A Comparison and Evaluation of Interpolation Methods for Visualising Discrete 2D Survey Data

Burkhard Wünsche and Ewan Tempero

Department of Computer Science
University of Auckland,
Private Bag 92019, Auckland, New Zealand
Email: {burkhard,e.tempero}@cs.auckland.ac.nz

Abstract

Discrete 2D survey data is common in the areas of business, science and government. Visualisation is often used to present such data. The best form of visualisation depends on how the visualisation is to be used. In this paper we present a study of interpolation and approximation techniques for creating visualisations of 2D survey data to be used to present the data to a non-technical audience. We use and modify techniques from the fields of computer-aided design, medical imaging and scientific visualization and we evaluate and compare their suitability for interpolation of the data.

Keywords: Interpolation methods, survey data, reconstruction methods, visualization

Note: Colour versions of the images in this article can be found at http://www.cs.auckland.ac.nz/˜burkhard/Research/InVis2004

1 Introduction

Discrete 2D survey data is common in many real-world applications. An example is a 2D grid with one axis being the income of a person (in steps of, say, NZ$ 10000) and the other axis being the happiness of a person on a scale, say, from 0 to 10. If n people are surveyed the relationship between income and happiness can be revealed by counting for each grid-point the number of people falling into that category and by visualizing these values with colour or height values.

Traditionally discrete data has been visualized with discrete graphical primitives such as bar charts or discrete colour maps. However, the lack of visual continuity makes it difficult to perceive relationships and patterns in the data and also makes it harder to encode additional information dimensions.

In order to find a solution to this problem we looked at the fields of scientific and biomedical visualization. Many scientific data sets are given by discrete sample values and must be transformed into continuous data before transformation. This type of data transformation is an example of data enrichment and can be accomplished by using interpolation techniques such as scattered data interpolation (Hoschek & Lasser 1992) or finite element interpolation (Burnett 1987). When dealing with 3D scalar data the problem is often referred to as volume reconstruction. Optimal reconstruction functions for specific application areas such as volume rendering are an ongoing topic of research (e.g., (Marschner & Lobb 1994, Moorhead II & Zhu 1995, Möller, Müller, Kurzion, Machiraju & Yagel 1998, Mueller, Möller & Crawfis 1999)). Reconstruction of vector and tensor data is more complicated. A technique based on spectral analysis has been suggested by Aldroubi and Basser (Aldroubi & Basser 1999).

In this paper we present, evaluate, and compare different interpolation methods from the fields of scientific visualization, biomedical imaging and computer-aided design and we apply them to discrete 2D survey data. The presented work is the result of a research project performed for an international consulting company. We motivate the various methods introduced in this work by using as example a data set of similar type to that used in our consulting work. We list the advantages and drawbacks of each interpolation method and as a result of our analysis we suggest one interpolation method as the preferred solution.

2 Visualisation Requirements

The data to be visualised comes from a particular kind of survey that might be carried out in some organisation. The survey consists of statements to which respondents answer on a scale of "strongly disagree" to "strongly agree". The statements relate to various aspects of the current state of the organisation and how it operates. The raw data consists of the number of respondents in each category for each statement. The purpose of the survey is to help the organisation understand aspects of itself and to help with planning, and so the results are presented to some part of the organisation, typically representatives of upper management.

This context has certain implications on the kind of data to be visualised, and the properties the visualisations should have. The data represents subjective responses, rather than objective measurements. This means that, while the results are discrete by virtue of how they are provided, the "truth" they represent is not as obvious as the numbers themselves may imply. For example, a person might be unable to decide between "agree" and "strongly" degree but can't choose a value in the middle of these two answers. Similarly two people might have different definitions of the word "agree". On the other hand there is a clear boundary between "agree" and "disagree". This suggests that using the discrete values directly in the visualisation may be misleading.

There is another reason for considering continuous approximations. The intended audience of the visualisations is interested in trends, and getting an overview of the organisation. Our client has extensively used 2D visualisations of data and the experience is that continuous visualisations are more "visually pleasing" than discrete ones, and so work best for this kind of audience. Generally, when choosing between visualisations for this audience, the preference is for the more visually pleasing ones.

Any form of interpolation produces artifacts in the result. For example, some forms are "more pointy" than others, some will produce curves that go through the sample points whereas others will not, and all handle boundary conditions differently. Correctly interpreting a visualisation requires some understanding of the consequences

Copyright ©2004, Australian Computer Society, Inc. This paper appeared at the Australasian Symposium on Information Visualisation, Christchurch, 2004. Conferences in Research and Practice in Information Technology, Vol. 35. Neville Churcher and Clare Churcher, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

of these artifacts. Our intended audience will generally not be knowledgeable in such things are interpolation artifacts, and so we must avoid techniques with artifacts that result in “obviously wrong” visualisations. A specific example of this is techniques that produce negative results — the audience will be aware that there cannot be a negative number of people selecting a certain answer in a survey. Another example is the “degree of smoothness”. Since our sample grid is relatively small we do not want to have excess smoothing since it might lead to potentially misleading results. On the other hand a limited degree of smoothing is desirable since it leads to a visually more pleasing surface, and reduces noise in the data.

