \bmod \lambda) \quad .$$

The top part is calculated via

$$T_s = (\ 1 - S(\delta - \varepsilon/2,\ \delta + \varepsilon/2,\ (h+\delta)\ \bmod \lambda)\ ) \quad .$$

Both functions smooth the line's edges by ε. We need to consider that the line shall be centered precisely at height h, extending by δ upwards and

 ISBN 978-80-86943-74-9

downwards, smoothing out over a distance ε. The blending weight W is 1 for h being an integer multiple of λ such that the final color is given as

$$color(h) = (1-W) \cdot LUT(h) + W \cdot Contour \quad .$$

Therefore, by using the above formulae, the edges of the contour lines are smoothened, implementing antialiasing.

Gradient Compensation

As contour lines are implemented in this approach by identifying a set of points within the same height range, we will naturally find more points fitting this criterion in shallow regions, resulting in a broadening of the lines depending on the terrain's slope. The problem is inherent by design because a contour line becomes undefined in a completely flat area. While this broadening of lines is visually displeasing as compared to the computation of an explicit contour line, it may also be considered as a depiction of the contour lines becoming numerically unstable in flat areas. This would not necessarily be evident from an explicit contour line. Therefore, this effect cannot be fully cured. But, it can be compensated to some extent by taking into account the slope of the terrain and introducing a dependency of the contour line's thickness δ on the surface normal vector n ($|n|=1$) via

$$\delta = \delta_0 \, (1 - {n_z}^2)^g \quad .$$

Here, g is an exponent allowing fine-tuning the 'gradient compensation'. "Mathematically correct" would be an exponential of 0.5, however in practice the ability of over-compensation turns out to be beneficial, as depicted in Figure 3.

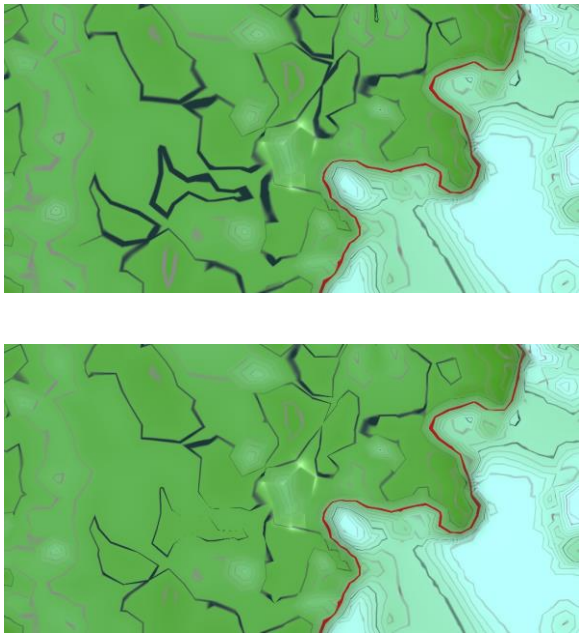


Figure 3 Procedural contour lines in a shallow wetland area, as-is and using a 'gradient compensation' exponent of 16 to narrow lines.

Visibility Fading

Contour lines of constant thickness as in Figure 4 come with the disadvantage that a thickness value working pleasingly well in the foreground leads to invisible or strongly aliased lines at further distances. In turn, a thickness value providing good background appearance looks ugly in the foreground. We thus relate both the blending weight W as well as the contour line thickness δ to the apparent size of the contour line on the screen, such to have lines in the distance being broader, but less prominent, and lines close to the observer being smaller but crisper. However, we cannot simply introduce a dependency of the surface point on the observer's location because this would fail in the case of an orthographic camera projection. Instead, we need to compute a visibility measure that is related directly to the apparent size of the contour line on the screen.

