Interpolants Induced by Marching Cases*

Hamish Carr[1] and Eoin Murphy[2]

1 **School of Computing and Informatics, University College Dublin**
Belfield, Dublin 4, Ireland
and School of Computing, University of Leeds
Woodhouse Lane, Leeds, LS2 9JT, United Kingdom
`h.carr@leeds.ac.uk`

2 **School of Computing and Informatics, University College Dublin**
Belfield, Dublin 4, Ireland
`eoinomurchu@gmail.com`

Abstract

Visualization depends among other things on the interpolant used in generating images. One way to assess this is to construct case tables for Marching Cubes that represent the chosen interpolant accuracy. Instead, we show how to construct the interpolants induced by Marching Cases for comparison and assessment, how to extend this approach to Marching Squares, Cubes and Hypercubes, and how to construct an interpolant which is computationally equivalent to the digital rules conventionally used in image processing. Furthermore, we demonstrate that unlike tetrahedral meshes, geometric measurements over multi-linear mesh cells are inherently non-linear and cannot be summed as in the Contour Spectrum.

1998 ACM Subject Classification G.1.1, I.3.5

Keywords and phrases Interpolation, Marching Cubes, Isosurfaces

Digital Object Identifier 10.4230/DFU.Vol2.SciViz.2011.48

1 Introduction

In scientific and medical visualization, properties such as object volume and surface area are commonly computed for individual isosurfaces. Since the introduction of the Contour Spectrum [1], these properties have been computed for entire families of isosurfaces and used either as part of a visual interface [1], for automated analysis of significant features in the data [16], and as a measure of importance for topological simplification [6].

However, while geometric properties for simplicial meshes are generally piecewise-polynomial splines [1], computations on non-simplicial meshes have used discrete approximations instead [6]. Recent work [3, 15] has however shown that discrete approximations such as histograms or other image statistics are a rather poor approximation of the underlying geometric properties, a problem which is exacerbated by aliasing for small features.

Given this, it is natural to ask whether piecewise polynomial splines exist for geometric properties of non-simplicial meshes, and in particular of rectilinear meshes. This in turn requires examining both multilinear interpolants and the conventional Marching Cubes cases that are commonly used for extracting isosurfaces.

It has also been known for some time that the standard Marching Cubes cases do not correspond exactly to the trilinear interpolant [8, 10, 14, 13]. It is therefore also natural to ask whether there is an interpolant whose contours are identical to these cases.

* This work was partially supported by Science Foundation Ireland.

Scientific Visualization: Interactions, Features, Metaphors. *Dagstuhl Follow-Ups, Vol. 2.*
Editor: Hans Hagen; pp. 48–58
DAGSTUHL FOLLOW-UPS Dagstuhl Publishing
Schloss Dagstuhl – Leibniz Zentrum für, Germany

This paper will demonstrate that piecewise-polynomial splines do not in general exist for multilinear interpolants or for Marching Cubes cases. In the process, we will also construct the interpolant corresponding to Marching Squares cases with a construction that extends to Marching Cubes cases, and demonstrate that these interpolants have undesirable qualities.

2 Previous Work

Isosurfaces are formally defined as the inverse image $f^{-1}(h)$ of a continuous function $f : \mathbb{R}^3 \to \mathbb{R}$ for a particular *isovalue* h, and are a core technique in scientific and medical visualization. In practice, isosurfaces are approximated using variations on Marching Cubes [9], which generates a triangulated separating surface between those samples known to be inside the isosurface and those samples known to be outside the isosurface.

More generally, marching cases can also be used to extract contours in 2D (isolines), 3D (isosurfaces) or higher (isohypersurfaces), using any type of mesh, whether simplicial (triangles in 2D, tetrahedra in 3D, pentatopes in 4D, &c.), rectilinear (squares or rectangles in 2D, cubes or hexahedra in 3D, hypercubes or hyperblocks in higher dimensions) or other cell types. Much of the literature on Marching Cubes has recently been surveyed by Newman and Yi [11].

While the intent of Marching Cubes was to generate surfaces that matched the trilinear interpolant on cubes, it was soon discovered that the mathematically correct bilinear and trilinear contours were hyperbolic in nature [14]. As a result, the standard Marching Cubes cases are only approximations, and subsequent work [13] has developed more complex tables that capture the topology of the isosurfaces in triangulated approximations.