Finally, in anticipation of planned development of the survey, our client has requested that we explore the use of three-dimensional visualisations of the data.

The data sets used in our research consist of a two-dimensional grid where each coordinate is an integer in the range $[-3, 3]$. The data was obtained from a range of survey questions with answers ranging from “strongly disagree” (-3), to “neutral” (0), to “strongly agree” (+3). Each survey record represents a point in a 2D grid that is pictured in figure 1 (a). Since the extreme values (-3 and 3) on either axis characterize the surveyed group of people into 4 distinct categories, a visually more pleasing representation is obtained by distorting the grid so that the axes are the diagonals of the data domain. Figure 1 (b) shows that each quadrant of the resulting coordinate system represents one of the before mentioned categories. We have changed the labels used in this and later figures at the request of our client. For this presentation, we use just ‘A’,‘B’,‘C’ and ‘D’.

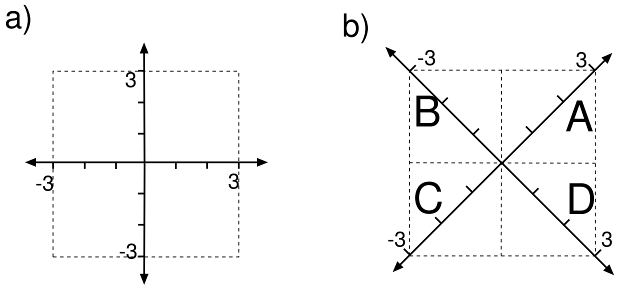

Figure 1: Data grid for discrete two-dimensional survey data before (a) and after (b) distortion which aligns coordinate quadrants with categories in the survey data.

3 Interpolation Methods

This section presents an overview of popular interpolation and reconstruction methods from the fields of scientific visualization, biomedical visualization and computer-aided design, and shows how they can be applied to our data set.

The two-dimensional interpolation methods we considered can be classified as either *scattered data interpolation, finite element interpolation, parametric spline surfaces* or *reconstruction filters*. These represent increasing restrictions on the structure of the sample points. Scattered data interpolation has no restriction on the structure of the data points, and generally produces, for a 2D grid of sample points, a surface that interpolates each of the sample values. The interpolation is usually global, that is, a change in a sample value can effect the shape of the entire surface. The finite element techniques essentially “stitch together” small pieces of surface into one large surface. An important issue with them is ensuring that the joins are sufficiently smooth. Finite element interpolation uses the data points to form a tessellation of the domain and the interpolation takes place only over each tile, that is, the interpolation only takes into account data points that are neighbours. Reconstruction filters are a simple way to reconstruct a function from a regular grid of sample points. The size and shape of a reconstruction filter determine the speed and smoothness of the reconstruction. Since the geometry of the grid is required to be regular the same reconstruction filter can be used for all data points, which makes it possible to efficiently compute a reconstructed surface approximating the sample point values.

3.1 Scattered Data Interpolation

Scattered data interpolation reconstructs a continuous function $f(\mathbf{x})$ from sample points $\{(\mathbf{x}_1, f_1), (\mathbf{x}_2, f_2), \ldots, (\mathbf{x}_N, f_N)\}$, where in our case each $\mathbf{x}_i$ is a 2D grid coordinate (Hoschek & Lasser 1992). One of the earliest and most popular class of scattered data interpolation methods are variations of the *Shepard method* (Shepard 1968) in which the interpolated function is defined as

$$f(\mathbf{x}) = \sum_{i=1}^{N} \omega_i(\mathbf{x}) f_i$$

where f_i are the sample values and ω_i are weighting functions based on the distance to the grid points. This means that every interpolated point is the weighted sum of every sample point. The method is a global method and hence computational expensive. Furthermore adding one more point requires a recomputation of all weighting function, though this is not a drawback in our application where the grid size is fixed. The main drawback of the Shepard method is that the interpolant is in general not particularly smooth. Therefore we consider instead two other approaches for scattered data interpolation.

The solution to the scattered data interpolation problem with the help of *radial basis functions* has the form

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i R(d_i(\mathbf{x})) + p_m(\mathbf{x})$$

where

$$p_m(\mathbf{x}) = \sum_{j=1}^{m} \beta_j p_j(\mathbf{x})$$

The univariate basis functions $R(d_i(\mathbf{x}))$ are non-negative functions of the distance $d_i(\mathbf{x})$ of the point $\mathbf{x}$ to the sample point $\mathbf{x}_i$. The low-dimensional polynomial $p_m(\mathbf{x})$ guarantees that a sampled polynomial function with the degree m can be exactly reproduced ($\alpha_i = 0$, i=1,…,N) (Dyn 1989, Alfeld 1989). The unknown coefficients are obtained from the interpolation constraints

$$f(\mathbf{x}) = f_i \tag{1}$$

which ensure that the surface goes through the sample points, and from the m additional conditions

$$\sum_{i=1}^{N} \alpha_i p_j(\mathbf{x}_i) = 0 \quad \text{for} \quad j = 1, \ldots, m \tag{2}$$

The resulting linear system of $N + m$ equations can easily become badly conditioned (Hoschek & Lasser 1992). A solution based on preconditioning the matrix has been suggested in (Dyn 1989). Carr et al. propose instead a modification that makes it possible to model large data sets consisting of million of data points (Carr, Beatson, Cherrie, Mitchell, Fright, McCallum & Evans 2001, Carr, Beatson, McCallum, Fright, Lennan & Mitchell 2003). More information about radial basis functions is found in (Dyn, Leviatan, Levin & Pinkus 2001).