We proceed as follows: given the normal vector $\boldsymbol{n}$ of the surface, we can get a vector $\boldsymbol{t}$ tangential to the contour line by taking the cross product with the 'up' vector $\boldsymbol{z}=(0,0,1)$ as $\boldsymbol{t} = \boldsymbol{n} \times \boldsymbol{z}$. The tangential vector $\boldsymbol{t}$ will always be parallel to the horizontal plane, i.e. $\boldsymbol{t}_z=0$. However, this expression also applies to a general underground, such as the curvature of Earth, by using an arbitrary local 'up'-vector z. From this tangential vector we compute a vector $\boldsymbol{s}$ parallel to the surface but orthogonal to the tangent via $\boldsymbol{s} = \boldsymbol{n} \times \boldsymbol{t}$. Now, given a point on the surface P we can compute the location Q of the upper boundary of our contour line as $Q = P + \delta \boldsymbol{s}$. Both Q and P, given in world coordinates, are then projected in screen coordinates my multiplying in 4D with the OpenGL projection and modelview matrix $\mathbf{M}$, yielding 4D homogenous screen coordinates ${}^sP=M\,P$, ${}^sQ=MQ$. 3D affine screen coordinates are computed via dividing by the 4th component: ${}^aP=({}^sP_x/\,{}^sP_w\,,\,{}^sP_y/\,{}^sP_w\,,\,{}^sP_z/\,{}^sP_w)$, ${}^aQ=({}^sQ_x/\,{}^sQ_w\,,\,{}^sQ_y/\,{}^sQ_w\,,\,{}^sQ_z/\,{}^sQ_w)$. We are only interested in the x and y components of these projected points as they are directly related to the size of the contour line in pixels. To get the actual number of pixels we would need to multiply x and y with the width and height of the viewport in pixels. However, we want a pixel-resolution independent algorithm and thus only consider normalized screen coordinates. We define a visibility factor V as

$$V = V_0 \sqrt{\left({}^aP_x - {}^aQ_x\right)^2 + \left({}^aP_y - {}^aQ_y\right)^2} \quad ,$$

based on some adjustment factor V_0 to fine-tune the visual appearance. The visibility factor will be small at large distances of the contour lines from the screen as well as the contour-line being viewed tangentially, which would result in a visibly small line as well. To compensate for aliasing effects and enhance distant lines, we increase the line width via dividing by the

 ISBN 978-80-86943-74-9

visibility factor while at the same time reducing its blending weight:

$$\delta_{eff} = \delta / V \quad , \quad W_{eff} = W \cdot V \quad .$$

Using this effective line thickness δ_{eff} and blending factor W_{eff}, we achieve visually pleasing crisp lines in the foreground, fading out into bigger lines emphasizing coarser structures in the background, as demonstrated vs. Figure 5.

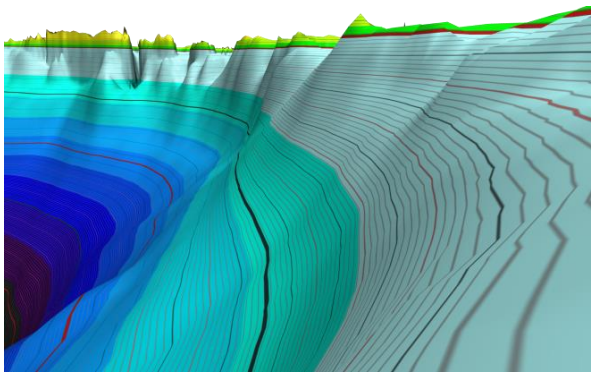


Figure 4 Contour lines with no distance dependency, leading to overly thick foreground lines while distant lines remain hardly visible.

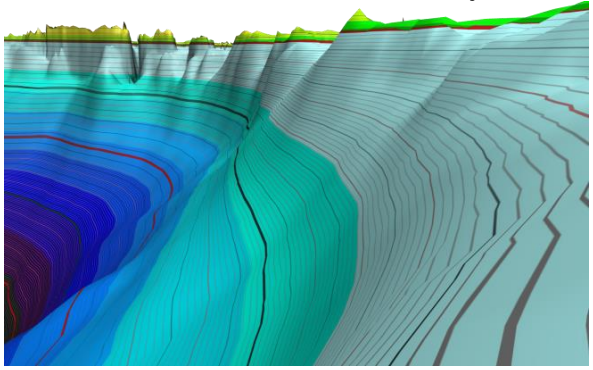


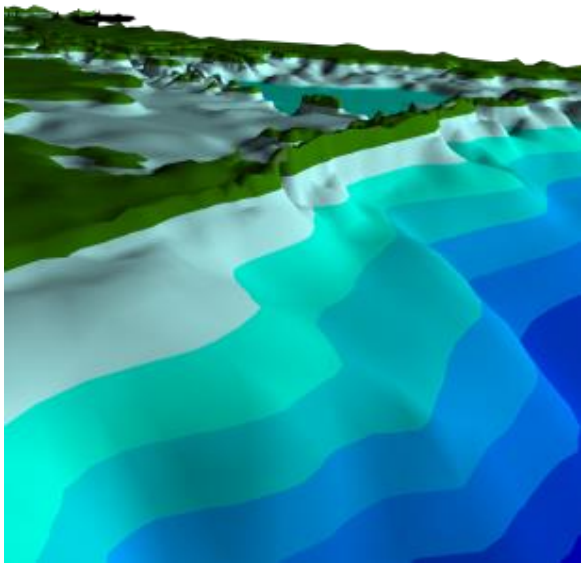
Figure 5 Contour lines with visibility fading, enhancing details crisper in the foreground as well as in background.

