

Analysis on Triangulations

N. Noble*, Kerner Graphics, Inc.

Abstract

We develop structures associated with triangulations which permit the application, directly to the vertices of a triangulation, of tools and processes normally restricted to continuous data, or at least data of sufficiently fine resolution and precision that the approximations of numerical analysis yield reasonable results. These structures include polynomial surfaces associated with, and reflecting many of the properties of, each vertex; mappings whose Fourier attributes capture additional properties of a vertex not reflected by those surfaces; and hierarchies created through a combination of smoothing and size reduction which extend the reach of these local structures to more distant vertices. The surfaces associated with vertices are generated as linear combinations of functions created by extending to the plane polynomial versions of the functions $\sin n\theta$ and $\cos n\theta$ defined on the unit circle.

The research presented here was motivated by the development of a video compression product. It was undertaken specifically in support of that objective.

1 Introduction

Triangulations are widely used in applications such as image processing, computer graphics, terrain evaluation, and elsewhere, as succinct representations of large data sets. Data is encoded both in the triangulation itself and, optionally, in augmenting values assigned to the vertices of the triangulation. We refer to such structures as *augmented* triangulations (terms shown in bold italics are being defined; see for example the monograph [1] for a definition of triangulations); formally an augmented triangulation is a triangulation, a vector space, and a map which assigns to each vertex of the triangulation a value in that vector space. By organizing the data involved and reducing its volume, these structures can increase the speed and efficiency of processing. Furthermore, because the extent of the reduction can be controlled by adjusting the placement and number of vertices, this efficiency can be achieved with an acceptable loss of accuracy. But the strength of triangulations is also their weakness: as sparse sets of data they are not directly subject to the powerful tools of traditional analysis, differential geometry, or differential topology, at least to the extent that these tools depend upon continuity or a reasonable approximation thereof.

Various approaches have been developed to compensate for such shortages of data, most commonly calculations such as convolutions over a rectangular window which utilize the sampling structure of regularly sampled data. Triangulations do not have a regular sampling structure and approaches focused on them usually proceed, at least implicitly, by associating a continuous surface with a vertex or set of vertices, then basing the calculations upon those surfaces. Examples include surfaces which are piecewise flat, quadric, conic, cubic, Bezier patches, and others. The article [2] includes a survey of early work with various digital operators; it and [3] treat all first and second order operators; the papers [4], [5], and [6] include surveys of traditional discrete methods of estimating curvature on triangulations. Others estimate curvature using cubic surfaces, [7]; calculate discrete mean curvature by first calculating the Laplace-Beltrami operator, [8]; and, viewing triangulations as finite metric spaces, calculate curvatures using techniques from the field of metric geometry, [9].

Common among these approaches for triangulations is a view of the triangulation as approximating a surface (or space of higher dimension), and a goal of approximating the properties of that object. Our interest, motivated by a specific application described below (compression of movie and video streams) focus upon two processes on triangulations – smoothing their associated values and establishing correlations between them – with the result that we are not wedded to any particular surface. Instead we associate a different surface with each vertex, our goals being repeatability (obtaining similar results for vertices derived from similar image features) and ease of computation.

1.1 Motivation

While we envision applications to a range of tasks involving matching, identification, analysis, or compression of large data sets, our immediate motivation, as mentioned above, is compression of video or “moving picture” streams and in particular the

*The author gratefully acknowledges partial support from the National Science Foundation, Grant IIP 0810023. All statements here are the responsibility of the author, not of the National Science Foundation.

automatic correlation of neighboring images in such a stream, with the goal that pixels will be associated with each other which a human viewer would describe as the “same” points in the two images. The correlations created between neighboring images can be used to create intermediate images, for example, as part of an artistic “morphing” function, as intermediate views in engineering simulations, or, in our case of interest, to recreate approximations of images which have been discarded as part of a compression process. In the decompressed stream these correlations can be used to pause between frames, to vary playback speed (accommodating different processor clock rates or display requirements), and to implement slow motion viewing.

We will focus on the process of correlating vertices in triangulations of neighboring images. The initial step in constructing such correlations is usually the identification of characteristics or attributes of a vertex and its value which help differentiate them from other vertices. A useful property of such attributes is that they be invariant under various operations, in particular translation or rotation. By centering our constructions on the vertex under consideration, we insure that the associated attributes in which we are interested will be invariant under translation. They will also respond, under rotation, in a manner which, while not invariant, is at least predictable. Our association of a polynomial surface with each vertex permits the use of traditional properties of surfaces, such as curvature and critical points. We use the Fourier transform to create these associations, facilitating the use of attributes such as spectral power/amplitude or spectral phase as well as others based upon Fourier coefficients. In addition, we show how much of the information ignored in our passage from vertices to the surfaces which we associate with them can be codified, separately, as Fourier attributes, using the same techniques we used for the surfaces.

Our intended compression algorithm imposes requirements on the correlation process which differ somewhat from the familiar processes of image registration and image warping, so we begin with a broad overview of that compression algorithm. For our purposes an *image* is a two dimensional $N \times M$ array of *pixels*, each pixel having an associated *value*, where the values are drawn from some d -dimensional vector space (N , M , and d being positive integers). An image is thus a two dimensional surface embedded in a space of $2 + d$ dimensions, simple examples being greyscale images where $d = 1$ and color images where $d = 3$ for additive displays and 4 (or more) for practical subtractive processes. Data structures used in computer graphics including possibly additional attributes such as texture,

opacity (transmissivity), and/or reflectivity can also be treated as “images” in this sense.

We will be concerned with images which are *structurally equivalent* in that they are based upon the same $N \times M$ array with their values drawn from the same d -dimensional vector space. An *image stream* is a finite sequence of structurally equivalent images indexed along some discrete axis which we can think of as equally spaced instances of time. By breaking image streams into shorter streams if necessary, we may assume that the each image in a stream is substantially similar to its neighbor(s). We seek to compress such streams using the steps listed below.

- Identify as *points of interest* (the initial vertices) both *critical points* and *jump points*: critical points are pixels whose values are local maxima, minima, or saddle points, and jump points are pixels whose values differ from that of a neighbor by more than some fixed threshold. (Here comparisons of values refer to measurements in the vector space of image values.)
- Identify *sub-boundaries* – “connected” curves of jump points which persist in time and move with respect to the critical points. Vertices on the perimeter of the $N \times M$ array are *boundary vertices*, vertices in sub-boundaries are *sub-boundary vertices*, and the remaining vertices are *interior vertices*. Note that in general sub-boundaries will be doubled, with one curve representing lower values and one higher values.
- Form a unique constrained Delaunay Triangulation (CDT) of the set of points of interest, using the segments connecting the boundary vertices and those connecting the sub-boundary vertices as constraints. A planar CDT always exists (unlike the case in higher dimensions [10]) and, as with Delaunay triangulations, CDTs maximizes the minimum angle among the admissible triangulations [11]. For vertices in general position (no four points co-circular) a CDT is unique, but our vertices are drawn from a rectangular grid and hence may well not be in general position. In order to guarantee that identical vertices and constraints in different images will receive the same triangulation, we use, as in the article [12], a pair of preferred directions to disambiguate diagonal flips. We augment this triangulation by associating, with each vertex, the value which was associated with it as a pixel in the image.
- Develop correlations between the vertices of augmented triangulations adjacent along the stream axis; these relations can be viewed as segments of splines connecting vertices along that axis. These splines, and the values associated with the vertices through which they pass, will be further compressed, but our focus in this paper is on the correlation method by which the spline segments are determined, so we do not elaborate that compression process here.

