NOVEL ANISOTROPIC MULTIDIMENSIONAL CONVOLUTIONAL FILTERS FOR DERIVATIVE ESTIMATION AND RECONSTRUCTION

David J Thornley, Member IEEE

Department of Computing, Imperial College London

ABSTRACT

The Savitzky–Golay convolutional filter matches a polynomial to even-spaced, one dimensional data and uses this to measure smoothed derivatives. We re-examine the fundamental concept behind this filter, and generate a formulation approach with multidimensional, heterogeneous, anisotropic basis functions to provide a general smoothing, derivative measurement and reconstruction filter for arbitrary point clouds using a linear operator in the form of a convolution kernel. This novel approach yields filters for a wide range of applications such as robot vision, medical volumetric time series analysis and numerical differential equation solution on arbitrary meshes or point clouds without resampling. The urge to extend polynomial filters to higher dimensions is obvious yet previously unfulfilled. We provide a novel complete, arbitrary-dimensional approach to their construction, and introduce anisotropy and irregularity.

Index Terms— Multidimensional digital filters, Adaptive filters, polynomial approximation, signal reconstruction, differentiation

1. INTRODUCTION

In 1964, Savitzky and Golay formulated a one-dimensional convolutional kernel of odd size for application to evenly spaced data which gives an equivalent smoothed sample value or an estimated derivative at the central point [1]. Their key insight was that fitting a polynomial to evenly spaced data and measuring the n^{th} ($n \geq 0$) derivative of the fit can be achieved using a convolutional kernel. A kernel may be synthesized so long as there are at least as many data points as terms in the polynomial model.

The Savitzky-Golay approach in one dimension has been extended to include even-sized kernels [2], measurements taken at arbitrary positions within the kernel [3], and application to systems on more general spacing [4]. Luo *et al* [5] analyse the frequency response of one dimensional Savitkzy-Golay filters.

We find extensions of the one dimensional approach to two dimensions, using either simplified basis sets [6] analysed in a manner comparable with that of [1], or using orthogonal polynomials which allow the generation of simple closed forms for smoothing in two dimensions [7] which demonstrates superior properties [8], and for measuring smoothed derivatives [9]. In reading Kuo, Wang and Pickup, note that they refer to the elements of the convolutional kernel as "weighting factors". In this present work, we introduce free weighting of the variance at data points, we distinguish between elements of a convolution kernel, and weighting applied to data points which modulates their significance in the solution for those kernel elements. Kuo *et al* [7] explain that their multidimensional approach is cumbersome in more than two dimensions, and while it can be decomposed into a series of one dimensional filters, these results operate at the centre of an odd sized patch aligned with the axes. This cumbersome nature is inevitable when producing closed forms for kernel values, which is achieved using a range of different orthogonal polynomial sets among the authors we cite. The results are valuable due to their efficiency of implementation, but the barriers to futher development limit the benefit to signal processing practitioners.

The value of polynomial filters lies in the fact that feature detection requires detail to be maintained despite smoothing. Traditional averaging or median filters perform smoothing to the detriment of trends in the data. In one dimension, this means for example that peaks and troughs are de-emphasized or re-shaped. The Savitzky–Golay polynomial filter improves the result by following trends at scales dictated by the interplay between the order of the polynomial used and the size of the filter's footprint in the data. When working in more than one dimension, more complex trends in the data become apparent, such as edges in 2D images. It is desirable to smooth such images while retaining salient features such as edges and corners. When using a simple smoothing filter, it is necessary to limit the extent of the support in the direction of the detail we wish to retain. In contrast, the use of a polynomial modeling filter enables the retention of detail to be decoupled from the extent of the support.

The order of the polynomials used in these filters is a compromise between retention of detail favouring a higher order, and rejection of noise favouring a lower order. This choice is commonly *ad hoc*, but identification of an appropriate test statistic [10] has allowed tuning for specific data sources (*e.g.* [11]).

Freemand and Adelson [12] describe steerable filters in which an arbitrarily oriented filter may be constructed from a finite basis set of filters, and Yang *et al* [13] describe a non-linear filter – later enhanced by Greenburg and Kogan [14] – which orients its kernel according to a novel measure of anisotropy. A filter based on matching a multidimensional spline attempts to circumvent this requirement by incorporating an approximation to the detail into the patch.