Non-linear Hypsometric Tints

Classically each entry in the color lookup table covers the same height range to allow easy identification of elevation by its color. A non-equidistant mapping in contrast provides the advantage of allowing emphasizing finer variations in elevation while covering the same height range. Whereas, the combination with contour lines still allows quantitatively precise read-off. One pretty simple choice of a non-linear mapping of height to colors is to utilize the arctangent function, as it maps an infinite range$[-\infty, +\infty]$ of input values to the range $[-\pi/2, +\pi/2]$. Mapping this range to the interval [0,1] for indexing a LUT is straightforward. We can further introduce a power function to parameterize the mapping of height h for additional adjustment:

$$C_l = \frac{1}{2}\left[\left(\frac{arctan(h)}{\pi/2}\right)^{\gamma} + 1\right]$$

where $C_l \in [0,1]$ is the LUT indexing value and $0 < \gamma < \infty$ is a tunable exponent. Values smaller than 1 will compress color equipotential lines around the sea level $h=0$ and enhance low-elevation details, as depicted in Figure 1.



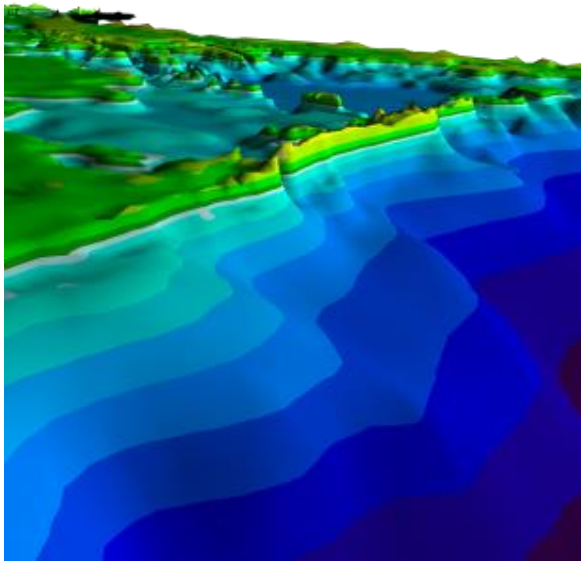


Figure 6 : Linear and non-linear colorization.

Shininess Variations

Modification of the shininess parameter as part Phong shading allows to view the surface and the below the sea level of the grid differently.

 ISBN 978-80-86943-74-9

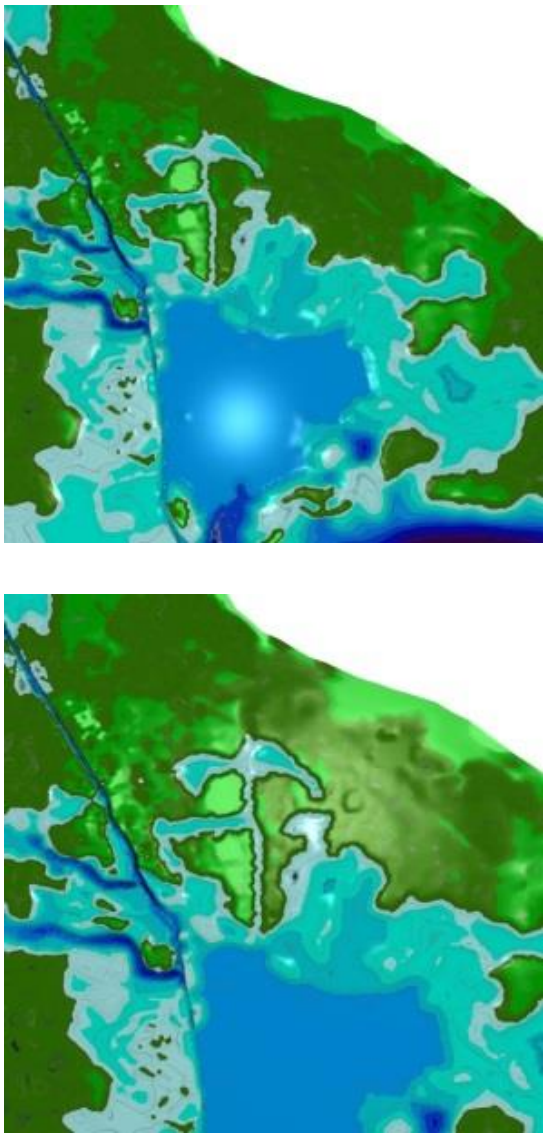


Figure 7: Variation of surface reflectivity (shininess) applied to water bodies within a shallow wetland area.

