Optimal Surface Smoothing as Filter Design

Gabriel Taubin[1], Tong Zhang and Gene Golub[2]

[1] IBM T.J.Watson Research Center, P.O.Box 704, Yorktown Heights, NY 10598
[2] Computer Science Department, Stanford University, Stanford, CA 94305.

Abstract. Smooth surfaces are approximated by polyhedral surfaces for a number of computational purposes. An inherent problem of these approximation algorithms is that the resulting polyhedral surfaces appear faceted. Within a recently introduced signal processing approach to solving this problem [7, 8], surface smoothing corresponds to low-pass filtering. In this paper we look at the filter design problem in more detail. We analyze the stability properties of the low-pass filter described in [7, 8], and show how to minimize its running time. We show that most classical techniques used to design finite impulse response (FIR) digital filters can also be used to design significantly faster surface smoothing filters. Finally, we describe an algorithm to estimate the power spectrum of a signal, and use it to evaluate the performance of the different filter design techniques described in the paper.

1 Introduction

The signal processing framework introduced in [7, 8], extends Fourier analysis to *discrete surface signals*, functions defined on the vertices of polyhedral surfaces. As in the method of Fourier Descriptors [9], where a closed curve is smoothed by truncating the Fourier series of its coordinate signals, a very large polyhedral surface of arbitrary topology is smoothed here by low-pass filtering its three surface coordinate signals. And although the formulation was developed mainly for signals defined on surfaces, it is in fact valid for *discrete graph signals*, functions defined on the vertices of directed graphs. Since this general formulation provides a unified treatment of polygonal curves, polyhedral surfaces, and even three-dimensional finite elements meshes, we start this paper by reviewing this formulation in its full generality.

2 Fourier Analysis of Discrete Graph Signals

We represent a *directed graph* on the set $\{1,\ldots,n\}$ of n nodes as a set of *neighborhoods* $\{i^\star : i=1,\ldots,n\}$, where $i^\star$ is a subset of nodes which does not contain i. The element of $i^\star$ are the *neighbors* of i. A *discrete graph signal* is a vector $x=(x_1,\ldots,x_n)^t$ with one component per node of the graph. A *discrete surface signal* is a discrete graph signal defined on the graph of vertices and edges of a polyhedral surface. We normally use *first order neighborhoods*, were node j is a neighbor of node i if i and j share an edge (or face), but other neighborhood structures can be used to impose certain types of constraints [8].

The Discrete Fourier Transform (DFT) of a signal x defined on a closed polygon of n vertices is obtained by decomposing the signal as a linear combination of the eigenvectors of the Laplacian operator

$$\Delta x_i = \frac{1}{2}(x_{i-1} - x_i) + \frac{1}{2}(x_{i+1} - x_i) \ . \tag{1}$$

The DFT of x is the vector $\hat{x}$ of coefficients of the sum. The Laplacian operator must be replaced by another linear operator to define the DFT of a discrete graph signal. This is the same idea behind the method of eigenfunctions of Mathematical Physics [1].

We define the *Laplacian* of a discrete graph signal x by the formula

$$\Delta x_i = \sum_{j \in i^\star} w_{ij} \left(x_j - x_i \right) , \tag{2}$$

where the weights w_{ij} are positive numbers that add up to one for each vertex. These weights can be chosen in many different ways taking into consideration the neighborhoods, but in this paper we will assume that they are not functions of the signal x. Otherwise, the resulting operator is non-linear, and so, beyond the scope of this paper. One particularly simple choice that produces good results is to set w_{ij} equal to the inverse of the number of neighbors $1/|i^\star|$ of node i, for each element j of $i^\star$. Other choices of weights are discussed in [7, 8]. Note that the Laplacian of a signal defined on a closed polygon, described in equation (1), is a particular case of these definitions, with $w_{ij} = 1/2$, for $j \in i^\star = \{i-1, i+1\}$, for each node i.

If $W = (w_{ij})$ denotes the matrix of weights, with $w_{ij} = 0$ when j is not a neighbor of i, and $K = I - W$, the Laplacian of a discrete signal can be written in matrix form as

$$\Delta x = -Kx \ . \tag{3}$$

Although the method applies to general neighborhood structures, in this paper we will restrict our analysis to those cases where the matrix W can be factorized as a product of a symmetric matrix times a positive definite diagonal matrix $W = ED$. In this case the matrix W is a *normal matrix* [3], because the matrix

$$D^{1/2} W D^{-1/2} = D^{1/2} E D^{1/2} \tag{4}$$

is symmetric. Note that such is the case for the first order neighborhoods of a surface with equal weights $w_{ij} = 1/|i^\star|$ in each neighborhood $i^\star$, where E is the *incidence matrix* of the neighborhood structure (a symmetric matrix for first order neighborhoods), the matrix whose ij-th. element is equal to 1 if the nodes i and j are neighbors, and 0 otherwise; and D is the diagonal positive definite matrix whose i-th. diagonal element is $1/|i^\star|$. When W is a normal matrix it has all real eigenvalues, and sets of n left and right eigenvectors that form dual bases of n-dimensional space. Furthermore, by construction, W is also a *stochastic matrix*, a matrix with nonnegative elements and rows that add up to one [6]. The eigenvalues of a stochastic matrix are bounded above in magnitude by 1. It follows that the eigenvalues of the matrix K are real, bounded below by 0, and above by 2.