For our application we choose the radial basis function

$$R(r_i) = r_i^2 \log r_i \quad \text{with} \quad r_i = d_i(\mathbf{x})$$

The resulting interpolant minimises the bending energy of a thin plate interpolating the sample points and is also called *Duchon's Thin Plate Spline* (Hoschek & Lasser 1992).

3.2 Finite Element Interpolation

The geometry of a *finite element* model is described by a set of nodes and a set of elements, which have these nodes as vertices. The nodal coordinates are interpolated over an element using *interpolation functions*. Curvilinear elements can be defined by specifying nodal derivatives (Burnett 1987).

In our application the cells of the regular grid can be considered as finite elements. In this case an interpolation of the geometry is not necessary and the finite element interpolation functions are used only to interpolate the sample values over the grid cells.

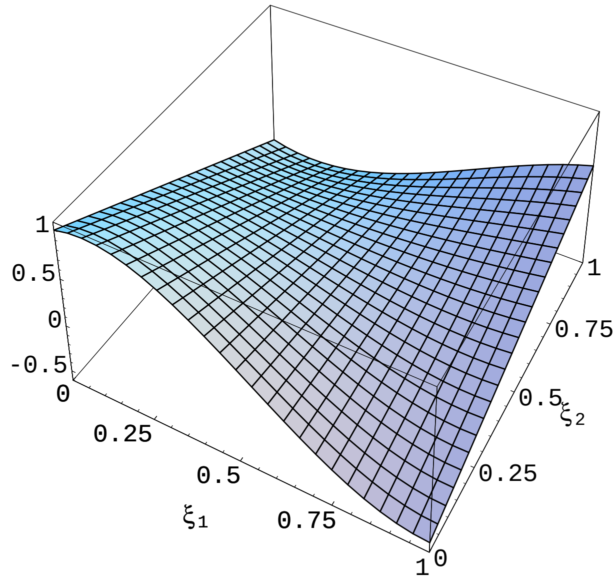

Figure 2: Cubic Hermite-Linear Lagrange interpolation of sample values at the corners of a square element.

As an example consider the cubic Hermite-linear Lagrange interpolation over a quadratic element shown in figure 2. Assumed we define local coordinates ξ_i such that a grid cell corresponds to a unit square in the local coordinates ($0 \leq \xi_1, \xi_2 \leq 1$). The value of some function f over the element is then specified by interpolating the variables f_i linearly in the given parameter direction. In our example we assume that additionally derivatives in ξ_1-direction $\left(\frac{\partial f}{\partial \xi_1}\right)_i$ $(i = 1, \ldots, 4)$ are specified at the element vertices. In this case a cubic Hermite interpolation is performed in that direction.
The cubic Hermite-linear interpolation of f over the entire 2D parameter space is then defined by the tensor products of the interpolation functions in each parameter direction:

$$\begin{aligned} & f(\xi_1, \xi_2) = \\ & H_1^0(\xi_1)L_1(\xi_2)f_1 + H_2^0(\xi_1)L_1(\xi_2)f_2 \\ + \; & H_1^0(\xi_1)L_2(\xi_2)f_3 + H_2^0(\xi_1)L_2(\xi_2)f_4 \\ + \; & H_1^1(\xi_1)L_1(\xi_2)\left(\frac{\partial f}{\partial \xi_1}\right)_1 + H_2^1(\xi_1)L_1(\xi_2)\left(\frac{\partial f}{\partial \xi_1}\right)_2 \\ + \; & H_1^1(\xi_1)L_2(\xi_2)\left(\frac{\partial f}{\partial \xi_1}\right)_3 + H_2^1(\xi_1)L_2(\xi_2)\left(\frac{\partial f}{\partial \xi_1}\right)_4 \end{aligned}$$

where

$$L_1(\xi) = 1 - \xi, \quad \text{and} \quad L_2(\xi) = \xi$$

are the one-dimensional linear Lagrange basis functions, and

$$\begin{aligned} H_1^0(\xi) &= 1 - 3\xi^2 + 2\xi^3, & H_1^1(\xi) &= \xi(\xi - 1)^2 \\ H_2^0(\xi) &= \xi^2(3 - 2\xi), & H_2^1(\xi) &= \xi^2(\xi - 1) \end{aligned}$$

are the one-dimensional cubic Hermite basis functions.