3. APPLICATION

Icosahedron

We use an icosahedron as a very simple example of a triangular mesh to validate aspects of our technique. Here, the height values range from -1.618 to +1.618. Taking this range into consideration, a set of contour lines is shown. Basically, three red thick lines are shown representing elevation levels of -1, 0, +1.

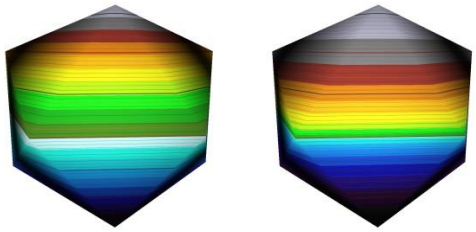


Figure 8: Linear and non-linear colorization without smoothening effect.

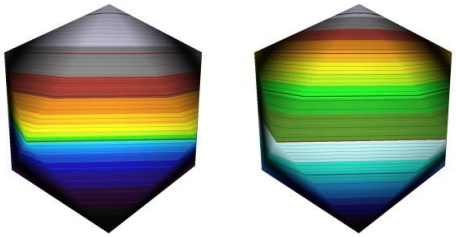


Figure 9: Non-Linear colorization with power less than one and greater than one

Barataria Bay

Barataria Bay is a 120 km long estuary located in the north-central Gulf of Mexico, just to the west of the Mississippi River Delta (Figure 10). It is bounded by the Mississippi River to the east and by a former channel of the Mississippi River, Bayou Lafourche, to the west. The estuary is connected to the Gulf of Mexico through four tidal passes and contains several large lakes and numerous marshes interconnected by ponds and channels. The average depth is only about 2 m but there are several deeper channels (e.g., ~ 4 m deep Barataria Bay Waterway) and tidal inlets (e.g., ~ 20 m deep Barataria Pass).

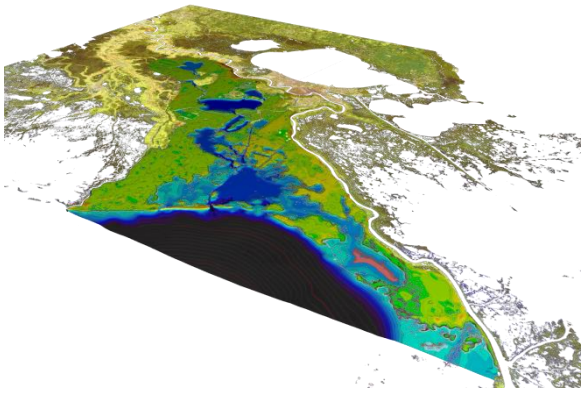


Figure 10 Barataria Bay estuary: Mississippi river is at right, New Orleans just above the center.

Here we explore a triangular mesh consisting out of 63190 vertices and 125038 triangle elements, covering a total area of 3140km^2, out of which 2950km^2 is open water. The grid resolution ranges from 18m in the channels to 1600m over wetlands and open water bodies. This mesh carrying precise

 ISBN 978-80-86943-74-9

bathymetric information is the basis for several numerical hydrodynamic models [Ino08, Das10, Das11, Das12].

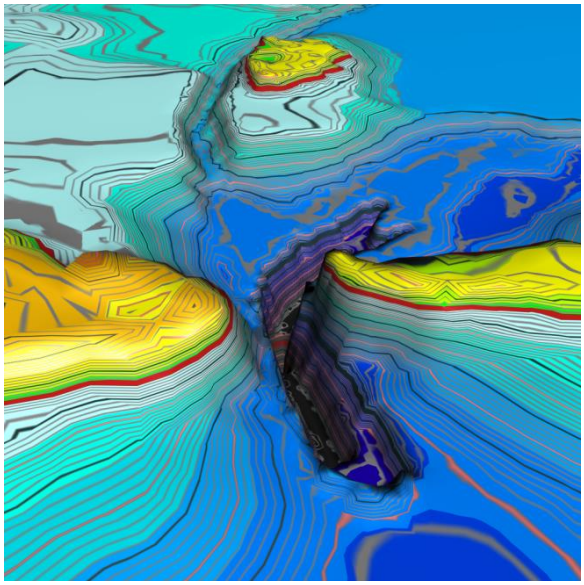


Figure 11 Barataria Pass

The maximum and minimum elevations are at 8.2m and -22.4m, respectively. Our contour line algorithm automatically places contour lines at the most appropriate power of 10 levels, therefore placing three thick red lines at levels of 0.0m, -10m and -20m. Thin red lines are placed at 1m levels, alternating with thin black lines offset by 0.5m. Gray lines represent elevation levels of 10cm.