Statistical properties such as histograms have long been used in user interfaces. Bajaj, Pascucci & Schikore introduced the Contour Spectrum [1], where geometric properties such as contour length (in 2D), area (in 3D), enclosed area (in 2D) and enclosed volume (in 3D) were computed for all isovalues and displayed in a separate panel to help a user identify significant isovalues. This work also identified that for simplicial meshes these properties are piecewise-polynomial splines that can be computed efficiently, as we will see in Section 3.

Subsequent work [6] showed that these properties can be computed for single contours during the construction of the contour tree, and used for topological simplification. The authors also observed in passing that these properties were not piecewise-polynomial for cubic meshes, instead using discrete approximations such as sample counts and isovalue summations.

Although this approach was necessary at the time, it has since been shown [3, 15] that discrete approximations (including histograms) effectively assume the nearest-neighbour interpolant, and as a result give poor approximations of these geometric properties, especially for small objects or at low sampling resolution.

We therefore return to the question of whether it is possible to find closed form solutions for non-simplicial meshes such as rectilinear meshes, either for multilinear interpolants or for marching cases. We will also show that an interpolant can be constructed that is equivalent to the marching cases, in the sense that its contours are identical to those extracted by the marching cases.

3 Spline Linearity

In the Contour Spectrum [1], the observation was made that for simplicial meshes, geometric properties took on the form of piecewise-polynomial splines. As this is the point of departure

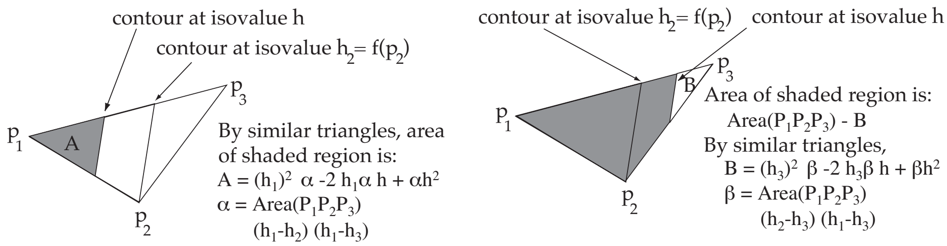

Figure 1 Computing Area in a Linear Triangle.

for the balance of this paper, it is worthwhile reviewing the construction, which we will perform in 2D with linear contours in triangles for simplicity. As is conventional, we shall assume that the isovalues h_1, h_2, h_3 at the vertices p_1, p_2, p_3 of the triangle are distinct, i.e. that $h_1 < h_2 < h_3$.

As we can see in Figure 1, each contour of a linear interpolant on a simplicial mesh is simply a line. And, as each contour is defined by a single isovalue h, the length of the contour will be a function $L(h)$, whose input is an isovalue, and whose output is the length of the corresponding contour. Similarly, the area behind the contour will also be a function $A(h)$

Now, we know that the contours are the intersection of a horizontal plane with the graph of the function in $\mathbb{R}^3$, so it follows that all contours are parallel to each other, and in particular are parallel to the contour at isovalue h_2 that passes through p_2. Since this contour is fixed and has a known length and area associated, it follows that $L(h)$ and $A(h)$ can be computed using similar triangles, depending on which half of the triangle the contour is in. Because similar triangles have linear relationships, it then follows that properties such as length and area are polynomials in h, which allows easy tracking by summation of coefficients.

In a simplicial (tetrahedral) mesh in 3D, contours of the linear interpolant are planes, and the corresponding construction applies. In general, therefore, geometric properties in a simplicial mesh can be computed by laws of similar simplices, and generate piecewise-polynomial properties.

4 Bilinear Non-linearity

As we have just seen, geometric properties of isosurfaces on a simplicial mesh are polynomials, which are easy to track and combine. Unfortunately, the same is not in general true for higher-order interpolants. We will demonstrate this by examining geometric properties of the bilinear interpolant.

To do so, we start with the paradigm case of the area bounded by the contour at isovalue h. We know from Nielson & Hamann [14] that the contours of a bilinear cell are hyperbolae, and we show a single contour in a bilinear cell in Figure 2.