In general we can express the interpolation of a variable as

$$f(\boldsymbol{\xi}) = \sum_{i,k} f_i^k \phi_i^k(\boldsymbol{\xi}) \tag{3}$$

where f_i^k are scalar field values and their partial derivatives (if any) at each vertex and ϕ_i^k are appropriate interpolation functions.

3.3 Reconstruction Filters

For a regular grid of sample values the interpolation problem can be stated as a convolution with a *reconstruction* filter. Let us denote the two-dimensional sample values by f_{ij} and the corresponding grid points by $\mathbf{x}_{ij}$. Then the reconstructed function is defined by

$$f(\mathbf{x}) = \sum_i \sum_j f_{ij} h(\mathbf{x} - \mathbf{x}_{ij})$$

where $h(\mathbf{x})$ is the reconstruction filter. *Separable* filters can be written as $h(\mathbf{x}) = h_s(x)h_s(y)$ and allow a more efficient computation of the reconstructed function.

A useful family of reconstruction filters are the *n-th degree B-Spline filters* that can be defined by box filtering a square pulse function. Let $*$ denote the convolution operator, i.e.,

$$(g * h)(x) = \int_{-\infty}^{\infty} g(u)h(x - u)du$$

and $B^0(x)$ the square pulse function (0-th degree B-Spline) defined by

$$B^0(x) = \begin{cases} 1 & -0.5 \leq x \leq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

Higher order reconstruction filters can be obtained by repeatedly convolving this function with the square pulse function. The most common B-Spline filters are the linear B-Spline filter

$$B^1(x) = (B^0 * B^0)(x) = \begin{cases} 1 & -0.5 \leq x \leq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

the quadratic B-Spline filter

$$B^2(x) = (B^1 * B^0)(x) = \begin{cases} \frac{(2x+3)^2}{8} & -1.5 \leq x < -0.5 \\ \frac{3}{4} - x^2 & -0.5 \leq x < 0.5 \\ \frac{(2x-3)^2}{8} & 0.5 \leq x < 1.5 \\ 0 & \text{otherwise} \end{cases}$$

and the cubic B-Spline filter

$$\begin{aligned} B^3(x) &= (B^2 * B^0)(x) \\ &= \begin{cases} \frac{1}{6}\left(x^3 + 6x^2 + 12x + 8\right) & -2 \leq x < -1 \\ \frac{1}{6}\left(-3x^3 - 6x^2 + 4\right) & -1 \leq x < 0 \\ \frac{1}{6}\left(3x^3 - 6x^2 + 4\right) & 0 \leq x < 1 \\ \frac{1}{6}\left(-x^3 + 6x^2 - 12x + 8\right) & 1 \leq x < 2 \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

The linear, quadratic and cubic B-Spline filter are illustrated in figure 3.

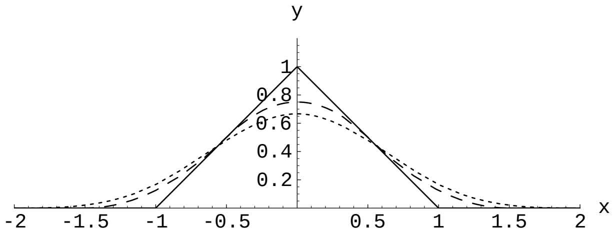

Figure 3: Graphs of the linear B-Spline filter (solid line), the quadratic B-Spline filter (dashed line) and the cubic B-Spline filter (dotted line).

Another popular cubic reconstruction filter is the Catmull-Rom spline (Catmull & Rom 1974) that is defined as

$$C(x) = \begin{cases} \frac{1}{6}\left(3x^3 + 15x^2 + 24x + 12\right) & -2 \leq x < -1 \\ \frac{1}{6}\left(-9x^3 - 15x^2 + 6\right) & -1 \leq x < 0 \\ \frac{1}{6}\left(9x^3 - 15x^2 + 6\right) & 0 \leq x < 1 \\ \frac{1}{6}\left(-3x^3 + 15x^2 - 24x + 12\right) & 1 \leq x < 2 \\ 0 & \text{otherwise} \end{cases}$$

and is pictured in figure 4. In contrast to the quadratic and the cubic B-Spline filter the Catmull-Rom filter smooths the data while still interpolating the sample points.

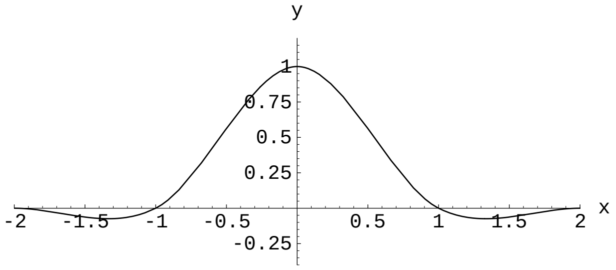

Figure 4: Graph of the Catmull-Rom spline.