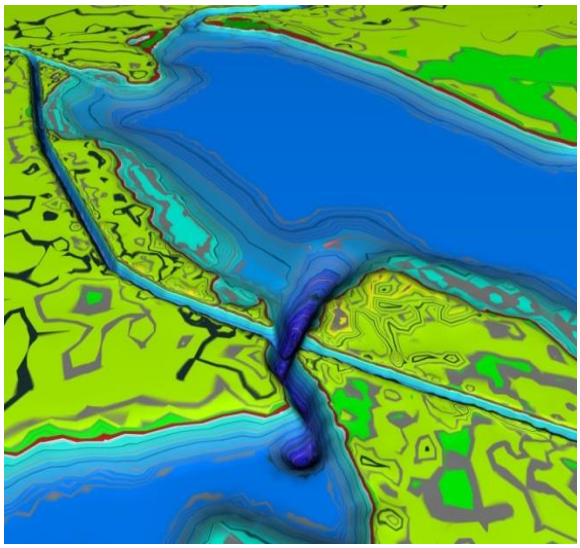


Figure 12 Lake Salvador outlet

Important features in this numerical mesh such as Barataria Pass, Figure 11, or Lake Salvador outlet, Figure 12, are clearly emphasized when applying our cartographic 3D rendering. With proper vertical scaling, Barataria Bay Waterway that carries a large portion of the water flow in the area is represented more prominently when compared with a conventional 2D cartographic view.

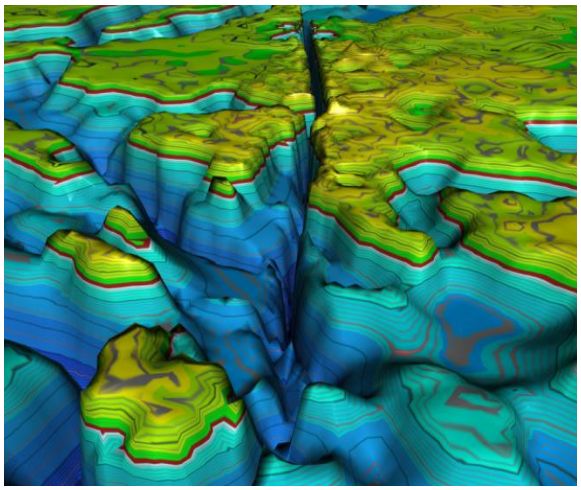


Figure 13 "Grand Canyon of Barataria Bay" - Barataria Bay Waterway.

Optionally, we may also just display the contour lines on their own by rendering the surface itself transparent but only the contour lines opaque. This effect on the Barataria Bay waterway is demonstrated in Figure 14, resulting in a fine detailed depiction of contour levels that are automatically adjusting to view distance.

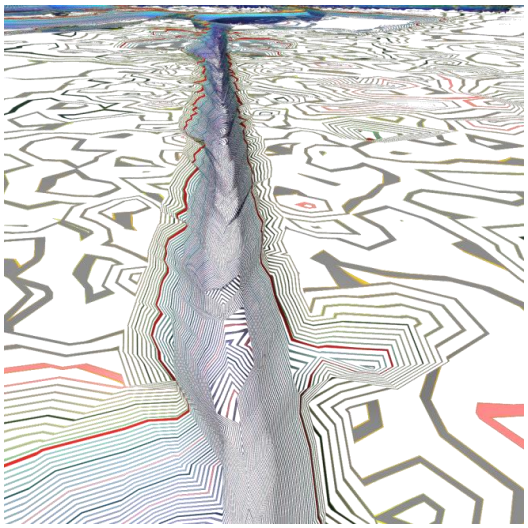


Figure 14 Barataria Bay Waterway displaying procedural contour lines as derived from the numerical triangular mesh.

Rheinfelden LIDAR Terrain

The “Rheinfelden” data set has been derived from an airborne light detecting and ranging (LIDAR) laser scan using a bathymetric green laser system, the Riegl hydro-graphic laser scanner VQ-820G [Stei10]. This laser system is able to penetrate a water surface to provide bathymetric data of shallow water grounds. Here, a river section around a power plant in Rheinfelden, Germany, was acquired. The scan

 ISBN 978-80-86943-74-9

was complemented by additional sonar measurements. We computed a digital terrain model (DGM) and a digital water model (DWM), resulting in a mesh consisting of 1880352 triangle elements with 944521 vertex nodes.