To compute the area of region A, we need to know the area B under the hyperbolic curve $y = \frac{k}{x-x_v} + y_h$, where x_v is the x coordinate of the vertical asymptote, y_h is the y coordinate of the horizontal asymptote, and k is a scaling constant. The area of B is then given by the integral:

$$\int_a^b \frac{k}{x - x_v} + y_h \, dx = k \ln |b - x_v| - k \ln |a - x_v| + (b - a) y_h \ . \tag{1}$$

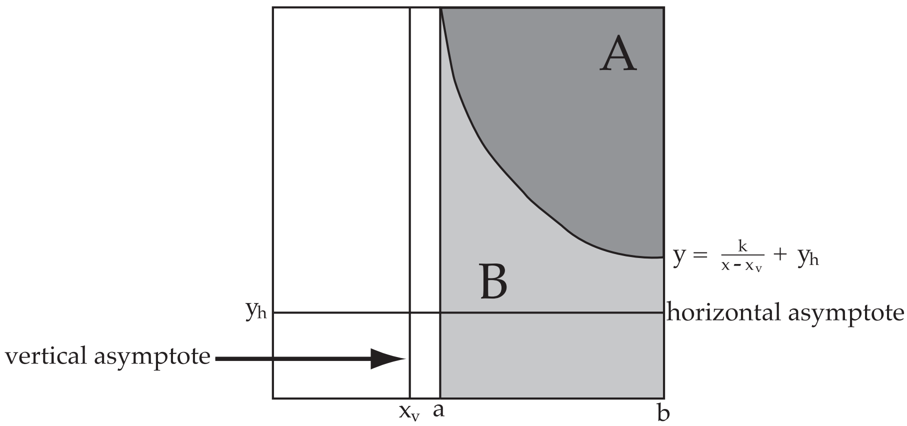

Figure 2 Computing Area in a Bilinear Cell. Since bilinear interpolants generate hyperbolic contours, computing local spatial measures involves logarithmic terms.

The appearance of a logarithmic term is problematic, because both the Contour Spectrum [1] and local geometric measures [6] rely on tracking the sum of polynomial terms with a single polynomial of $O(1)$ size.

It is possible on principle to sum logarithmic terms using the identity:

$$\log(a) + \log(b) = \log(ab) \ , \tag{2}$$

i.e. we can exponentiate each term and multiply them together, then take a logarithm any time we need it. For an extended sequence of log terms, however, we will have to worry about the numerical precision of performing a large number of multiplications.

The arclength of the contour is even worse, being defined by:

$$\begin{aligned} L(x) &= \int_a^b \sqrt{1 + (f'(x))^2} \\ &= \int_a^b \sqrt{1 + \left(\frac{-1}{x^2}\right)^2} \\ &= \int_a^b \sqrt{1 + \frac{1}{x^4}} \end{aligned}$$

for which no simple integration formula exists.

A further complication is that computing these formulae first requires finding the bounds of integration a, b. While this is feasible for the bilinear interpolant, for even the trilinear interpolant, it requires accurate analysis of the MC cases required to represent the trilinear interpolant correctly [12, 4].

Given the difficulties associated with non-polynomial function representations, analytically intractable integrals and hard-to-determine bounds of integration, it is natural to ask whether the task is any simpler if we simply use the known Marching Squares (or Cubes) cases.

5 Interpolant Induced by Marching Squares

Both for theoretical reasons and for practical reasons, therefore, it is desireable to identify what function, if any, corresponds to the contours generated by Marching Squares (Cubes) cases. Moreover, since the end points of the contour fragments are defined by linear interpolation,