The matching decompression process reconstructs approximations of the vertices derived from points of interest, their associated values, and the corresponding sub-boundaries. Their CDT is then constructed and an approximation of the initial image is created through interpolation within the triangles of that triangulation.

Our correlation process draws upon the work of Shinagawa and Kunii [13] and Habuka and Shinagawa [14]. They provide a method, known as the Critical Point Filter (CPF) algorithm, employing multi-resolutional filters which, with each application to an image, reduce the resolution of the image by a factor of two in each dimension while attempting to preserve critical points. The result is a resolution hierarchy which is then traversed in reverse, matching points at low resolution and using these matches to guide the match at the next level.

The CPF algorithm has been incorporated into a commercial video compression product marketed by FrameFree, Inc., and our work grew out of a project, itself a continuation of work performed under the NSF grant mentioned above, seeking to improve that product. A drawback of the CPF algorithm is that it follows a rigid resolution reduction program which ignores the content of the image being processed. Motivated by a comment by James Sethian, [15], we decided to explore a more flexible approach which pursues similar goals but is guided by the image content, as represented by an initial triangulation using points of interest as its vertices. We decided to explore the use of Gaussian and other smoothing filters, in place of the CPF filters, and to use thinning of the triangulations in place of rigid grid reduction. As a matter of computational efficiency it became desirable to proceed with calculations based upon these derived triangulations, rather than calculations referenced back to the image; hence the development of the processes and structures reported here.

The CPF algorithm, and our adaptations of it, use triangulations to guide the matching of points of interest in separate images. Broadly, the distortion of the triangulation created by proposed matches is used to define a cost function which the matching process attempts, subject to some additional conditions, to minimize.

1.2 Augmented Triangulations

The process by which we will associate a polynomial surface with each vertex of a triangulation can be carried out in quite general settings: given a set of “vertices” and a vector space of potential “values”, it suffices to have functions which associate each vertex with a value and with a “link” – a circularly ordered finite set of distinct vertices. If an “orientation” is available by

which an initial element can be identified in each link, that information can be incorporated into the surfaces, and if distances between vertices are defined they can be reflected, imperfectly, in the surfaces. Of course, the utility of such a process depends upon how well the surfaces associated with such a set of vertices reflect properties of that set’s related structures which are of interest.

The requirements just listed, and in particular the notion of a link, were abstracted from augmented meshes of 2D surfaces, in particular triangulations, and we now focus on such triangulations (generalizations to higher dimensions are discussed briefly in Section 6). By triangulating just their pixels, images can be represented as augmented planar triangulations, a desirable quality because planar triangulations are quite well behaved, in contrast with triangulations, even of 2D surfaces, embedded in higher dimensions. In the same vein, triangulations of 2D surfaces in spaces of a given dimension can sometimes be viewed as augmented triangulations in a lower dimension by transferring some of the dimensional values of the vertices to augmentation values; an example is the height fields often used with terrain data.

1.3 Standard and Associated Surfaces

For use in the process of associating surfaces with vertices we define a collection of bivariate (2D) polynomial “basis” functions; these functions are orthogonal and the surfaces generated as linear combinations of them will be called “standard surfaces”. We develop a process for associating a unique standard surface with a given vertex and refer to a surface so associated as its “associated surface”. The surface associated with a vertex V of valence N is constructed as a linear combination of the first N of these basis functions. (The *valence*, or *degree*, of a vertex is the number of other vertices to which it is connected by an edge; the basis functions are described in Section 2.1 and the standard and associated surfaces are defined in Section 2.2 and Section 3 respectively.) The coefficients in the linear combination defining an associated surface are derived from the values at V and at the vertices with which V is connected, and are calculated using the Discrete Fourier Transform (the “DFT”, also referred to as the Finite Fourier Transform). Referencing this associated surface, processes such as calculation of curvature, classification of critical points, or the application of smoothing kernels can be carried out.

1.4 A Hierarchy of Triangulations

Smoothing an image eliminates small features and brings larger features into prominence, resulting in an

increase, in the sense of scale space theory – see, for example, Lindberg’s monograph [16] – of the scale of that image. Because fewer pixels are required to resolve these larger features, it is reasonable (and computationally efficient) to reduce the size of the smoothed image, for example by making proportional reductions in the number of rows and columns of pixels. We implement a similar process on triangulations: smoothing referencing the associated surfaces increases their scale and eliminating some of the vertices through thinning (coarsening, simplification, decimation) reduces their size, i.e., the number of vertices.

By repeated applications of these processes – smoothing followed by the removal of vertices which, at that scale, have little influence on their neighbors, we construct a sequence of triangulations of increasing scale but decreasing size - an hierarchical scale space of triangulations. In removing vertices, preference is given to removals which do not change the topology or unduly distort the geometry of the triangulation. As a result, features of increasingly larger scale can be brought within the scope of the local tools, such as differentiation or Gaussian smoothing, available at each vertex by reference to the associated surface.

1.5 Classification of Vertices

The application which motivates this construction requires the correlation of “similar” vertices in two hierarchies such as those described above. Attributes of a vertex useful in this process can be determined from the surface which we associate with that vertex, but in addition, useful attributes can be calculated by, in essence, reversing the portion of the process by which the vertex-surface association was created. This process produces as attributes the Fourier coefficients of functions, described, for the 3D case, in Section 4.2. Those functions map the equally spaced points on the unit circle to points which separately correspond to the original vertices with respect to distance, angle, and height, respectively. As with the associated surface, these attributes are calculated using the DFT.

1.6 Notation

Throughout “N” will represent a positive integer; we will often think of N as its set of non-negative predecessors, $N = \{0, 1, \dots, N-1\}$, so \sum_N , summation over the set N, indicates a summation of terms indexed from 0 to N-1. For r real, $\lfloor r \rfloor$ indicates the **floor function**, the greatest integer less than or equal to r, and for n and m integers, $\delta_{m,n}$ is the **Kronecker delta**: $\delta_{m,n} = 1$ if $m = n$ and 0 otherwise.

We will generally focus on a single vertex, V, of valence N, and its **surrounding vertices** (those with which it shares an edge) V_0, V_1, \dots, V_{N-1} . We assume the

existence of an underlying coordinate system inherited from whatever space – e.g., \mathbf{R}^3 , the Euclidean space of three dimensions – was triangulated; that V_0 has been chosen, with respect to this coordinate system, in some systematic way (for example, lexicographically based upon its coordinates); and that the V_n have been indexed in counterclockwise order around the **link** of V (those edges of triangles for which V is a vertex of the triangle but not an endpoint of the edge). In our applications it really does not matter how V_0 is selected, how the orientation determining “counterclockwise” is defined, or even how the vertices are well ordered; what is important is that these choices be made in a consistent manner so that when vertices in different images are being compared, the choices will agree.

We label the values at V and V_n as v and v_n respectively, and the vector of values $(v_0, v_1, \dots, v_{N-1})$ as \mathbf{V}_N . As with \mathbf{V}_N , vectors and matrices will be indicated by bold uppercase letters, and their terms will be indicated by the corresponding lowercase letters. As usual, \mathbf{X}^T is the transpose of \mathbf{X} ;

We indicate as $\boldsymbol{\Omega}_N = (\omega_0, \omega_1, \dots, \omega_{N-1})$ the vector of N points (or angles), starting at zero degrees, equally spaced around the unit circle: $\omega_n = 2n\pi/N$ radians or $360n/N$ degrees.