The 'sea level' of this data set has been set as 266.5m to correspond with the height of the water surface at power plant's location. Figure 15 and Figure 16 provide views along the river using identical visualization parameters.

For comparison, contour lines were computed using a CPU marching squares algorithm, a simpler 2D variant of the 3D marching cubes [Lor87]. For each elevation meter a contour line was created in the Rheinfelden dataset, operating on the raw data representation as a point cloud (1.0m resolution). The computation time of the serial code was 12.5 seconds Intel-i7-2670QM@ 2.20GHz. Using parallelization via OpenMP, 8 threads on 8 CPUS reduced the computation time to 3.5seconds. In contrast, using an NVIDIA Quadro3000 on the same system, the triangular surface with procedural contour lines renders at 9.0 frames per second (fps) or 0.111 seconds. This results in a speed up of 32.1 in comparison to the OpenMP parallel CPU code for visual scene exploration or rendering of contour plots. If contour lines are required as explicit geometry for further processing, the shader-based procedural contour lines cannot replace the CPU algorithm.

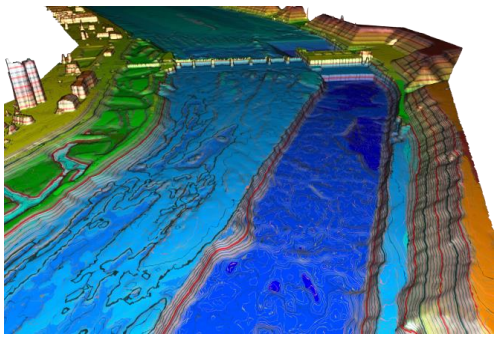


Figure 15 LIDAR-acquired bathymetry of a river power plant in Rheinfelden.

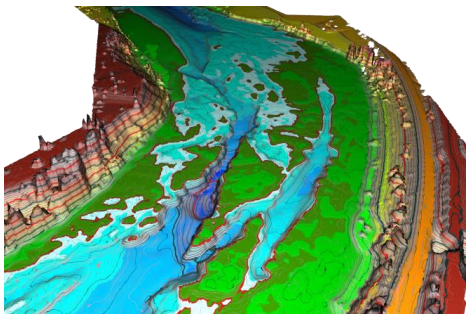


Figure 16 Bathymetry of a river bend at the Rheinfelden dataset, exposing some canyons.

Baltic LIDAR Terrain

Another LIDAR dataset was acquired by the same airborne laser scan system as used for the Rheinfelden area. The raw dataset provides a point cloud of 385 million elements. It was reduced to a triangular surface representing the ground surface (DGM) and a second triangular surface representing the water surface (DWM). The DGM has 1659958 vertices with 3228653 triangle elements and the DWM 2670748 vertices with 5271291 triangle elements. Rendering of the DGM results in a frame rate of 6.3 fps, rendering of both, DGM and DWM, in 2.9 fps, measured on an Intel-Xeon-X5650@2.67Ghz(x6) NVIDIA-Quadro5000 system.

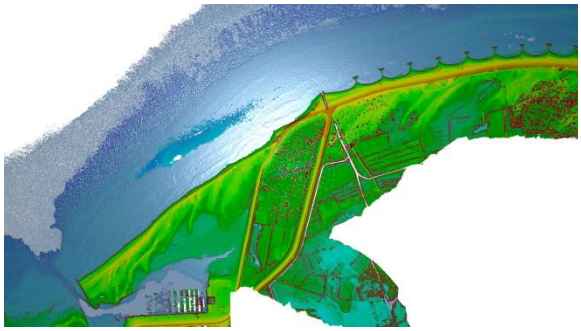


Figure 17: Variation in surface reflectivity. The coast line is illustrated by the border of the water surface extracted from the classified point cloud.

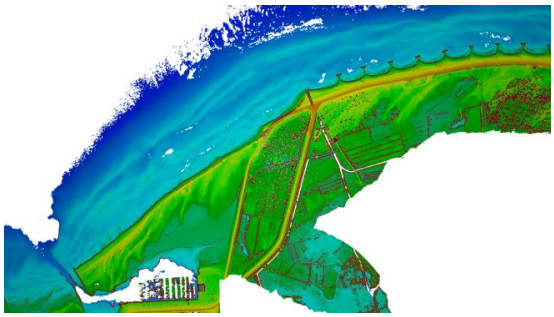


Figure 18: Highly sensitive hypsometric tinting illustrates structures on the ocean ground. The coast line matches the coast line of Figure 17.