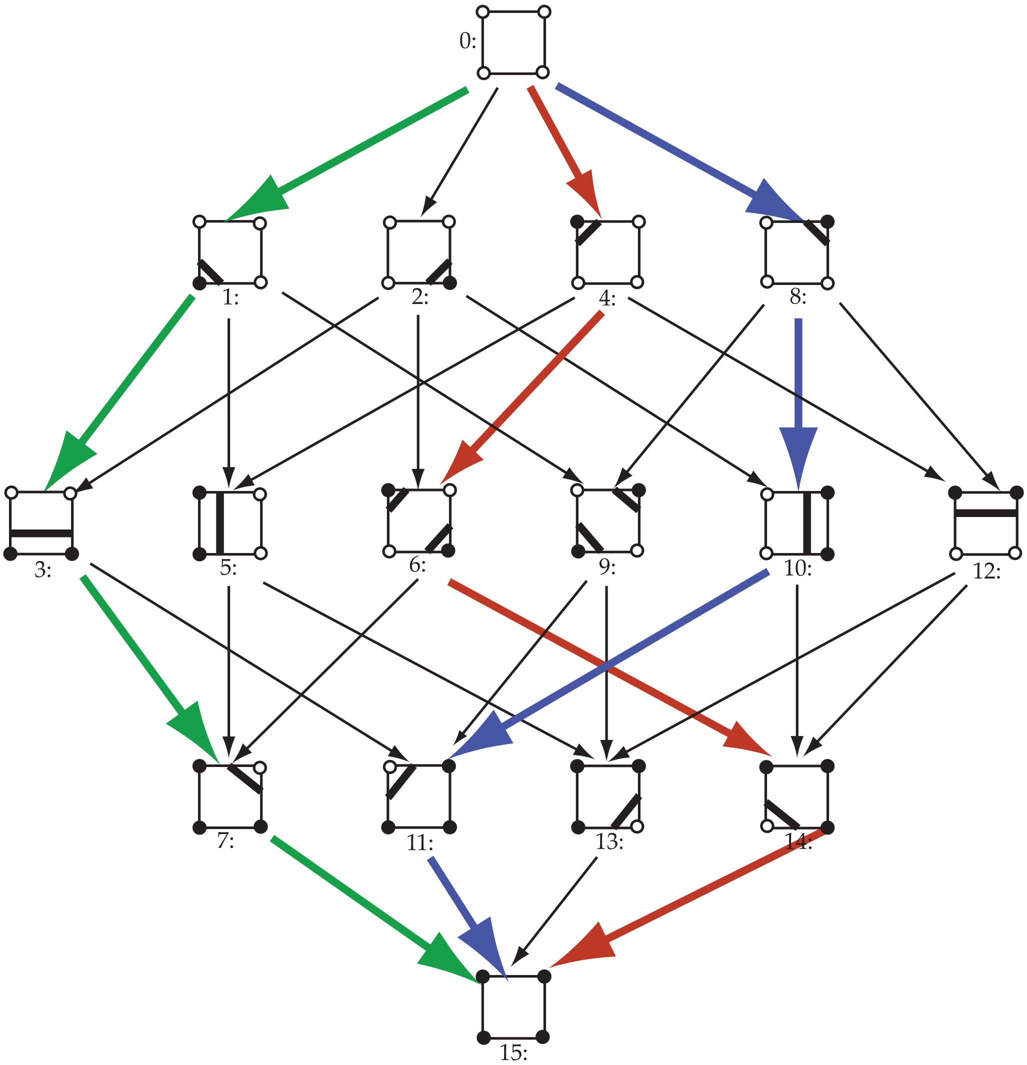

Figure 3 Subject to symmetry, only three distinct types of cell are possible, as shown by the coloured arrows.

we would like to know whether their geometric properties are polynomial splines as in the case of simplicial meshes.

We will see shortly that there is not in fact a single interpolant induced by Marching Squares. Instead, the interpolant depends on the set of Marching Squares cases swept through by the contour over all isovalues. Figure 3 illustrates the possibilities using a finite state machine to represent all possible sequences of cases [5].

In any given cell, if we sweep a contour from high isovalues to low, it will pass through a sequence of cases dictated only by the order in which the vertices are swept past (i.e. when they turn from white to black as the sweep passes their isovalue). Moreover, for the range of isovalues corresponding to the case, the points that define the contour fragments are linearly interpolated along the edges. Since the contours are linear combinations of these edge points, it follows that the contour sweeps out a continuous region of the cell, which can be represented as a surface, as in Figure 4. So, if each case sweeps out a surface over a range of isovalues, can the surfaces corresponding to different cases intersect? We will see in

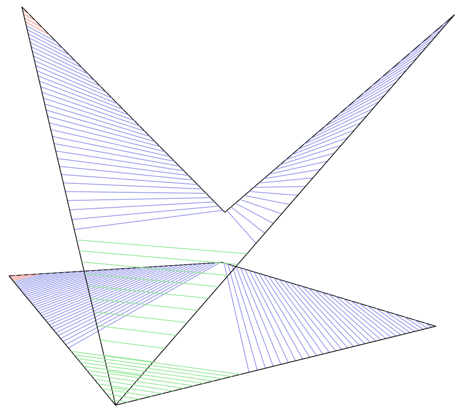

Figure 4 Illustration of Surface for Ambiguous Case with Projection to Ground Plane.

Section 7 that we can guarantee that they do not, but for now, we simply assume that they do not.