1.7 The Discrete Fourier Transform

We make repeated use of the DFT, viewed in two distinct ways: as a method of interpolating a trigonometric polynomial through a finite set of points, and as a transform from the spatial to the frequency domain. But our use is limited to real trigonometric polynomials of the form $S(\theta) = \sum a_n \cos n\theta + b_n \sin n\theta$, with a_n and b_n real, defined on the unit circle. The sum is from 0 to $\lfloor N/2 \rfloor$, and includes at most N non-zero terms: for $n = 0$ or $n = N/2$ (N even) we take $b_n = 0$ because $\sin n\theta = \text{zero}$ for $n = 0$ or θ in $\boldsymbol{\Omega}_N$ (where $(N/2)\omega_k = (N/2)(2k\pi/N) = k\pi$ when $n \neq 0$). Thus if N is even there are $2(N/2+1) - 2 = N$ terms and when N is odd there are $2((N-1)/2+1) - 1 = N$ terms.

The notation in terms of the functions $\cos n\theta$ and $\sin n\theta$ and real numbers a_n and b_n is convenient in that it presents the coefficients without redundancy, and in a grouping appropriate when spectral components are being considered, but is in some other cases awkward. To simplify expressions in those cases we re-label the functions and coefficients as follows:

$$\begin{aligned} S_0(\theta) &= \cos 0\theta = 1, & c_0 &= a_0; \text{ and for } n > 0 \\ S_{2n-1}(\theta) &= \cos n\theta, & c_{2n-1} &= b_n; \\ S_{2n}(\theta) &= \sin n\theta, & c_{2n} &= a_n. \end{aligned}$$

Note that the always zero $\sin 0\theta$ term has been omitted. With this notation the expression for $S(\theta)$ becomes:

$$S(\theta) = \sum_N c_n S_n(\theta).$$

The real values $\{c_n\}$ are the Fourier coefficients; as a vector $\mathbf{C}_N = (c_0, c_1, \dots, c_{N-1})$ they are calculated as

$$\mathbf{C}_N = \mathbf{U}_N \times \mathbf{M}^T \times \mathbf{J}$$

where \mathbf{U}_N is an N-fold vector of values on the unit circle which are being interpolated, \mathbf{M} is the N x N matrix $\mathbf{M}_{m,n} = S_m(\omega_n)$, and \mathbf{J} is a diagonal N x N “normalizing” matrix $\mathbf{J}_{m,n} = (\delta_{m,n} (2 - \delta_{m,0} - \delta_{m,N-1} \delta_{N,2 \lfloor N/2 \rfloor})) / N$. (We review this calculation in an Appendix.)

2 Standard Surfaces

We can, and now do, view the functions $S_0(\theta)$, $S_1(\theta), \dots$, labeled above as defined on the unit circle and hence as functions of x and y, defined when $x^2 + y^2 = 1$, i.e., as functions of x and y where $x = \cos \theta$ and $y = \sin \theta$ for some angle θ . We begin by expressing $\cos n\theta$ and $\sin n\theta$ in terms of powers of $\cos \theta$ and $\sin \theta$, i.e., as functions of x^n and y^n . We then extend these functions to the plane to create the “standard” surfaces which will be used to create the surfaces associated with vertices.

2.1 The Basis Functions $S_n(x,y)$

Applied to $\cos n\theta$ and $\sin n\theta$ the multiple angle formulas yield:

$$\begin{aligned} \cos n\theta &= \cos(n-1)\theta \cos \theta - \sin(n-1)\theta \sin \theta \\ \sin n\theta &= \sin(n-1)\theta \cos \theta + \cos(n-1)\theta \sin \theta. \end{aligned}$$

For S_n as before ($S_0 = 1$ and, for $n > 0$, $S_{2n-1} = \cos n\theta$, $S_{2n} = \sin n\theta$), substituting $x = \cos \theta$ and $y = \sin \theta$ in these formulas yield recursions: $S_{2n} = xS_{2n-2} + yS_{2n-3}$ and $S_{2n+1} = xS_{2n-1} - yS_{2n}$, whence

$$\begin{aligned} S_0 &= 1, \\ S_1 &= x, \\ S_2 &= y, \\ S_3 &= x^2 - y^2, \\ S_4 &= 2xy, \\ S_5 &= x^3 - 3xy^2, \\ S_6 &= 3x^2y - y^3, \\ S_7 &= x^4 - 6x^2y^2 + y^4, \\ S_8 &= 4x^3y - 4xy^3, \\ S_9 &= x^5 - 10x^3y^2 + 5xy^4, \\ S_{10} &= 5x^4y - 10x^2y^3 + y^5, \\ S_{11} &= x^6 - 15x^4y^2 + 15x^2y^4 - y^6, \\ S_{12} &= 6x^5y - 20x^3y^3 + 6xy^5, \\ S_{13} &= x^7 - 21x^5y^2 + 35x^3y^4 - 7xy^6, \\ S_{14} &= 7x^6y - 35x^4y^3 + 21x^2y^5 - y^7, \\ S_{15} &= x^8 - 28x^6y^2 + 70x^4y^4 - 28x^2y^6 + y^8, \\ S_{16} &= 8x^7y - 56x^5y^3 + 56x^3y^5 - 8xy^7, \end{aligned}$$

and so forth. We will use these functions $\{S_n\}$ to extend any trigonometric polynomial from the circle to the plane: $S(\theta) = \sum c_n S_n(\theta)$ becomes $S(x,y) = \sum c_n S_n(x,y)$.

Note that there are extensions of $\cos n\theta$ and $\sin n\theta$ alternative to those above depending upon if and how one uses the identity $\cos^2(\theta) + \sin^2(\theta) = 1$. For example, the Chebyshev polynomials express $\cos n\theta$ and $(\sin n\theta)/y$ solely in terms of x: the Chebyshev polynomials of the first kind corresponding to our S_3, S_5 , and S_7 are

$$\begin{aligned} T_2 &= x^2 - (1-x^2) = 2x^2 - 1; \\ T_3 &= x^3 - 3x(1-x^2) = 4x^3 - 3x; \text{ and} \\ T_4 &= x^4 - 6x^2(1-x^2) + (1-x^2)^2 = 8x^4 - 8x^2 + 1, \end{aligned}$$

while the Chebyshev polynomials of the second kind which correspond to $S_6/y, S_8/y$, and S_{10}/y are

$$\begin{aligned} U_2 &= 3x^2 - (1-x^2) = 4x^2 - 1; \\ U_4 &= 4x^3 - 4x(1-x^2) = 8x^3 - 4x; \text{ and} \\ U_6 &= 5x^4 - 10x^2(1-x^2) + (1-x^2)^2 = 16x^4 - 12x^2 + 1. \end{aligned}$$

We have chosen to not use such substitutions in order to maintain approximately equal contributions from both x and y, so that the surfaces formed by extending these functions off the unit circle will tend toward radial symmetry.