3.4 Parametric Spline Surfaces

A parametric spline surface can be defined as

$$\mathbf{p}(s,t) = \sum_i \phi_i(s,t)\mathbf{p}_i$$

where

$$s \in [s_{min}, s_{max}], \text{ and } t \in [t_{min}, t_{max}]$$

and s, t are the parameters of the surface, $\phi_i(s,t)$ are the *basis functions* (interpolation functions) and $\mathbf{p}_i$ are control points or other control parameters such as coordinate curve tangents. If the control points $\mathbf{p}_i$ are homogeneous the spline is called *rational*.

The arguably most important parametric splines are *Non-uniform rational B-Splines* (NURBS) that are the industry standard tool for the representation and design of geometry (Piegl 1991, Farin 1995). A NURBS surface is defined as

$$\mathbf{p}(s,t) = \sum_i \sum_j B_i^k(s) B_j^l(t)(s,t)\mathbf{p}_i$$

where

$$s \in [s_{min}, s_{max}], \text{ and } t \in [t_{min}, t_{max}]$$

and B_i^k are one dimensional non-uniform B-Splines of degree k that can be recursively defined by the Cox-de Boor algorithm (de Boor 1972)

$$\begin{aligned} B_i^0(t) &= \begin{cases} 1 & t_i \leq t \leq t_{i+1} \\ 0 & \text{otherwise} \end{cases} \\ B_i^k(t) &= \frac{t - t_i}{t_{i+k-1} - t_i} B_i^{k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} B_{i+1}^{k-1}(t) \end{aligned}$$

The *knotvector* $t_0, t_1, \ldots, t_{m-1}$ forms the parameter interval for the t parameter. The basis functions and knotvector for the s parameter are defined similarly.

For our interpolation problem we can use NURBS surfaces by using the sample point coordinates (i, j) as knots and by using the sample point values as control points. If the grid points are uniformly spaced and we choose k-th degree B-Spline basis functions then we obtain a k-th degree B-Spline reconstruction filters as introduced in subsection 3.3. The advantage of the NURBS representation is that we can also use non-uniform grids. For our application we choose another useful property of the NURBS interpolation. If we multiply the first and last knot for a k-th degree non-uniform B-Spline $(k + 1)$ times then the B-Spline will interpolate its endpoints (Hoschek & Lasser 1992).

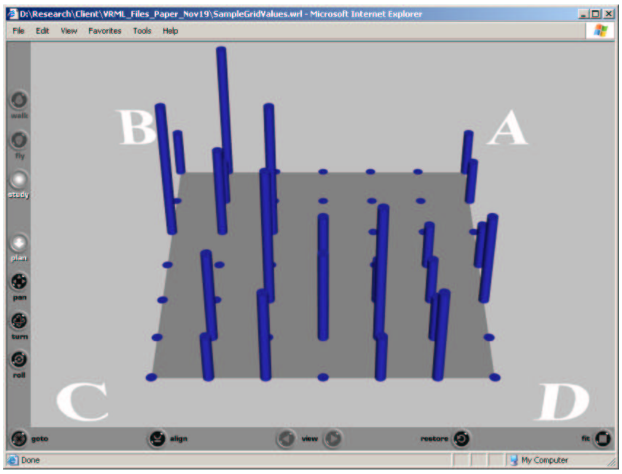

Figure 5: VRML file of a height field constructed from a 2D survey and viewed with the Cortona VRML viewer.

4 Results

We have implemented the scattered data interpolation and reconstruction filter techniques discussed above. The implementation is in Java, and produces VRML as output. We chose VRML as our output medium due to requirements of our client, specifically the ability to provide the visualisations on web pages, while still retaining some level of interactivity, such as changing the viewing angle. By using VRML, any presentations can be done without the need to install a Java runtime, which simplifies the client's setup. A VRML viewer does need to be installed, but doing so is significantly easier than Java. The viewer we use is Cortona by Parallel Graphics (Parallel Graphics Inc. 2002). An example is shown in figure 5. The viewer provides several different navigations options such as "walking", "flying" and "exploring". Additional features such as restoring the default view, fitting the visualization into the window and changing rendering parameters are also available.

To demonstrate the different techniques, we use an example survey data set with 40 survey records. The data is transformed into an 7 × 7 grid as shown in figure 1. Each grid point is associated with an integer that represents the number of respondents choosing that answer in the survey. The resulting data set is visualized in figure 5 as a height field.

4.1 Scattered Data Interpolation

As an initial attempt to solve the interpolation problem for discrete 2D survey data we use the radial basis functions described in subsection 3.1 with

$$R(r_i) = r_i^2 \log r_i \quad \text{with} \quad r_i = d_i(\mathbf{x})$$

and

$$p_3(\mathbf{x}) = \beta_0 + \beta_1 x + \beta_2$$

where $\mathbf{x} = (x, y)$ are the coordinates of the 2D sample point grid shown in figure 1. Since we have 7×7 sample points and $m = 3$ polynomial coefficients the constraints 1-2 result in a linear system of 52 equations that can be efficiently solved using an LU solver with partial pivoting (Press, Vetterling, Teukolsky & Flannery 1992). The resulting visualization is shown in figure 6. Adding the height field in figure 5 into the visualization shows that the method exactly interpolates sample points. The resulting surface is very smooth and visually pleasing. However, in some regions the surface becomes negative, which indicates an impossible result since we can't have a negative number of people choosing a certain answer in a survey.