Consider an isovalued sweep through a particular cell. At high isovalues, all four vertices are white and the contour fails to intersect the cell (case 0). Some time later, one vertex becomes black, and the contour is in case $1, 2, 4$, or 8, which are rotationally symmetric to each other. If we follow the path marked in red in Figure 3, we sweep past vertex 2 to case 4, as illustrated. Thereafter, the next vertex to turn black can be diagonally opposite to the first (i.e. case 6), or adjacent to it (e.g. case 5 or 12). Again, following the red path, we sweep past vertex 1 to get case 14. Finally, the last vertex is swept to reach case 15.

Clearly, there are symmetries involved here, and several other paths equivalent to the red path. Up to these symmetries, there are three possible paths, marked in red, blue and green respectively in Figure 3. Having identify the set of possible paths, we now consider each type of path separately.

5.1 Ambiguous Face: The Red Path

Interestingly, the red path can only occur in a cell with an ambiguous face [14], as it requires diagonally opposing black and white vertices for case 6. While this might seem to be the hardest case to analyse, it is in fact the easiest, as can be seen in Figure 4.

This path uses cases $0, 4, 6, 14, 15$, of which 0 and 15 may be ignored as they have no contours. In case 4, the contour is defined by two edge points which move linearly, and thus sweep out a triangular surface until the isovalue of the opposite corner is reached. At this point, a separate contour starts at this new corner and similarly sweeps out a triangular surface, while the contour in the upper left corner continues to sweep out its triangular surface. Once the third vertex isovalue is reached however, these two contours disappear and a new one is swept towards the lower left corner. This is the exact opposite of case 2, and sweeps a final triangular surface.

Note however that this triangle only meets the previous ones at the edge points, and that a large region in the centre of the cell is never used by any contour. We can, however, set the isovalue of all points in this region to the vertex isovalue, forming a flat sheet that connects the three swept triangles. Thus, we have used the sweeping metaphor to construct a continuous function over the cell whose contours match those extracted using Marching

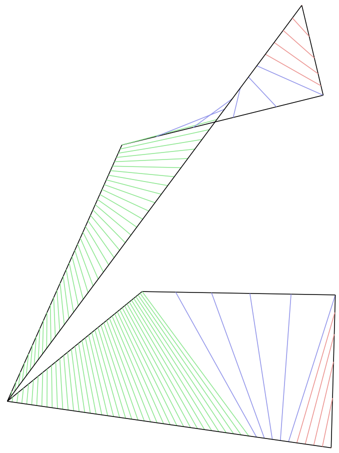

Figure 5 Illustration of Surface for Cyclic Case with Projection to Ground Plane.

Squares for this cell. Moreover, we note that this surface is composed of triangles, so that geometric properties will continue to be represented by tidy polynomials.

Unfortunately, however, this is not true for non-ambiguous faces, which we turn our attention to next. Moreover, the interpolant is data-dependent, and is not the same in every cell, which poses a problem for frequency domain analysis.

5.2 Cyclic Faces: The Blue Path

Our next possibility is the path shown in blue, where the vertices are swept in cyclic order around the face. Here, we start off with a single edge in case 8, and end up with a triangular surface in case 11. In between, however, we have one vertex moving along the top edge and one along the bottom edge. If we label these points $p(h)$ and $q(h)$, each linearly interpolated along the edge with respect to the isovalue h, we can derive the following formula for edge length:

$$\begin{aligned} L(h) &= \sqrt{\|p(h) - q(h)\|} \\ &= \sqrt{\left\| \left(v_{00} + \frac{h - h_{00}}{h_{01} - h_{00}} (v_{01} - v_{00}) \right) - \left(v_{10} + \frac{h - h_{10}}{h_{11} - h_{10}} (v_{11} - v_{10}) \right) \right\|} \end{aligned}$$

which is clearly non-polynomial.

5.3 Zigzag Faces: The Green Path

Finally, we deal with faces corresponding to the green path. Here, the vertices are swept in a zigzag fashion. As with the cyclic faces, we have triangles swept through when the edges defining the contour share a vertex, and an intermediate section with non-polynomial behaviour.

5.4 Asymptotic Decider

One additional possibility is to use the Asymptotic Decider [14] to adjudicate ambiguous faces. In this case, as shown in Figure 7, we end up with not three but four triangular sections induced by the actual contours, and a flat diamond in the middle connecting them.