Note that, while our standard surfaces, $\{S_n\}$, provide a basis for the surfaces which we wish to generate, they do not span all bivariate polynomials: the space of bivariate polynomials of degree less than or equal to d has dimension $(d+1)(d+2)/2$, while our corresponding standard surfaces number only $2d+1$. In particular, the polynomial $x^2 + y^2$ is not spanned by $\{S_n\}$. (For reasons discussed in the next section, we will sometimes use $x^2 + y^2$ as a basis function, using it to replace the constant function ‘1’.)

2.2 The Standard Surfaces S_N and S_{PN}

In the context of an augmented triangulation or class of such triangulations, we take as **standard surfaces** all linear combinations of the basis functions $S_n(x,y)$ described above, with the coefficients of those linear combinations being drawn from the vector space from which the values at the vertices were drawn. We indicate as \mathcal{S}_N a standard surface created from just the first N basis functions. Such a standard surface is uniquely determined by its values on the equally spaced points of Ω_N : for $u_n = \mathcal{S}_N(\omega_n)$ and $\mathbf{U}_N = (u_0, u_1, \dots, u_{N-1})$, the DFT provides N Fourier coefficients c_0, c_1, \dots, c_{N-1} such that for $S_N(\theta) = \sum_N c_n S_n(\theta)$, $S_N(\Omega_N) = \mathbf{U}_N$. It follows (because the basis functions are independent) that on the plane $\mathcal{S}_N(x,y) = \sum_N c_n S_n(x,y)$. We indicate a standard surface determined from the N values \mathbf{U}_N by $\mathcal{S}_N(\mathbf{U}_N)$.

For x and y real, $S_n(x,y)$ is real and hence the graph of $z = S_n(x,y)$ is a 2D surface in \mathbf{R}^3 . If \mathcal{V} is the space from which the values of an augmented triangulation are drawn, then the graph of $v = \mathcal{S}_N(\mathbf{U}_N)(x,y)$ will be a surface in $\mathbf{R}^2 \times \mathcal{V}$. Thus for example in the case of a triangulation of a color image with values in the RGB color space, $\mathcal{S}_N(\mathbf{U}_N)$ will be a 2D surface in the five dimensional space $X \times Y \times \text{RGB}$ – as is the triangulated color image itself.

The standard surface $\mathcal{S}_N(\mathbf{U}_N)$ passes through a value c_0 at the origin equal to the average of the values of \mathbf{U}_N . In the process of associating a standard surface with a particular vertex V the values from which \mathbf{U}_N is derived are normalized so that in effect the value at V is zero. In order that the associated surface, which passes through each value in \mathbf{U}_N , pass through zero at V it is necessary that the standard surface be zero at the origin. To achieve this, we modify \mathcal{S}_N by replacing the basis function “1” with the basis function “ $x^2 + y^2$ ” thereby replacing the constant term $c_0 S_0$ with $c_0(x^2 + y^2)$ – in effect adding $c_0(x^2 + y^2 - 1)$ – to create a “parabolized” version \mathcal{SP}_N of \mathcal{S}_N . (The surface $z = c_0(x^2 + y^2 - 1)$, with c_0 non-zero, forms a parabolic bowl, its direction determined by the sign of c_0 , as illustrated in the accompanying surface plots. Adding it to another surface distorts that surface with progressively greater effect as the distance from the origin increases.) Note that the parabolized surface remains a polynomial and in fact, it’s degree will usually not be increased: the special case of boundary vertices, which could be problematic, will be treated so that c_0 is zero there, and for interior vertices it is only when all of the points (ω_n, u_n) lie in a plane (one which does not pass through the origin) that parabolizing causes an increase in degree. Given the manner in which the values in \mathbf{U}_N are created through interpolation over most likely different distances, this case will be uncommon.

To calculate the associated surfaces for vertices of valence N we define \mathcal{S}_{XN} to be the vector composed of the first N $\{S_n\}$, $\mathcal{S}_{XN} = (S_0, \dots, S_{N-1})$; and we define \mathcal{SP}_{XN} to be equal to \mathcal{S}_{XN} except with initial entry $x^2 + y^2$ in place of $S_0 = 1$. (In our implementation we limit the valence of vertices to 16 or less, which seems a practical number for our application. There is of course no theoretical impediment to continuing the recursively defined list in Section 2.1 and extending accordingly the calculation described here.)

Given a vector \mathbf{U}_N of values, the standard surface \mathcal{S}_N passing through those values over the points ω_n is calculated as:

$$\mathcal{S}_N(\mathbf{U}_N) = \mathbf{C}_N \times \mathcal{S}_{XN}^T$$

and its parabolized form as:

$$\mathcal{SP}_N(\mathbf{U}_N) = \mathbf{C}_N \times \mathcal{SP}_{XN}^T$$

where, as above, \mathbf{C}_N is the vector of Fourier coefficients. Combining these formulas with the calculation of the Fourier coefficients \mathbf{C}_N yields, for example:

$$\mathcal{S}_N(\mathbf{U}_N) = \mathbf{U}_N \times \mathbf{M}_N^T \times \mathbf{J}_N \times \mathcal{S}_{XN}^T.$$

2.3 Example with $N = 4$

Values $\mathbf{U}_N = (u_0, u_1, u_2, u_3, u_4)$ produce corresponding Fourier coefficients $c_0 = (u_0 + u_1 + u_2 + u_3)/4$, $c_1 = (u_0 - u_2)/2$, $c_2 = (u_1 - u_3)/2$, and $c_3 = (u_0 - u_1 + u_2 - u_3)/4$, so the standard surface over four points is

$$\mathcal{S}_4(\mathbf{U}_N) = c_0 + c_1 x + c_2 y + c_3(x^2 - y^2)$$

while the parabolized surface is

$$\mathcal{SP}_4 = c_1 x + c_2 y + (c_0 + c_3)x^2 + (c_0 - c_3)y^2.$$

At the end of this paper we present plots of a sampling of the basis functions defined in Section 2.1 and the effect of adding a “parabolizing” term equal to $x^2 + y^2$. These plots were created using the Wolfram “plot” tool at the web site [17].

3 Associated Surfaces

The standard surfaces are defined with respect to the uniformly spaced points Ω_N on the unit circle and values \mathbf{U}_N at those points. A vertex V of valence N will have neighbors V_0, V_1, \dots, V_{N-1} , in that order, along its link, and values $\mathbf{V}_N = (v_0, v_1, \dots, v_{N-1})$ at those points. We create an association of a standard surface \mathcal{S}_N with such a vertex by relating the values \mathbf{V}_N to a set of values \mathbf{U}_N which are then used to define \mathcal{S}_N . A standard surface associated with a vertex in this way is referred to as an *associated surface*.

3.1 Interior Vertices

To form an association of an interior vertex V with one of our standard surfaces we associate V_n directly with ω_n and use this association to interpolate a value u_n at ω_n . In our applications we interpolate linearly between V_n and V : $u_n = (v_n - v)/L_n$ for L_n the length of the edge VV_n . We use length measured in the ambient vector space, but length measured along the triangulated surface could be used. Further, other measures could be used: for example, where the vertices represent point masses, it might make more sense to interpolate based on the square of the distance. With this association of each V_n with ω_n and the calculation of the n -th entry of \mathbf{U}_N as $u_n = (v_n - v)/L_n$, the standard surfaces associated with V are $\mathcal{S}_N(\mathbf{U}_N)$ and $\mathcal{SP}_N(\mathbf{U}_N)$.