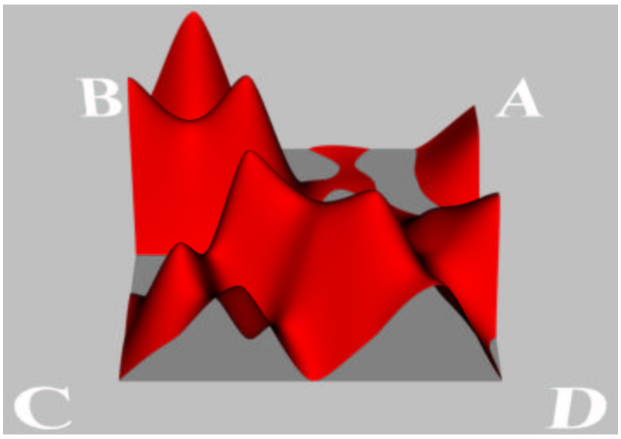

Figure 6: The 2D example survey data interpolated with radial basis functions.

4.2 Reconstruction Filters

In this subsection we present the results obtained by using various reconstruction filters. Figure 7 shows the visualization obtained by using a Catmull-Rom spline reconstruction filter. The surface is again smooth and interpolates the sample point values. In addition it can be seen that the surface is less "pointy", i.e., its curvature at the peaks of the sample points is lower then for the interpolation obtained using radial basis functions. However, the surface has again negative regions with is due to the fact that the reconstruction filter shown in figure 4 has negative values.

Negative surface values can be avoided by using B-Spline reconstruction filters. Since the B-Spline basis functions lie between zero and one and sum up to one for any point (Hoschek & Lasser 1992) the interpolated surface will always lie in the convex hull of the sample points.

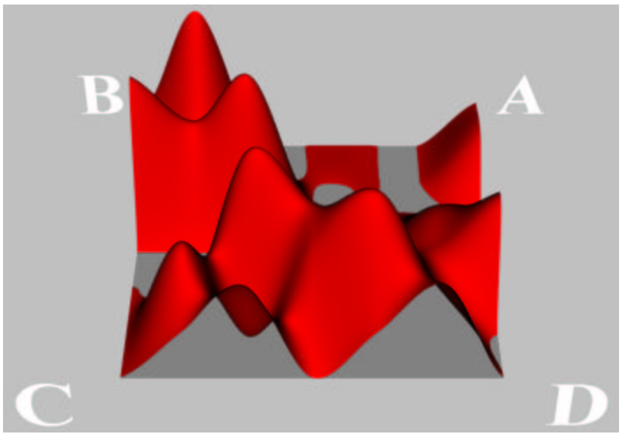

Figure 7: The 2D example survey data interpolated with a Catmull-Rom spline reconstruction filter.

Figures 8- 10 show the visualizations obtained with a bilinear, biquadratic and a bicubic B-Spline reconstruction filter.

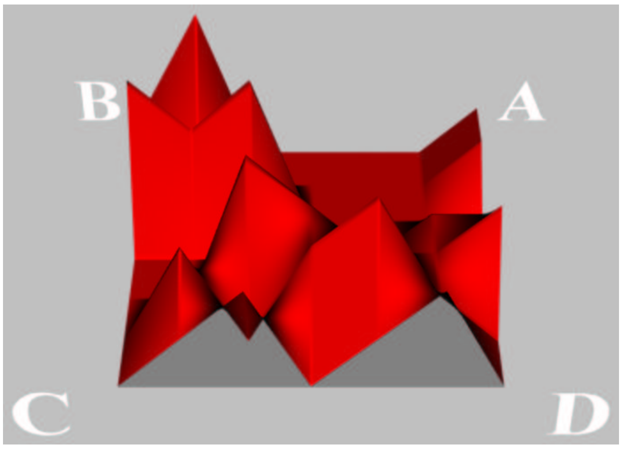

Figure 8: The 2D example survey data interpolated with a bilinear B-Spline reconstruction filter.

It can be seen that the bilinear reconstruction filter interpolates the sample point values whereas the higher order filters do not. The degree of smoothing increases with an increase in the degree of the reconstruction filter. This is due to the definition of the B-Spline reconstruction filters that were defined by repeatedly box-filtering a square-pulse function.

As we observed in section 2, there is an inherent uncertainty in the sample values, and so it is not strictly necessary to interpolate the sample point values. We also mentioned that we would prefer more smooth to less smooth. Nevertheless, the higher order filters, while being more smooth, will also produce surfaces that are further from the sample points, and we must also avoid visualisations that are misleading.

Our conclusion is that the *support* (extent) of the smoothing filter should be at most equal to the length of the "agree" and "disagree" sections of the sample grid axes. The quadratic B-Spline filter has a support of three units (-1.5 to 1.5) that is equal to half the length of an axes of our sample grid (-3 to 3) and it gives therefore in our opinion a suitable trade-off between smoothing and data accuracy.