Feasibility, accuracy and stability of calculation of a general filter is crucial. The earliest work referenced by Savitzky and Golay is that of Kerawala from 1941 [15] (which itself references Birge and Shea from 1924 regarding the propagation of errors) in which an accelerated means is formulated for calculating the least-squares fit of a polynomial to data in which the "values of the independent variable form an arithmetic series." The Moore-Penrose pseudo-inverse [16, 17] gives us a conceptually simpler solution to least squares problems.

The pseudo-inverse has been applied in solution for a 2D patch in a similar spirit to Savitzky and Golay's original work in a result reported by Krumm on the Web [18]. This uses a polynomial in x^m and y^n up to a maximum total order $m + n \leq k$ for filtering and derivative measurement at the centre of a patch of odd or even size, with the proviso that an even sized patch gives the measure at the centre of the central four points. This adds the use of even-sized kernel patches in two dimensions to the state of the art in two dimensions represented by [9] and [7].

We use the pseudo-inverse in the subset of our formulations which results in integer valued and rational expressions for the sake of simplicity and speed. The more generally stable QR factorization [19] is appropriate for filters which use non-integer weighting or irregular data placement, in which the direct construction of the pseudo-inverse is numerically problematic. By synthesizing a simple, uniform formulation which does not rely on establishing and manipulating orthogonal polynomials, we provide general filters in multiple dimensions.

2. POLYNOMIAL INSTRUMENTATION BY CONVOLUTION

To extend the one dimensional case to two dimensions we allow the parameters of the polynomial in any given dimension themselves to vary as polynomials in each additional dimension. We believe this is the first occasion on which this has been expressed this for the purpose of filter construction. This process can be continued to any number of dimensions, resulting in a model comprising the product of the polynomials in each dimension. We demonstrate the derivation in one and two dimensions, and note the simple extension required for each additional dimension, resulting in an expression for the derivative in arbitrary dimensions.

We begin with a one-dimensional filter. Note that in all these derivations, the coordinates do not have to lie on a regular lattice. This is discussed later. Using a polynomial model $f(x)$ of order n as follows:

$$f(x) = \sum_{p=0}^{n} a_p x^p \tag{1}$$

Let the x coordinate of the i^{th} data point be x_i, $1 \leq i \leq k$ and its sampled value be g_i. The solution for a_i we require optimizes the fit of $f(x_i)$ to samples g_i in the simplified least squares sense, *i.e.* measured on constant x_i (though not necessarily evenly spaced). This solution is expressed as follows:

$$\epsilon_i = f(x_i) - g_i \tag{2}$$

$$u = \sum_{i=1}^{k} {\epsilon_i}^2 \tag{3}$$

$$\frac{\partial u}{\partial a_p} = 0; 0 \leq p \leq n \tag{4}$$

This is subsumed in the well-known pseudo-inverse solution of the following form [17]:

$$f(x_i) = g_i \tag{5}$$

$$B := \begin{pmatrix} 1 & x_1 & \cdots & x_1^p & \ldots & x_1^n \\ \vdots & \vdots & & \vdots & & \vdots \\ 1 & x_i & & x_i^p & & x_i^n \\ \vdots & \vdots & & \vdots & & \vdots \\ 1 & x_k & & x_k^p & & x_k^n \end{pmatrix} \tag{6}$$

$$a := (a_0 \cdots a_p \cdots a_n)^{\mathrm{T}} \tag{7}$$

$$Ba = g \Rightarrow B^{\mathrm{T}}Ba = B^{\mathrm{T}}g \tag{8}$$

$$\Rightarrow a = (B^{\mathrm{T}}B)^{-1}B^{\mathrm{T}}g \tag{9}$$

Here, B is a matrix whose rows provide the terms of the polynomial $f(x)$. These are thus multiplied with the coefficients laid out in vector a. The sample values corresponding to coordinates used in the rows of B appear in column vector g. This matrix B is generally not square, with more columns than rows as we use more data points than the order of the polynomial. This overspecifies the solution and hence rejects noise when solving in the least-squares sense.