Note that by using $v_n - v$ in the definition of u_n we have in essence taken v to be our origin: if the v_n were themselves on the unit circle, the associated surfaces $\mathcal{S}_N(\mathbf{U}_N)$ and $\mathcal{S}_{PN}(\mathbf{U}_N)$ would pass through $v_n - v$. If we add v to the constant term of the parabolized surface $\mathcal{S}_{PN}(\mathbf{U}_N)$, the resulting surface will pass through both v at the origin and each v_n on the circle.

The polynomials $\mathcal{S}_N(\mathbf{U}_N)(x,y)$ and $\mathcal{S}_{PN}(\mathbf{U}_N)(x,y)$ represent smooth surfaces passing through the interpolated values derived from V . They provide a two dimensional Euclidean region which can be associated with V and within which traditional analysis can be performed. These surfaces clearly differ from the faceted surface often associated with a vertex and composed of the triangles which include that vertex.

We give no attention to the manner in which these associated regions, viewed with respect to a global coordinate system, could be considered to meet or overlap - rather, it is our intention to use scale expansion and resolution reduction to approach properties involving vertices not originally connected by an edge. In this way we hope to create and utilize an abstract "triangulation space" while avoiding the often difficult problems encountered when performing practical computations of what may be theoretically trivial intersections of surfaces. (For background on that problem as it arises in graphics, CAGD, and engineering analyses, with some solutions, see the project report [18].)

3.2 Boundary and Sub-Boundary Vertices

As an image pans, or as a foreground object reveals and conceals detail in the background, vertices which are interior to the triangulation of one image in a stream may appear as a boundary or sub-boundary vertex in the triangulation of a neighboring image. Because of this, we wish to allow correlation among interior, boundary, and sub-boundary vertices, especially if they do not seem to correspond well with vertices of their same type. Attempting such correlations based only upon the vertices neighboring a boundary vertex or those available without crossing a sub-boundary to a sub-boundary vertex can be expected to produce poor results, so we increase the vertices available by means of one or two reflections. To that end, we classify the extent to which boundary and sub-boundary vertices are "surrounded" by their neighbors as 100%, 50%, or 25% and correspondingly map their neighbors to a complete circle, a hemi-circle, and a quarter circle.

In the case of vertices mapped to semicircles or quarter circles, we "surround" that vertex by reflecting the half circle across its edge or by reflecting the quarter circle first across one edge, then the other. As a result the average value over all four vertices will be zero so

that "parabolizing" is not needed. In this process the valence of a vertex will be increased from N to $2(N-1)$ in the first case and to $4(N-1)$ in the second. Surfaces are now associated as in the case of an interior vertex.

Making comparisons between a surface associated by this process and a surface associated with a genuine interior vertex requires care, as does the use of other classifying attributes. Often additional information is available and can be used to indicate which hemi-circles and quarter circles should be compared.

4 The Classification of Vertices

As has been mentioned several times, an expected application of these considerations is the correlation of vertices in triangulations of different images. Such correlations are made based upon similarity of attributes associated with the vertices, and the procedures which we have been describing generate or can be used to generate several interesting data which can be considered as such attributes. These data fall into three categories: properties like spectral power/amplitude or spectral phase available because of the expression of the associated surface in terms of Fourier coefficients; additional properties which we are about to describe which characterize the differences between a particular vertex V and the generalized situation represented by the associated surface; and traditional properties of surfaces, such as curvature and critical points, which need no further discussion here.

4.1 Fourier Attributes of the Associated Surfaces

The associated surfaces are themselves invariant under translation and their spectral amplitudes and powers are invariant under rotation. Because of the stylized manner in which the surfaces are associated with a particular vertex, their association with rotation is limited, but usefully so: rotation, indeed any rigid transformation of a vertex V , preserves the values \mathbf{V} , the order of those values in the link of V , and the lengths of the edges VV_n . It may not preserve the starting point in the link, that is, the indexing used in listing the values in \mathbf{V} . Consequently, for a vertex of valence N , the associated values \mathbf{U}_N on the unit circle will be unchanged except that they may be rotated through an angle of $2k\pi/N$ radians for some integer k . Thus to compare two vertices V and V' to determine if V' is the image of V under a rigid transformation, we can compare their spectral phases, say ϕ_n and ψ_n . If V and V' are the same we should find that $N(\phi_n - \psi_n)/2\pi \approx k$ for some integer k and all n in N .

4.2 Additional Fourier Attributes of a Vertex

In the process of passing from a vertex to an associated surface we ignore a substantial quantity of

information potentially useful in the correlation of vertices. Much of this ignored information can be codified, separately, as Fourier attributes, using the same techniques we used for the surfaces. We do this by changing viewpoints: consider the problem, with reference to a vertex V of valence N , of mapping the points Ω_N on the circle to the vertices surrounding V (to the points themselves, not their values), rather than the previously considered problem of bringing the values at the surrounding vertices to the circle. We can create such a mapping from the circle by means of trigonometric polynomials simply by applying the DFT in each dimension. Since we are only interested in the Fourier properties of the resulting functions, rather than, for example, smoothly mapping the unit disc to a 2D surface incorporating the surrounding vertices, we need not be concerned with avoiding intermediate topological changes, e.g., by coordinating the parameterizations implicit in these maps.

For convenience we assume the subject vertices are located in \mathbf{R}^3 . Their positions can thus be represented by triples of real numbers, say $V = (a,b,c)$ and $V_n = (a_n, b_n, c_n)$. Different coordinate systems will produce different functions which may be more expressive of structure in certain kinds of data, but for our considerations the choice of coordinate system is not important, other than we assume that the system is centered on V , so $(a,b,c) = (0,0,0)$. Even though we do not require a specific coordinate system, it is helpful to have some system in mind; our terminology reflects a view of a cylindrical coordinate system (r,θ,z) , radius, angle measured in the plane perpendicular to the z axis, and elevation along the z axis respectively. We view the points Ω_N , as lying on the unit circle in this system: $\omega_n = (1, \omega_n, 0)$. Utilizing these axes separately, we create three functions $e(\theta)$, $d(\theta)$, and $c(\theta)$ which we refer to as expansion, distortion, and crenellation, respectively.

With respect to V , we define the **expansion** function $e(\theta)$ through Fourier interpolation of the values, for each n in N , $e(\omega_n) = a_n$. For vertices confined to the plane described with polar coordinates $e(\theta)$ expands or contracts the unit circle along the rays from the origin through the points on the circle. If compatibility with the interpolation of vertex values to the circle is deemed desirable, a coordinate system could be used in which the first coordinate is the distance from the origin; e.g., spherical coordinates.

In a similar manner, we define the **distortion** function $d(\theta)$ through Fourier interpolation of the values $d(\omega_n) = b_n$. In the case of polar coordinates, $d(\theta)$ distorts the circle, disrupting the equal spacing of the points ω_n by moving them (while preserving order) around the circle.

As expected, we define the **crenellation** function $c(\theta)$ through Fourier interpolation of the values $c(\omega_n) = c_n$. In the case of Cartesian or cylindrical coordinates, $c(\theta)$ introduces vertical displacements of points around the circle (crenellations are the battlements – parapets with periodic openings – frequently seen at the tops of medieval castles, and hence, metaphorically, shapes similar to them).