Note that the interpolation with reconstruction filters is efficient since the reconstruction filters have a finite support. Computing a surface point using k-th degree B-Spline reconstruction filters requires the addition of

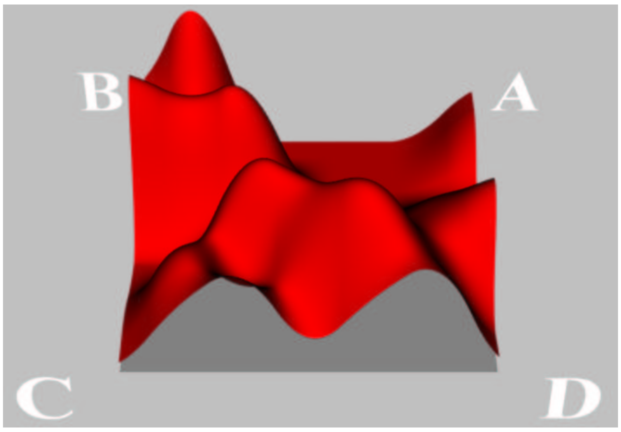

Figure 9: The 2D example survey data interpolated with a biquadratic B-Spline reconstruction filter.

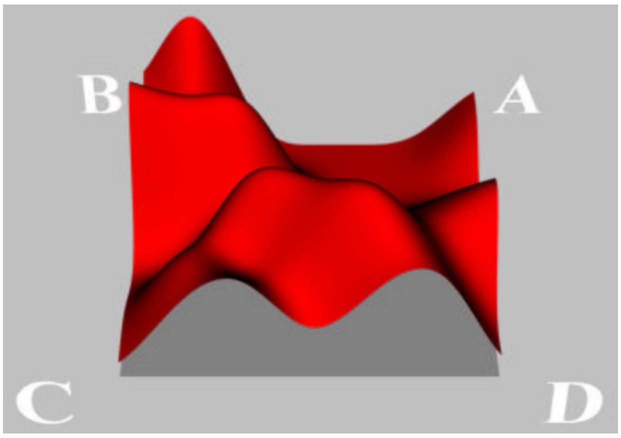

Figure 10: The 2D example survey data interpolated with a bicubic B-Spline reconstruction filter.

$(k+2) \times (k+2)$ products of sample points and filter functions.

4.3 Parametric Spline Surfaces

In order to complete our comparison of different classes of interpolation methods we have also implemented a bicubic NURBS surface. Endpoint interpolation is achieved by multiplying the first and the last sample point four times so that the knotvector is -3,-3,-3,-3,-2,-1,0,1,2,3,3,3,3. Since four knot intervals define one B-Spline basis function this knot vector gives us nine basis functions. However, we have only 7 sample values in each parameter direction (sample values are defined for the grid point $-3, -2, \ldots, 3$). We therefore also multiply the first and last sample point value in each parameter direction.

The resulting surface is visualized in figure 11. Comparing this image with the bicubic B-Spline filter in figure 10 it can be seen that the surface now interpolates the sample points over the boundary of the domain. The volume under the surface is, however, different. In some sense, the B-Spline filters produce volumes that are the same as the original sample points (figure 5), and so for this reason we prefer them.

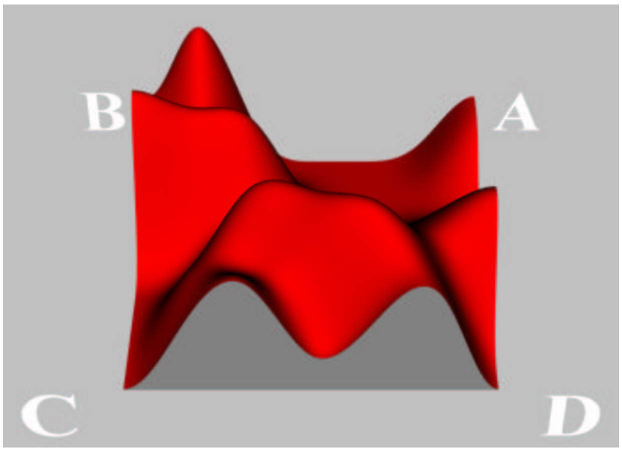

Figure 11: The 2D example survey data interpolated with a bicubic endpoint-interpolating NURBS surface.

5 Conclusion

We have briefly surveyed interpolation techniques from the fields of scientific visualization, biomedical imaging and computer-aided design for two-dimensional data, with the aim to identify the most suitable technique to apply to our particular set of requirements. These requirements prefer visually pleasing over accurate representations, and prefer not to have representations with artifacts that are obviously invalid. We have implemented a number of these techniques to allow our client to evaluate them.

Based on these requirements, we concluded the biquadratic B-Spline reconstruction filter suited our purposes best. While it does not exactly interpolate the sample points, it does so well enough and the sample points themselves cannot be regarded as exact. It also provides surfaces that are not negative, thus avoiding distracting artifacts for the non-technical audience. Finally, it provides a surface that is sufficiently smooth to be visually pleasing. When presented with the different solutions our client was indeed most happy with this choice. This solution has been trialled by our client and reports that participants were happy with it.