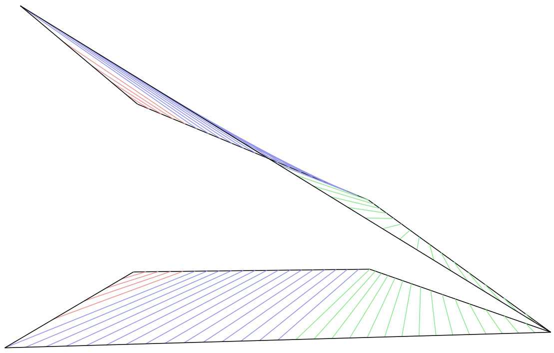

Figure 6 Illustration of Surface for Zigzag Case with Projection to Ground Plane.

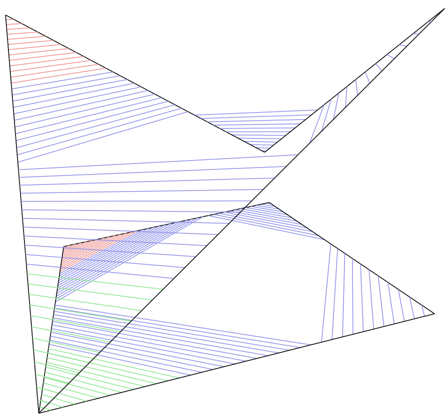

Figure 7 Illustration of Surface for Ambiguous Face with Asymptotic Decider.

5.5 Interpolant Properties

From the foregoing sections, we can see that the interpolant thus induced, while continuous, has several undesireable properties:

1. Piecewise: the interpolant is defined piecewise within each cell,
2. Non-Smooth: the interpolant is not smooth - i.e. it is only C^0 continuous
3. Non-Uniform: the details of the interpolant depend on the actual isovalues and their discrete relationships, rendering analysis based on signal processing difficult or impossible.

but, compared to the nearest neighbour (NN) interpolant, is at least piecewise continuous.

6 Interpolant Induced by Marching Cubes

Unsurprisingly, this approach can also be extended to Marching Cubes, following the same general principles:

1. There will be as many cases as there are distinct paths through the finite state machine. Note that while there will be reductions due to symmetry, it will not be true that the paths can simply be counted in a symmetry-reduced finite state machine. For example, consider our cyclic (blue) and zigzag (green) cases above. In both paths the sequence of symmetry reduced cases is: 0, 1, 2, $1C$, $0C$. But, as we can see from the corresponding surfaces, the details of the interpolants differ.
2. Each marching case in the path through the finite state machine will correspond to a patch of the interpolant, as will the isovalues at which the cases change.
3. For each marching case, the corresponding range of isovalues will generate a sequence of smoothly varying contours.
4. For each surface fragment, if the edges used to interpolate share a common vertex, the fragment will move linearly, and sweep out a barycentric interpolant over a simplex. Such fragments will have polynomial spline properties as in the Contour Spectrum [1].
5. For surface fragments whose vertices are not interpolated along edges with a common vertex, the corresponding patch will in general have non-linear behaviour, and will not generate polynomial spline properties.
6. The remaining regions of the cell will be defined by contours immediately above and below the isovalue of some vertex of the cell, and can be assigned that isovalue to generate a piecewise-continuous but not smooth interpolant.

Since the conclusion from Marching Squares is that non-linear behaviour is to be expected, neither the paths nor the details of the interpolants for Marching Cubes have been enumerated.

7 Extension to Marching Hypercubes

More generally, a similar approach can be applied to the Marching Hypercubes defined by Bhaniramka, Wenger & Crawfis [2]. For this, in any given case, the white vertices and the edge points are collected, and their convex hull is computed to serve as the exterior (lower level set) of the contour, the surface of the convex hull being used as the contour itself except at the boundaries.

In two dimensions, this rule generates the cases shown in Figure 3, and in three dimensions, it generates cases nearly identical to the standard set of crack-free Marching Cubes [10].

Because of the convex hull construction, the interpolated edge points expand continuously, and the convex hull surfaces can therefore be shown to be distinct from each other. Take two isovalues $h_1 > h_2$ in the same case and observe that an edge point e_2 for the isovalue h_2 must be in the interior of the edge from an edge point e_1 at isovalue h_1 to some vertex v with isovalue $h_2 > h$. Since all points on the contour at h_2 are convex combinations of edge points constructed in this way, it follows that they are strictly in the interior of the contour at h_1 and the result follows.