We create a square system by pre-multiplying by the transpose of B, and this allows us to invert the system to give a in terms of the samples g. The $(B^{\mathrm{T}}B)^{-1}$ term may seem computationally daunting, or at least potentially unstable, but all the elements are integers. This means that the determinant and cofactor matrix are calculated with integers, and the inverse matrix is composed of rationals. We write the required derivative in terms of the coefficients a and the free variable(s).

$$\frac{\partial^c f}{\partial x^c} = \sum_{p=c}^{n} a_p \frac{p!}{(p-c)!} x^{p-c} \tag{10}$$

The terms on the right hand side are constant multiples of the values in a, so these can be written:

$$\frac{\partial^c f}{\partial x^c} = s(B^{\mathrm{T}}B)^{-1}B^{\mathrm{T}}g \tag{11}$$

$$s_p = \begin{cases} 0 & \text{if} \quad 0 \leq p < c \\ \frac{p!}{(p-c)!}x^{p-c} & \text{if} \quad c \leq p \leq n \end{cases} \tag{12}$$

Where s is a vector with elements s_p, noting that $0! = 1$. The dot product of $v = s(B^{\mathrm{T}}B)^{-1}B^{\mathrm{T}}$ with the vector of samples g gives us the c^{th} derivate at the required x value. $s(B^{\mathrm{T}}B)^{-1}B^{\mathrm{T}}$ therefore corresponds to a Savitzky-Golay kernel if we set k odd, and x to the central abscissa of the set of points.

In one dimension, this approach therefore provides for calculation of the same filters as the original Savitzky–Golay formulation, plus off-centre measurements and even sized kernels in commmon with the state of the art. Our method of synthesis is less cumbersome than some approaches, but our practical contribution is more obvious in filters of higher dimensionality.

2.1. Two dimensions

For two dimensions, expressions appear in the literature which provide filters which respond to an uninterrupted regular *rectangular lattice* of data points such as pixels in an image. These give measurements at the centre of a square patch of odd size with a the option for approximated Gaussian weighting on the data points, and at the centre of a square patch of even size with constant weighting using a polynomial of constrained order.

The state of the art in two dimensions is characterized by; a demand that the measurement be at the centre of a square patch, symmetry of weighting, and inclusion of all grid points. The solution most similar to ours is that of Krumm [18], using a polynomial model in two dimensions of the following form:

$$f(x,y) = \sum_{p=0}^{n} \sum_{q=0}^{n-\kappa} a_{p,q} x^p y^q \tag{13}$$

Where n is the maximum sum of the powers of each coordinate variable, and κ is identical to the variable p. This expression does not achieve our notion of the filter allowing the descriptive polynomial's coefficients to vary independently in each variable. Extending our one-dimensional expression to two dimensions by allowing each parameter to vary as a polynomial as we suggest, we have the same expression, but with κ identical to zero

To take derivatives, we use the coefficients of the polynomial model in a similar manner to the one dimensional case. We provide here an expression for the the c^{th} derivative with respect to x (to give the the expression for derivatives with respect to y, simply exchange x for y throughout):

$$\frac{\partial^c f}{\partial x^c} = \sum_{p=c}^{n_x} \sum_{q=0}^{n_y} a_{p,q} \frac{p!}{(p-c)!} x^{p-c} y^q \tag{14}$$

The elements of vector a come from expression 9 using two dimensional extensions of the formulation of matrix B from expression 6 to include all the cross polynomial terms, and the derivative selection vector s of expression 19 similarly augmented with the additional dimension's free variable. To calculate the kernel, we construct the appropriate B matrix and s vector into expression 11. We develop this in detail for the general case in section 2.2 below.

2.2. Higher dimensional filters

To generate a three-dimensional filter we define a polynomial basis function:

$$f(x,y,z) := \sum_{p=0}^{n_x}\sum_{q=0}^{n_y}\sum_{r=0}^{n_z} a_{p,q,r}x^p y^q z^r \tag{15}$$

Where n_x is the order of the polynomial used to model parallel to the x axis, and analogously for n_y and n_z. The matrix B for solving this system has $(n_x+1)(n_y+1)(n_z+1)$ columns, one for each term of the crossed polynomials from each dimension.