As indicated above, we are not interested in using these functions as maps; instead, we view the coefficients by means of which they are expressed as the Fourier transform of a function descriptive of the vertex V , specifically of its relationships to the surrounding vertices. These components, with their translational invariance and rotational coherence, provide a collection of attributes useful in classification of vertices or estimation of rotations which might be needed to bring associated vertices into alignment.

5 Development of Triangulation Hierarchies

As has been mentioned, a prime motivation for the constructions considered here was a desire to construct a hierarchy of triangulations through alternating processes of smoothing the data associated with a triangulation, thereby increasing the “scale” of that data, and thinning of the triangulation, thereby reducing its resolution, preferably in a manner reasonably coordinated with the increase in scale (so that “feature size” remains nearly constant). Thinning of triangulations is useful in other contexts, for example, the reduction of large data sets while limiting the loss of detail, and has been extensively studied elsewhere: see for example the paper [19] and Edelsbrunner’s monograph [11]. Delaunay triangulations are useful in this regard because the void (polygonal in the planar case; the property discussed here holds for Delaunay tetrahedralizations in any dimension) created by removing a single vertex from a Delaunay triangulation can be repaired locally: a Delaunay triangulation of that region, combined with the unmodified portion of the triangulation external to it, creates a Delaunay triangulation of the entire region. (This can be seen by considering the duality of the Delaunay triangulation with the downward facing portion of the convex hull of a certain parabolic set of points generated, as described, for example in the paper [20], by “lifting” its vertices into an added dimension: the Delaunay triangulation of the reduced set of vertices corresponds to the convex hull of the reduced set of lifted vertices, and in creating such a convex hull only the faces which contained the vertex which was removed will be affected.)

Thus thinning of a Delaunay triangulation can proceed by removing, one at a time, the vertices with the least influence on their surroundings until the desired

reduction in size is achieved. The constant term c_0 in the surface associated with a vertex can be used as a measure of its influence, as can, of course, more complicated measures.

For CDTs the situation is not so simple. Maintaining a constrained triangulation while thinning along a desirable sequence may require removal of more than one vertex at a time: [21], Figure 11.

6 Potential Generalizations

We have focused upon triangulations of 2D surfaces because of our interest in conventional images and to work out procedures without the complications of added dimensions. However, images created by slice sequences, computed tomography, magnetic resonance imaging, or the like, as well as temporal sequences, require or could benefit from treatment in an additional dimension. The theory of tessellations in three and higher dimensions, in particular that of Delaunay and weighted Delaunay tessellations, is well developed and algorithms exist which guarantee their creation, e.g., the star splaying of [20]. Iteration of the one dimensional Fourier transform provides transforms of higher dimensions: at each iteration the functions produced at the previous step become the “values” for the next step. Thus there seems to be no particular impediment to generalizing these results to higher dimensions.

Other triangulations, for example weighted Delaunay and regular triangulations have been studied [21], as have of course more general meshes. Applications of the methods developed here to such more general structures might be of interest.

In the context of increasing the “scale” of triangulated data, we have considered only smoothing the values associated with each vertex. Triangulations themselves can be smoothed, by moving, adding, or deleting vertices, and it would be interesting to combine such a process with that applied to the data. Also, with respect to eliminating vertices from triangulations, we have only considered thinning (successive removal of individual vertices accompanied by triangulation repair) but more general coarsening of a triangulation need not proceed so directly: moving or temporarily adding vertices can lead to ultimately better reductions in size. See the paper [22] for a discussion of smoothing of triangulations, as well as processes which can relate to coarsening.

Finally, our constructions, with respect to triangulations, refer to the vertices, not to the edges which connect them. Surface representations which utilize vertices but replace edges with specified curves should also be amenable to the approaches we have described. For such representations the distances between vertices might be calculated as distance along

the connecting curves, and the angles used in calculating the distortion function might be replaced by the angles formed by tangents to those curves at either end, and similar treatment might be afforded the remaining axis since in this case “elevation” might not be a useful measure. Additional information relating to behavior along the curves could be incorporated if needed. Even more general representations, in which vertices are replaced by control points, could be considered.

7 Future Work

Due to the interesting and challenging financial developments over the past year, the image compression project which motivated the considerations reported here is proceeding very slowly. As a result, we are not able to report completed results, nor to give comparisons of our techniques with other approaches. We hope that we will be able to do so in the future.

We also have not had the resources to investigate how determinations of quantities such as surface normal, curvature, and principal directions, calculated utilizing our associated surfaces, compare to calculations based upon other approaches. (As noted above, the relevant comparison would be repeatability between similar images and ease of calculation, not “accuracy”). As to these, we invite our readers, particularly those who have previously implemented versions of such calculations, to draw such comparisons.

Appendix: The Discrete Fourier Transform

We adopt here the view of the Fourier Transform as a method of interpolating a function through N given values at equally spaced points around the unit circle, one of them being at the angle 0. We assume we are given z axis values $\mathbf{U}_N = (u_0, u_1, \dots, u_{N-1})$ over the points, $\omega_0, \omega_1, \dots, \omega_{N-1}$ respectively, of Ω_N .

A.1 Orthogonality

As functions of N equally spaced angles, S_0 through S_{N-1} are *orthogonal* (with respect to the L^2 inner product and the finite measure taking the value $1/N$ at the specified angles and zero elsewhere): none is identically zero and $\sum_N S_p(\omega_n)S_q(\omega_n) = 0$ if $p \neq q$. They are not *orthonormal*, i.e., $\sum_N S_p(\omega_n)^2$ is not 1 (except for $N = 1$), but they are easily normalized because for $p = 0$ or $p = N-1$ and N even, $\sum_N S_p(\omega_n)^2 = N$, and otherwise $\sum_N S_p(\omega_n)^2 = N/2$. (See any textbook discussion of the DFT for proofs or at least references to proofs.)

A.2 Calculation of the Fourier Coefficients

Assume $N > 1$, let \mathbf{M} be the $N \times N$ matrix with entries $\mathbf{M}_{m,n} = S_m(\omega_n)$, and let \mathbf{J} be the $N \times N$ “normalizing” matrix $\mathbf{J}_{m,n} = (\delta_{m,n} (2 - \delta_{m,0} - \delta_{m,N-1}) \delta_{N,2 \lfloor N/2 \rfloor}) / N$. Note that the (m,n) -th entry of $\mathbf{M} \times \mathbf{M}^T$ is $\sum_N S_m(\omega_p)S_n(\omega_p)$,

which, by the comments above, is zero off the diagonal and $1/\mathbf{J}_{m,m}$ on it, whence:

$$\mathbf{M} \times \mathbf{M}^T \times \mathbf{J} = \mathbf{I}.$$

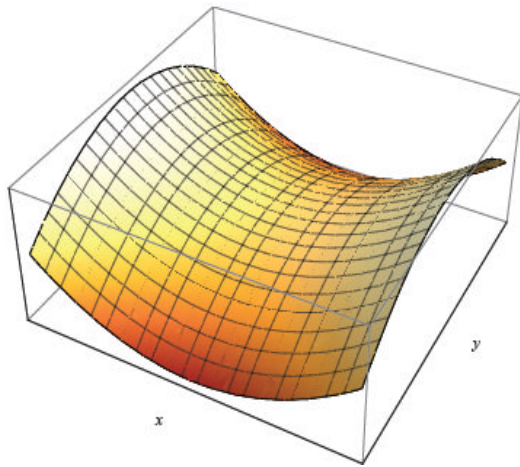
Here, as usual, \mathbf{I} is the identity matrix. The requirement that $S(0)$ pass through the given values u_n over the points ω_n , that is, that $S(\Omega_N) = \mathbf{U}_N$, provides N simultaneous equations in N unknown coefficients $\{c_n\}$ which can be expressed as $\mathbf{U}_N = \mathbf{C}_N \times \mathbf{M}$. Multiplying on the right by $\mathbf{M}^T \times \mathbf{J}$ and applying the identity above yields a solution for \mathbf{C}_N :