8 Digital Connectivity Rules

Carr & Snoeyink [5] showed that the connectivity of the standard crack-free Marching Cubes cases [10] and of Marching Hypercubes [2] are equivalent to connectivity rules from digital imaging, and it follows that the interpolants discussed in Section 5 and Section 6 are also equivalent to these connectivity rules.

9 Conclusions and Future Work

We have shown that the piecewise linear splines of the Contour Spectrum [1] do not easily generalize to non-simplicial meshes, that geometric properties of contours defined by Marching Squares or Marching Cubes cases are not piecewise linear, and and have shown how to construct continuous interpolants whose contours are exactly those of a given set of marching cases.

In practice, the implications of this are that for non-simplicial meshes, geometric properties for the Contour Spectrum [1] or for topological analysis [7] will continue to be computed either by discrete approximations such as voxel count or by explicit extraction of individual contours.

Although it may not be tractable, we would still like to find closed form equations for geometric properties of multilinear interpolants and of marching case contours. Subject to concerns about accuracy, we would also like to find approximate splines of geometric properties.

Acknowledgements

Thanks are due to Science Foundation Ireland for research grant funding and to University College Dublin for equipment grant funding. Further thanks are due to the many individuals who stimulated the ideas that led to this work.

References

1 Chandrajit L. Bajaj, Valerio Pascucci, and Daniel R. Schikore. The Contour Spectrum. In *Proceedings of Visualization 1997*, pages 167–173, 1997.

2 Praveen Bhaniramka, Rephael Wenger, and Roger A. Crawfis. Isosurface Construction in Any Dimension Using Convex Hulls. *IEEE Transactions on Visualization and Computer Graphics*, 10(2):130–141, 2004.

3 Hamish Carr, Brian Duffy, and Barry Denby. On histograms and isosurface statistics. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1259–1266, September/October 2006.

4 Hamish Carr and Nelson Max. Subdivision analysis of the trilinear interpolant. *IEEE Transactions on Visualization and Computer Graphics*, to appear 2009.

5 Hamish Carr and Jack Snoeyink. Representing interpolant topology for contour tree computation. In Hans-Christian Hege, Konrad Polthier, and Gerik Scheuermann, editors, *Proc. Topology-based Methods in Visualization 2007 Workshop*. Springer, 2007.

6 Hamish Carr, Jack Snoeyink, and Michiel van de Panne. Simplifying Flexible Isosurfaces with Local Geometric Measures. In *Proceedings of Visualization 2004*, pages 497–504, 2004.

7 Hamish Carr, Jack Snoeyink, and Michiel van de Panne. Flexible isosurfaces: Simplifying and displaying scalar topology using the contour tree. *Computational Geometry: Theory and Applications*, 43:42–58, 2010.

8 M.J. Dürst. Letters: Additional Reference to "Marching Cubes". *Computer Graphics*, 22(4):65–74, 1988.

9 William E. Lorenson and Harvey E. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *Computer Graphics*, 21(4):163–169, 1987.

10 Claudio Montani, Riccardo Scateni, and Roberto Scopigno. A modified look-up table for implicit disambiguation of Marching Cubes. *Visual Computer*, 10:353–355, 1994.

11 Timothy S. Newman and Hong Yi. A survey of the marching cubes algorithm. *Computers And Graphics*, pages 854–879, 2006.

12 Gregory M. Nielson. On Marching Cubes. *IEEE Transactions on Visualization and Computer Graphics*, 9(3):283–297, 2003.

13 Gregory M. Nielson. Dual Marching Cubes. In *Proceedings of Visualization 2004*, pages 489–496, 2004.

14 Gregory M. Nielson and Bernd Hamann. The Asymptotic Decider: Resolving the Ambiguity in Marching Cubes. In *Proceedings of Visualization 1991*, pages 83–91. IEEE, 1991.

15 Carlos Eduardo Scheidegger, John M. Schreiner, Brian Duffy, Hamish Carr, and Cláudio T. Silva. Revisiting histograms and isosurface statistics. *IEEE Trans. Vis. Comput. Graph.*, 14(6):1659–1666, 2008.

16 Shivaraj Tenginakai, Jinho Lee, and Raghu Machiraju. Salient Iso-Surface Detection with Model-Independent Statistical Signatures. In *Proceedings of Visualization 2001*, pages 231–238, 2001.