The derivative value we require is selected in the same manner as expression 11. We take the c^{th} derivative with respect to free variable ϕ_d. This is the d^{th} dimension, such that, for example, $x=\phi_1$, $y=\phi_2$ and $z=\phi_3$ in the three dimensional case. Generalizing to h dimensions, our measurment is written as follows, with p_* as short-hand for a list of the indices p_1 through p_h (corresponding to p,q,r in expression 15):

$$\frac{\partial^c f}{\partial \phi_d^c} = \sum_{p_1=0}^{n_1}\cdots\sum_{p_d=c}^{n_d}\cdots\sum_{p_h=0}^{n_h} a_{p_*}\frac{p_d!}{(p_d-c)!}\phi_d^{p_d-c}\prod_{v\neq d}\phi_q^{i_v} \tag{16}$$

Matrix B and vector s contain the model terms expressed at the input data points and the derivative of the model terms respectively, and the terms in s and the rows of B must correspond. We therefore define a mapping from column to model term as follows:

$$p_{v,t} = \left\lfloor \frac{t}{\prod_{j=1}^{v-1}(n_j+1)} \right\rfloor \bmod (n_v+1) \tag{17}$$

Where $p_{v,t}$ is the power to which free variable ϕ_v is raised in the t^{th} term of the model, $1\le v\le h$ (*i.e.* h dimensions) and $1\le t\le \prod_{v=1}^{h}(1+n_v)$. $\phi_{v,i}$ is the value of variable ϕ_v at sample i. Thus we write the elements of matrx B as $b_{i,t}$ and of vector s as s_t, and hence the required derivative as follows:

$$b_{i,t} = \prod_{v=1}^{h}\phi_{v,i}^{p_{v,t}} \tag{18}$$

$$s_t = \begin{cases} 0 & \text{if} \quad 0\le p_{d,t}<c \\ \frac{p_{d,t}!}{(p_{d,t}-c)!}\phi_d^{p_{d,t}-c}\prod_{v\neq d}\phi_{v,i}^{p_{v,t}} & \text{if} \quad c\le p_{d,t}\le n_v \end{cases} \tag{19}$$

$$\frac{\partial f}{\partial \phi_d} = s(B^{\mathrm{T}}B)^{-1}B^{\mathrm{T}}g \tag{20}$$

The convolutional patch is thus $s(B^{\mathrm{T}}B)^{-1}B^{\mathrm{T}}$.

3. WEIGHTING AND RECONSTRUCTION

The derivation so far applies unit weight to each data point. If we wish to modulate the influence of the data points according to some constant map – for example, de-emphasizing distant pixels for a smoothing operation using a low-order polynomial, or ignoring pixels which we know to be "dead" – this is formulated as below for expression 4:

$$\epsilon_i = f(x_i) - g_i \tag{21}$$

$$u = \sum_{i=1}^{k} w_i\epsilon_i^2 \tag{22}$$

$$g = Ba \Rightarrow B^{\mathrm{T}}Wg = B^{\mathrm{T}}WBa \tag{23}$$

$$a = (B^{\mathrm{T}}WB)^{-1}B^{\mathrm{T}}Wg \tag{24}$$

W is a diagonal matrix of weights $w_i\ge 0$ for data g_i. These weights may be integers (1 for used points, 0 for discarded points in a reconstruction problem), which yield rational elements of $(B^{\mathrm{T}}WB)^{-1}$, or arbitrary (positive) to allow matching with a general weighting regime. For a solution to exist, the data layout and weighting values must give $\det(B^{\mathrm{T}}WB)\neq 0$.

The weighted patch for derivative measurement corresponding to that given in section 2.2 is thus $s(B^{\mathrm{T}}WB)^{-1}B^{\mathrm{T}}W$.

4. ALIGNMENT OF ANISOTROPY

Using the weighted or unweighted patches of the previous two sections, we rotate the coordinate system of the data into a different alignment in the filter such that isotropic polynomial orders may be steered. We calculate the rows of B using the coordinates of the data points transformed into the rotated axes, along which the polynomials are defined. This generally results in non-integer values in B, but it *does not require resampling*.