$$\mathbf{C}_N = \mathbf{U}_N \times \mathbf{M}^T \times \mathbf{J}.$$

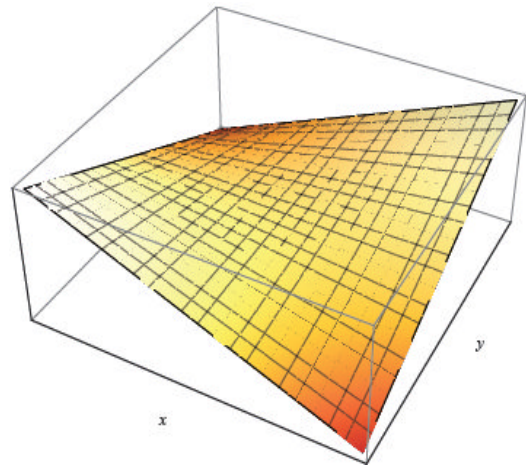
In an implementation $\mathbf{M}^T \times \mathbf{J}$ can be generated very efficiently from lookup tables. Even so, simply carrying out the indicated multiplications and additions provides a “brute force” DFT. There are a number of more efficient implementations, known as Fast Fourier Transforms (FFT), which make use of the symmetries in this calculation and which can be optimized for particular hardware configurations.

References

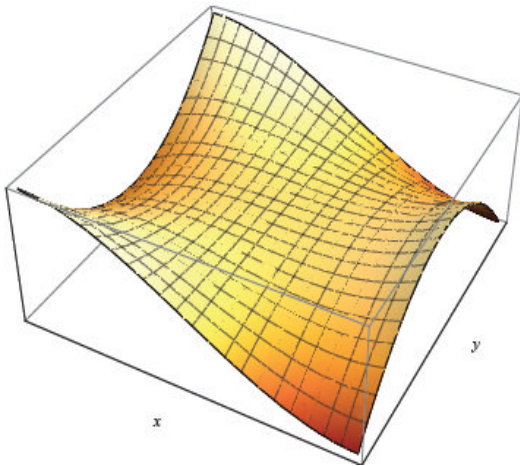
- [1] Ø. Hjelle and M. Daehlen; *Triangulations and Applications*, Springer 2006, pp. 5-6.
- [2] M. Meyer, M. Desbrum, P. Sehröder, and A. H. Barr, *Discrete differential-geometry operators for triangulated 2-manifolds*, In *Visualization and Mathematics III*, H. C. Hege and K. Polthier, Eds., Springer-Verlag, 2003, pp. 35-57.
- [3] X. Jiao and H. Zha, *Consistent computation of first- and second-order differential quantities for surface meshes*. In *ACM Solid and Physical Modeling Symposium*, 2008, pp. 159-170.
- [4] J.-L. Maltret and M. Daniel, *Discrete curvatures and applications: a survey*, Rapport de recherche Laboratoire des Sciences de l'Information et des Systemes, March 2002.
- [5] T. D. Gatzke and C. M. Grimm, *Estimating curvature on triangular meshes*, *International Journal of Shape Modeling*, 2005, pp. 1-29.
- [6] Z. Xu and G. Xu, *Discrete schemes for Gaussian curvature and their convergence*, *Computers and Mathematics with Applications*, Elsevier, 2009, pp. 1187-1195.
- [7] J. Goldfeather and V. Interrante, *A novel cubic-order algorithm for approximating principal direction vectors*, *ACM Transactions on Graphics*, V23, 2004, pp. 45-63.
- [8] A. I. Bobenko and B. A. Springborn, *A discrete Laplace-Beltrami operator for simplicial surfaces*, *Discrete and Computational Geometry*, V38, 2007, pp. 740-756.
- [9] E. Sukan and E. Appleboim, *Metric methods in surface triangulation*; In *Mathematics of Surfaces XIII*, E. R. Handcock, et al. Eds., Springer-Verlag 2009; pp. 335-355.
- [10] J. R. Shewchuk, *A condition guaranteeing the existence of higher-dimensional constrained Delaunay triangulations* *Proceedings of the Fourteenth Annual Symposium on Computational Geometry*, ACM, 1998, pp 76-85.
- [11] H. Edelsbrunner; *Geometry and Topology for Mesh Generation*, Cambridge University Press, 2001, pp. 68-88.
- [12] C. Dyken and M. S. Floater, *Preferred directions for resolving the non-uniqueness of Delaunay triangulations*, *Computational Geometry*, V34, 2006, pp. 696-101.
- [13] Y. Shinagawa and T. L. Kunii; *Unconstrained automatic image matching using multiresolutional critical-point filters*. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(9):1998, pp. 994-1001.
- [14] K. Habuka and Y. Shinagawa; *Image interpolation using enhanced multiresolution critical-point filters*, *International Journal of Computer Vision* 58(1), 2004, pp. 19-35.
- [15] J. Sethian; Personal communication, May 2008.
- [16] T. Lindberg, *Scale Space Theory in Computer Vision*, Springer 1993.
- [17] S. Wolfram; WolframAlpha.com, 2009.
- [18] T. J. Peters, J. Bisceglia, D. R. Ferguson, C. M. Hoffmann, T. Maekawa, N. M. Patrikalakis, T. Sakkalis, and N. F. Stewart; *Computational topology for regular closed sets (within the I-TANG) project*, *Topology Atlas Invited Contributions* 9 (2004) no. 1, 12pp.
- [19] W. J. Schroeder, J. A. Zarge, and W. E. Lorensen; *Decimation of triangular meshes*, *Computer Graphics*, 1992, pp. 65-70.
- [20] J. R. Shewchuk; *Star splaying: an algorithm for repairing Delaunay triangulations and convex hulls*, *Proceedings of the Twenty-First Annual Symposium on Computational Geometry*, ACM, 2005, pp. 237-246.
- [21] J. R. Shewchuk, *Updating and constructing constrained Delaunay and constrained regular triangulations by flips*, *Proceedings of the Nineteenth Annual Symposium on Computational Geometry*, ACM, 2003, pp 181-190.
- [22] B. M. Klingner and J. R. Shewchuk; *Aggressive tetrahedral mesh improvement*, *Proceedings of the 16th International Meshing Roundtable*, 2007, pp 3-23.