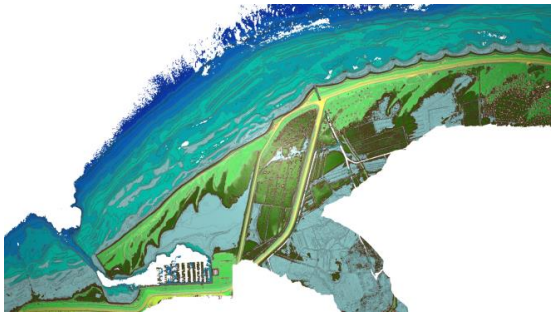


Figure 19: Procedural contour lines emphasize the sand dunes on the ocean ground. Sand dunes have a length of about 50 to 800 meters.

 ISBN 978-80-86943-74-9

Figure 17 demonstrates the effectiveness of varying the surface reflectivity to emphasize sea level. It shows how the coastline, illustrated by the shader (opaque surface), matches the coast line that was generated from laser echoes which had been classified as water surface (transparent surface). This is, for example, well-illustrated in the harbor region. Figure 18 provides an alternative view clearly depicting sub-surface structures on the ocean ground, such as sand dunes or the excavated harbor entrance. Figure 19 complements the colorization scheme with procedural contour lines. This scheme emphasizes the sand dune structures on the ocean ground.

4. PERFORMANCE

We found the impact of including procedural contour line in addition to usual color-coding of surfaces by height information to be negligible. For the Barataria Bay the rendering frame rate was found to be at most four percent slower when compared to rendering without contour lines. For the larger Rheinfelden dataset, the rendering frame rate of the grid was only 0.3 percent slower with contour lines than without. Table 1 summarizes our measurements. Since the rendering is view-dependent, the view point and direction influences the rendering time, so the compared views from the top, the front and the side. The values for the rendering frame rate are averaged from repeated execution of the actual OpenGL rendering call.

Name of the dataset	View	Frame rate with contour lines	Frame rate without contour lines	Relative increase of rendering frame rate
Baratar ia Bay	Top	341.764	357.782	4.4%
	Front	344.94	345.423	0.13%
	Side	346.86	359.066	3.3%
Rhein-felden	Top	17.139	17.1701	0.17%
	Front	16.983	17.0394	0.32%
	Side	17.149	17.1859	0.215%

Table 1: Rendering time performance of the grid with contour lines and without contour lines

To calculate the performance, the graphics processor used was a Quadro FX5600 with memory 1536 MB. Also, eight CPU's with model name Intel® Xeon® CPU with memory 32GB was used.

These measurements merely refer to the performance of the shader. The triangular surfaces have been rendered as-is, without dynamic triangle refinement and level-of-detail. Those mechanisms, common from terrain rendering in particular when rendering height data given on a regular grid, are orthogonal to our shading technique and can be used to complement the overall performance, but are not part of our algorithm.

5. SUMMARY

In this article we presented a highly efficient method to render a height field given on a triangular mesh in a manner similar to well-known cartographic views, enhanced by essential or useful features such as line smoothing, gradient compensation, visibility fading, non-linear elevation mapping and shininess variation. The method can be implemented fairly simple on modern graphics cards to yield real-time renderings allowing precise quantitative assessment of data values. The effectiveness of this method has been demonstrated on a diverse set of geoscientific datasets from current research projects.

6. FUTURE WORK

It has been noted [Pat12] that the use of hypsometric tints may be misleading when representing terrains. E.g., low areas will always appear 'lush' even if they refer to a desert. Similarly open water above sea level will look as if it were on a land mass. This problem may be addressed by using cross-colorization, which is modifying the colors used in the LUT based on an additional scalar data field given on the terrain. Doing such requires a two-dimensional LUT and faces the challenge of non-repeating color entries to keep the rendering meaningful. In lieu or additionally, incorporating blending with RGB information from aerial photography may be useful as well. Also, alternative non-linear functions, in particular mappings based on histograms [Khu12], may be used to emphasize local details even better. The visual quality may well be enhanced by adding a geometry or tessellation shader to insert more triangles where needed, based on high order interpolation methods (i.e., dynamic level of detail). However, from a scientist's perspective to accurately identify their data elements, such an enhancement might not necessarily be desirable. Finally, the technique presented here is neither limited to elevation data nor to surfaces. All methods apply as well to a general scalar field replacing the elevation. Such may be given on a set of lines or non-planar surfaces as well, once the gradient of the scalar field and the base domain is known to allow for gradient compensation and visibility fading.