An example application of a rotated 2D filter is in an image (2D set of regularly spaced intensity samples) of a fingerprint, such as shown in Greenberg and Kogan's second figure [14]. We see some regions well described by parallel stripes of distinct intensity. Using our approach, we may model the image with a low order polynomial along the stripes, and a high order polynomial perpendicular to the stripes to capture the strong edge gradients.

Consider the same scenario as figure 1 in Greenberg and Kogan. This figure depicts an ellipse of unequal radii rotated by $\theta<\pi/2$ counterclockwise from the data's x-axis, the larger radius therefore falling in the first quadrant. We can align our axes at θ to the x-axis by applying a transformation to the coordinates of the data points we use. As we explain below, it is also possible to use a non-rectangular patch.

5. NON-RECTANGULAR GRIDS AND POINT CLOUDS

It has been amply demonstrated in the literature that any regular data lattice may be mapped onto one with unit spacing, with no loss of generality when performing least squares polynomial filtering, including when the data lies on axes subject to simple transformations from even spacing [4]. The general treatment we provide allows the filtering of a regular lattice of alternative shape by applying a single patch at a range of locations over the data lattice. A simple but powerful example could be composed of triangles rather than rectangles in two dimensions. An example use of this would be in finite element or difference representations of systems. The simplest implementation involves mapping the triangular lattice onto orthogonal axes. Note that this does not require re-sampling the data - just recalculation of the coordinates placed in the B matrix of expression 6.

Perhaps surprisingly, this does not necessarily lead to a requirement for a non-integer B matrix: we may index a point on a two dimensional lattice i along the base direction with vector $(1,0)$ and j along a side of an equilateral triangle at vector $(cos(\pi/3), sin(\pi/3))$. This can be mapped for example to vectors $(2,0)$ and $(1,1)$ *without resampling*. This also preserves the integer nature of matrix B if we do not apply general weighting (see section 3). This approach can be extended to higher dimensions in a similar manner, and to irregular point layouts by selecting locations on a finer lattice.

The triangular lattice is simple to conceptualize and fits with common methods for space subdivision, but we can use our formulation for any shape of array, or entirely unstructured data locations, so long as $\det(B^{\mathrm{T}}WB)$ is non-zero. There is a tacit assumption in much of the literature on polynomial filters that the system should repeat over the space of interest for consecutive points, but this is a restriction on the application, not the filter. Our approach is applicable to general point clouds.

6. DEPLOYMENT

Any desired partial derivative (or re-estimation given by the zeroth derivative) is found by use of a single convolution instead of the smoothing followed by differencing commonly performed in current image processing applications. If many derivatives are desired at a given point, depending on the orders of the polynomials used and the number of dimensions involved, it may be more efficient to generate patches which find the polynomial coefficients (using expression 9), then directly calculate the derivatives from those. Also, when calculating derivatives at a boundary of the data, a band of approximately half the width of the patch will be inaccessible, so it may be desirable to directly interrogate the polynomial (identified by coefficients a in expression 24) for a given patch position at a number of locations within its domain. The following are simple examples of application of the filters:

Spacial derivative estimation Currently, derivative estimation is most commonly performed using smoothing followed by a differencing operator. Using the present approach, this can be achieved using a single filter which preserves detail in a controllable manner over arbitratry dimensions.

Reconstruction in a faulty sensor We can make a best guess at the intensity at a faulty location on a sensor by substituting a value generated by one of our filters of appropriate dimension. We can also calculate spatial derivatives ignoring unreliable data by zero-weighting them in, *e.g.* broken pixels or sensor elements with other spatial distributions such as fibres in an endoscope.

Volumetric time series analysis We model a sequence of volumetric scans of a breathing patient with coordinates x, y, z and t (time). Intensity may be modelled, for example, as quadratic in time to follow smooth dynamics, and cubic in space if we wish to perform edge detection which commonly looks for a zero crossing of a second derivative.

7. CONCLUSIONS

Through reconsidering the traditional approach to formulating polynomial filters, we have provided a straightforward, practical approach to calculating convolutional kernels which estimate derivatives in arbitrary point sets in arbitrary dimensions with steerable anisotropy.