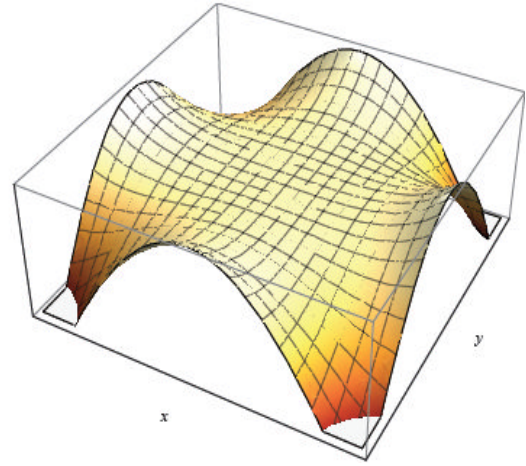
$$S3 = x^2 - y^2$$



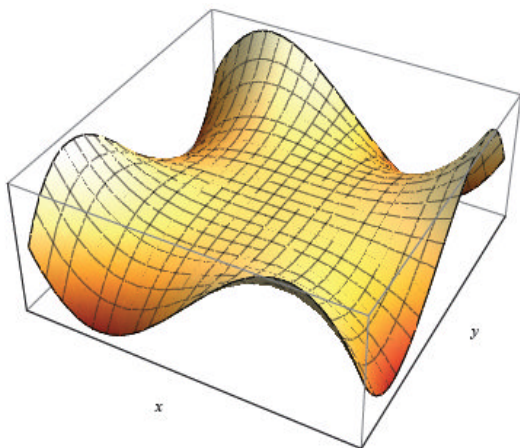
$$S4 = 2xy$$



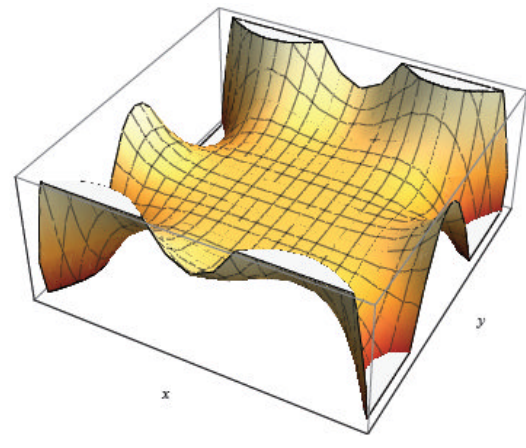
$$S5 = x^3 - 3xy^2$$



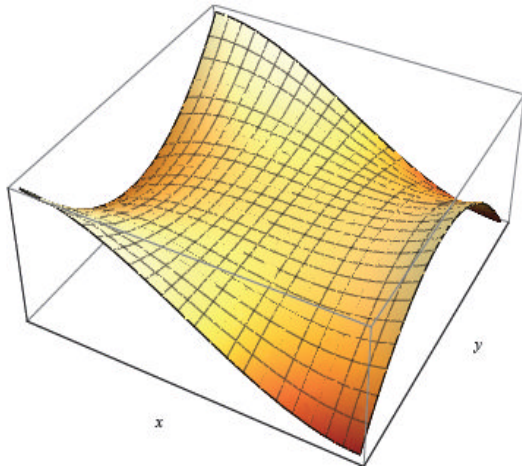
$$S7 = x^4 - 6x^2y^2 + y^4$$



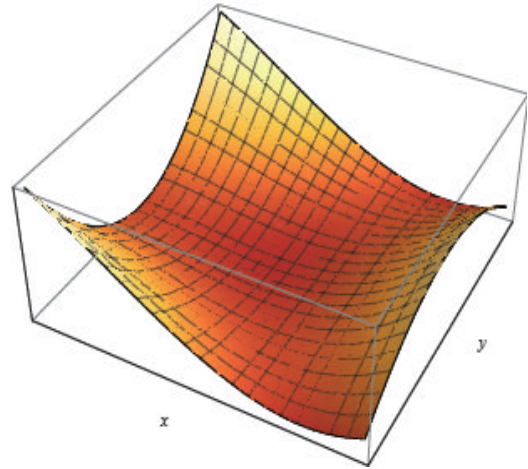
$$S8 = 4x^3y - 4xy^3$$



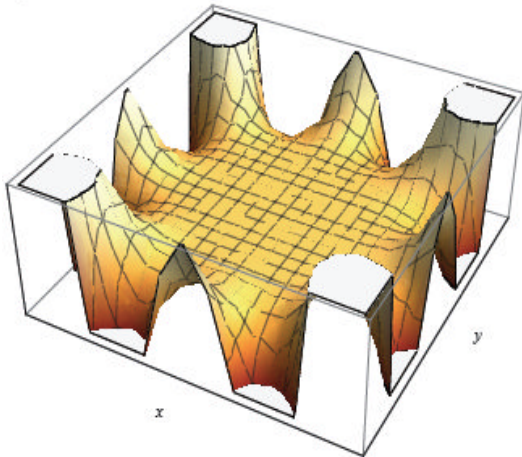
$$S11 = x^6 - 15x^4y^2 + 15x^2y^4 - y^6$$



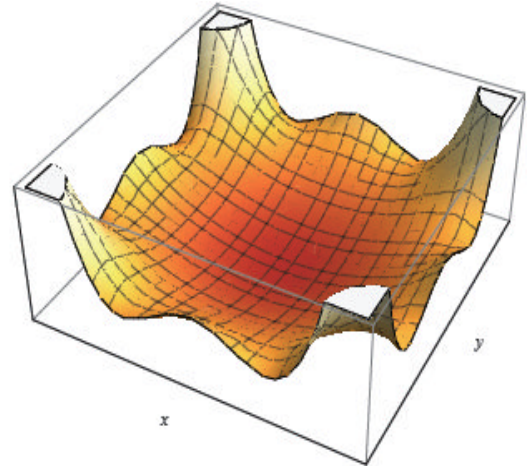
$$S5 = x^3 - 3xy^2$$



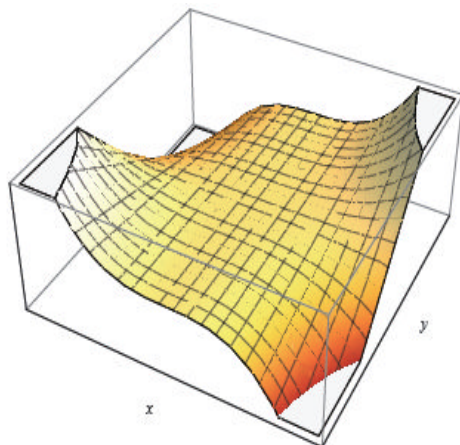
$$SP5 = S5 + x^2 + y^2$$



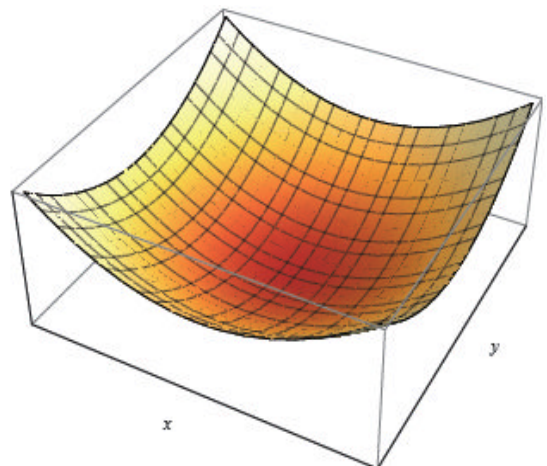
$$S15 = x^8 - 28x^6y^2 + 70x^4y^4 - 28x^2y^6 + y^8$$



$$SP15 = S15 + .2(x^2 + y^2)$$



$$S16 = 8x^7y - 56x^5y^3 + 56x^3y^5 - 8xy^7$$



$$\text{Parbola: } x^2 + y^2$$