 ISBN 978-80-86943-74-9

7. ACKNOWLEDGMENTS

This study was funded in part by BP through the grants to Louisiana State University and Coastal Waters Consortium. Special thanks to Frank Steinbacher for providing the LIDAR datasets. Thanks to Wolfgang Dobler for providing time measurements on the data set Rheinfelden. This work was supported by the Austrian Science Foundation FWF DK+ project Computational Interdisciplinary Modeling (W1227), and grant P19300. This research employed resources of the Center for Computation and Technology at Louisiana State University, which is supported by funding from the Louisiana legislatures Information Technology Initiative.

8. Additional Authors

Marcel Ritter, University of Innsbruck & Airborne Hydromapping OG, Technikerstrasse 13&21 A-6020, Innsbruck, Austria; Dubravko Justic, Department of Oceanography and Coastal Sciences, School of the Coast and Environment, 2221 Energy Coast and Environment, Louisiana State University, Baton Rouge, LA 70803, djusti1@lsu.edu; Lixia Wang, Department of Oceanography and Coastal Sciences, School of the Coast and Environment, 2223 Energy Coast and Environment, Louisiana State University, Baton Rouge, LA 70803, lxwang@lsu.edu

9. REFERENCES

[Gul10] Gul, S.; Khan, M.F., "Automatic Extraction of Contour Lines from Topographic Maps," Digital Image Computing: Techniques and Applications (DICTA), 2010 International Conference on , vol., no., pp.593,598, 1-3 Dec. 2010

[Du04] Du Jinyang; Zhang Yumei, "Automatic extraction of contour lines from scanned topographic map," Geoscience and Remote Sensing Symposium, 2004. IGARSS '04. Proceedings. 2004 IEEE International , vol.5, no., pp.2886,2888 vol.5, 20-24 Sept. 2004

[Gro10] Grochow, K.; Stoermer, M.; Fogarty, J.; Lee, C.; Howe, B.; Lazowska, E., "COVE: A Visual Environment for Multidisciplinary Ocean Science Collaboration," e-Science (e-Science), 2010 IEEE Sixth International Conference on , vol., no., pp.269,276, 7-10 Dec. 2010

[Stei10] F. Steinbacher, M. Pfennigbauer, A. Ulrich, and M. Aufleger, "Vermessung der Gewässersohle - aus der Luft – durch das Wasser," in Wasserbau in Bewegung ... von der Statik zur Dynamik. Beiträge zum 15. Gemeinschaftssymposium der Wasserbau Institute TU München, TU Graz und ETH Zürich, 2010.

[Pat12] Patterson, T., & Jenny, B. (2012). The Development and Rationale of Cross-blended Hypsometric Tints. *Cartographic Perspectives, 0*(69), 31-46. Retrieved from http://www.cartographicperspectives.org/index.php/journal/article/view/cp69-patterson-jenny

[Ino08] Inoue, M., Park, D., Justic, D., Wiseman, W. J., Jr. 2008. A High-Resolution Integrated Hydrology-Hydrodynamic Model of the Barataria Basin System. Environmental Modelling and Software 23: 1122-1132.

[Das10] Das, A., Justic, D., Swenson, E. 2010. Modeling estuarine-shelf exchanges in a deltaic estuary: Implications for coastal carbon budgets and hypoxia. Ecological Modelling 221: 978-985.

[Das11] Das, A., Justic, D., Swenson, E., Turner, R. E., Inoue, M., Park., D. 2011. Coastal land loss and hypoxia: The 'outwelling' hypothesis revisited. Environmental Research Letters 6: 025001 (9 pp); doi:10.1088/1748-9326/6/2025001.

[Das12] Das, A., Justic, D., Inoue, M., Hoda, A., Huang, H., Park., D. 2012. Impact of Mississippi River diversions on salinity gradients in a deltaic Louisiana estuary: Ecological and management implications. Estuarine, Coastal and Shelf Science 111: 17-26.

[Vin03] Leonardo da Vinci, A map of central Italy, 1503, Royal Collection by 1690; available from http://www.royalcollection.org.uk/eGallery/object.asp?maker=12196&object=912277&row=436&detail=magnify

[Lor87] Lorensen, William and Harvey E. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. Computer Graphics (SIGGRAPH 87 Proceedings) 21(4) July 1987, p. 163-170)

[Khu12] S. Khurana, N. Brener, B. Karki, W. Benger, S. Roy, S. Acharya, M. Ritter, S. Iyengar, Multi Scale Color Coding of Fluid Flow Mixing Indicators along Integration Lines, WSCG2012, Plzen, 2012