The approach we provide using polynomials produces identical kernels to existing results for their particular target filters, and also generalizes to a wider class of filters in the same number of dimensions, then extends this through a simple, unform process to higher dimensions.

We therefore provide direct estimation of derivatives on a larger class of polynomial models than previously feasible, designed to accurately reproduce detail in an adaptable manner. Another important provision is a simple method for reconstruction of data from damaged sensors. This approach to filter kernel calculation is therefore immediately applicable to problems where none was previously available due to high dimensionality, or complexity of weighting or data placement.

8. REFERENCES

[1] Abraham Savitzky and Marcel J. E. Golay, "Smoothing and differentiation by simplified least squares procedures," *Anal. Chem.*, vol. 36, no. 8, pp. 1672–1639, July 1964.

[2] Jianwen Luo, Kui Ying, and Jing Bai, "Savitzky-golay smoothing and differentiation filter for even number data," *Signal Processing*, vol. 85, pp. 1429–1434, 2005.

[3] Peter A. Gorry, "General least-squares smoothing and differentiation by the convolution (savitzky golay) method," *Anal. Chem.*, vol. 62, pp. 570–573, 1990.

[4] Peter A. Gorry, "General least-squares smoothing and differentiation of nonuniformly spaced data by the convolution method," *Anal. Chem.*, , no. 63, pp. 534–536, 1991.

[5] Jianwen Luo, Kui Ying, and Jing Bai, "Properties of savitzky-golay digital differentiators," *Digital Signal Processing*, , no. 15, pp. 122–136, 2005.

[6] Kenneth L. Ratzlaff and Jean T. Johnson, "Comuputation of two-dimensional polynomial least squares convolutional smoothing integers," *Anal. Chem.*, pp. 1303–1305, 1989.

[7] John E. Kuo, Hai Wang, and Stephen Pickup, "Multidimensional least squares smoothing using orthogonal polynomials," *Anal. Chem.*, vol. 63, pp. 630–635, 1991.

[8] P. Nikitas and A. Pappa-Louisi, "Comments on the two-dimensional smoothing of data," *Analytica Chimica Acta*, vol. 415, pp. 117–125, 2000.

[9] Peter Meer and Isaac Weiss, "Smoothed differentiation filters for images," *Journal of Visual Communications and Image Representation*, vol. 3, no. 1, pp. 58–72, March 1992.

[10] Philip Barak, "Smoothing and differentiation by an adaptive-degree polynomial filter," *Anal. Chem.*, pp. 2758–2762, 1995.

[11] Malgorzata Jakubowska and Wladyslaw W. Kubiak, "Adaptive-degree polynomial filter for voltammetric signals," *Analytica Chimica Acta*, vol. 512, pp. 241–250, 2004.

[12] William T. Freeman and Edward H. Adelson, "The design and use of steerable filters," *IEEE transactions on pattern analysis and machine intelligence*, vol. 13, no. 9, September 1991.

[13] G. Z. Yang, P. Burger, D. N. Furmin, and S. R. Underwood, "Structure adaptive anisotropic image filtering," *Image and Vision Computing*, , no. 14, pp. 135–145, 1996.

[14] Shlomo Greenberg and Daniel Kogan, "Improved structure-adaptive anisotropic filter," *Pattern Recognition Letters*, vol. 27, pp. 59–65, 2006.

[15] S. M. Kerawala, "A rapid method for calculating the least squares solution of a polynomial of degree not exceeding the fifth," *Indian Journal of Physics*, vol. 15, pp. 241, 1941.

[16] E. H. Moore, "On the reciprocal of the general algebraic matrix," *Bulletin of the American Mathematical Society*, vol. 26, pp. 394–395, 1920.

[17] Roger Penrose, "On best approximate solution of linear matrix equations," *Proceedings of the Cambridge Philosophical Society*, vol. 52, pp. 17–19, 1956.

[18] John Krumm, *http://research.microsoft.com/users/jckrumm/SavGol/SavGol.htm - dated Aug 2001, as seen Apr 2007.*

[19] G. Golub and C. van Loan, *Matrix computations, Third Ed.*, The Johns Hopkins University Press, London, 1996.