This investigation has raised a number of interesting questions about the use of interpolation techniques for information visualisation that we wish to pursue. We initially did not consider the finite element interpolation for implementation, since the standard implementation does not provide smoothing between grid cells. However, we now think that variations of this method might enable us to get more control over the smoothing process while maintaining the constraint that surfaces are non-negative. Whether such finite element interpolation methods are useful for the field of information visualisation remains to be seen.

References

Aldroubi, A. & Basser, P. J. (1999), Reconstruction of vector and tensor fields from sampled discrete data, *in* 'The Functional and Harmonic Analysis of Wavelets and Frames', Vol. 247 of *Contemporary Mathematics*, American Mathematical Society, pp. 1–15.

Alfeld, P. (1989), Scattered data interpolation in three or more variables, *in* T. Lyche & L. Schumaker, eds, 'Mathematical Methods in Computer Aided Geometric Design', Academic Press, pp. 1–34. `http://citeseer.nj.nec.com/alfeld89scattered.html`.

Burnett, D. S. (1987), *Finite Element Analysis - From Concepts to Applications*, Addison-Wesley Publication Company Inc.

Carr, J. C., Beatson, R. K., Cherrie, J. B., Mitchell, T. J., Fright, W. R., McCallum, B. C. & Evans, T. R. (2001), Reconstruction and representation of 3d objects with radial basis functions, *in* 'Proceedings of ACM SIGGRAPH 2001', Computer Graphics Proceedings, Annual Conference Series, pp. 67–76. `http://aranz.com/download/modelling/papers/siggraph01.pdf`.

Carr, J. C., Beatson, R. K., McCallum, B. C., Fright, W. R., Lennan, T. J. & Mitchell, T. J. (2003), Smooth surface reconstruction from noisy range data, *in* 'Proceedings of GRAPHITE 2003', ANZGRAPH and SEAGRAPH, pp. 119–226. `http://aranz.com/download/modelling/papers/graphite03.pdf`.

Catmull, E. & Rom, R. (1974), A class of local interpolationg splines, *in* R. E. Barnhill & R. Riesenfled, eds, 'Computer Aided Geometric Design', Academic Press.

de Boor, C. (1972), 'On calculating with b-splines', *Journal of Approximation Theory* **6**, 50–62.

Dyn, N. (1989), Interpolation and approximation by radial and related functions, *in* C. K. Chui, L. L. Schumaker & J. D. Ward, eds, 'Approximation Theory VI', Vol. 1, Academic Press, pp. 211–234.

Dyn, N., Leviatan, D., Levin, D. & Pinkus, A., eds (2001), *Multivariate Approximation and Applications*, Cambridge University Press.

Farin, G. E. (1995), *NURBS curves and surfaces: from projective geometry to practical use*, A.K. Peters.

Hoschek, J. & Lasser, D. (1992), *Fundamentals of Computer Aided Geometric Design*, 2^{nd} edn, AK Peters Ltd., Wellesley, MA 02181, chapter 14, pp. 572 – 601.

Marschner, S. R. & Lobb, R. J. (1994), An evaluation of reconstruction filters for volume rendering, *in* R. D. Bergeron & A. E. Kaufman, eds, 'Proceedings of Visualization '94', IEEE, pp. 100–107.

Möller, T., Müller, K., Kurzion, Y., Machiraju, R. & Yagel, R. (1998), Design of accurate and smooth filters for function and derivative reconstruction, *in* 'Proceedings of the 1998 Symposium on Volume Visualization (VOLVIS-98), Research Triangle Park, North Carolina, October 19-20)', ACM Press, pp. 134–151.

Moorhead II, R. J. & Zhu, Z. (1995), 'Signal processing aspects of scientific visualization', *IEEE Signal Processing Magazine* **12**(5), 20–41. URL: `http://www.erc.msstate.edu/research/labs/vail/pubs/sig_proc/spmag_06.html`.

Mueller, K., Möller, T. & Crawfis, R. (1999), Splatting without the blur, *in* D. Ebert, M. Gross & B. Hamann, eds, 'Proceedings of Visualization '99', IEEE, pp. 363 – 370.

Parallel Graphics Inc. (2002), 'Cortona VRML client homepage'. URL: `http://www.parallelgraphics.com/products/cortona`.

Piegl, L. (1991), 'On NURBS: A survey', *IEEE Computer Graphics and Applications* **11**(1), 55–71.

Press, W. H., Vetterling, W. T., Teukolsky, S. A. & Flannery, B. P. (1992), *Numerical Recipes in C - The Art of Scientific Computing*, 2^{nd} edn, Cambridge University Press. URL: `http://www.library.cornell.edu/nr/bookcpdf.html`.

Shepard, D. (1968), A two-dimensional interpolations function for irregularly spaced data, *in* 'Proceedings of the 23rd Nactional Conference of the ACM', pp. 517–523.