In general, the eigenvectors and eigenvalues of K have no analytic expression, but for filtering operations it is not necessary to compute the eigenvectors explicitly.

If $0 \leq k_1 \leq \cdots \leq k_n \leq 2$ are the eigenvalues of K, $e_1, \ldots, e_n$ a set of corresponding right eigenvectors, and $\delta_1, \ldots, \delta_n$ the associated dual basis of $e_1, \ldots, e_n$, the identity matrix I, and the matrix K can be written as follows

$$I = \sum_{i=1}^{n} e_i \delta_i^t \qquad K = \sum_{i=1}^{n} k_i \, e_i \delta_i^t \; ,$$

and every discrete graph signal x has a unique decomposition as a linear combination of $e_1, \ldots, e_n$

$$x = I\, x = \sum_{i=1}^{n} \hat{x}_i \, e_i \; , \tag{5}$$

where $\hat{x}_i = \delta_i^t x$. We call the vector $\hat{x} = (\hat{x}_1, \ldots, \hat{x}_n)^t$ the Discrete Fourier Transform (DFT) of x.

Note, however, that this definition does not identify a unique object yet. If a different set of right eigenvectors of K is chosen, a different DFT is obtained. To complete the definition, if $W = ED$, with E symmetric and D positive definite diagonal, we impose the right eigenvectors of K to be of unit length with respect to the norm associated with the inner product $\langle x, y \rangle_D = x^t D y$. With this constraint, Parseval's formula is satisfied

$$\|x\|_D^2 = \|\hat{x}\|^2 \; , \tag{6}$$

where the norm on the right hand side is the Euclidean norm. This result will be used in sections 6 and 7.

To filter the signal x is to change its frequency distribution according to a transfer function $f(k)$

$$x' = \sum_{i=1}^{n} f(k_i) \, \hat{x}_i e_i = \left(\sum_{i=1}^{n} f(k_i) \, e_i \delta_i^t \right) x \; . \tag{7}$$

The frequency component of x corresponding the the natural frequency k_i is enhanced or attenuated by a factor $f(k_i)$. For example, the transfer function of an ideal low-pass filter is

$$f_{\text{LP}} = \begin{cases} 1 & \text{for } 0 \leq k \leq k_{\text{PB}} \\ 0 & \text{for } k_{\text{PB}} < k \leq 2 \end{cases} \; , \tag{8}$$

where k_{PB} is the *pass-band frequency.*

Since there is no efficient numerical method to compute the DFT of a discrete graph signal, the computation can only be performed approximately. To do this the ideal low-pass filter transfer function is replaced by an analytic approximation, usually a polynomial or rational function, for which the computation can be performed in an efficient manner. A wide range of analytic functions of one variable $f(k)$ can be evaluated in a matrix such as K [3]. The result is another matrix $f(K)$ with the same left and right eigenvectors, but with eigenvalues $f(k_1),\ldots,f(k_n)$

$$f(K)=\sum_{i=1}^{n} f(k_i)\, e_i\delta_i^t \, .$$

The main reason why the filtering operation $x'=f(K)x$ of equation (7) can be performed efficiently for a polynomial transfer function of low degree, is that when K is sparse, which is the case here, the matrix $f(K)$ is also sparse (but of wider bandwidth), and so, the filtering operation becomes the multiplication of a vector by a sparse matrix.

In Gaussian smoothing the transfer function is the polynomial $f_N(k) = (1-\lambda k)^N$, with $0<\lambda<1$. This transfer function produces shrinkage. The algorithm introduced in [7, 8] is essentially Gaussian smoothing with the difference that the scale factor λ changes from iteration to iteration, alternating between a positive value λ and a negative value μ. This simple modification still produces smoothing, but prevents shrinkage. The transfer function is the polynomial $f_N(k)=((1-\lambda k)(1-\mu k))^{N/2}$, with $0<\lambda<-\mu$ and N even. The *pass-band frequency* of this filter is defined as the unique value of k in the interval $(0,2)$ such that $f_N(k)=1$. Such a value exists when $0<\lambda<-\mu$, and turns out to be equal to $k_{\text{PB}}=1/\lambda+1/\mu$. This polynomial transfer function of degree N results in a linear time and space complexity algorithm. From now on we will refer to this algorithm as the $\lambda-\mu$ algorithm.

3 Fast Smoothing as Filter Design

We are faced with the classical problem of digital filter design in signal processing [5, 4], but with some restrictions. Note that because of the linear complexity constraint discussed above, only polynomial transfer functions (FIR filters) are allowed. We leave the study of rational transfer functions (IIR filters) for the future. And because of space restrictions, of all the traditional FIR filter design methods available in the signal processing literature, we only cover here in some detail the method of windows, which is the simplest one. With this method we can design filters which are significantly faster, or sharper, than those obtain with the $\lambda-\mu$ algorithm for the same degree.

4 Optimizing the $\lambda-\mu$ algorithm

The $\lambda-\mu$ algorithm can be described in a recursive fashion as follows

$$f_N(k)=\begin{cases} 1 & N=0 \\ (1-\lambda_N\, k)\, f_{N-1}(k) & N>0 \end{cases}$$

where $\lambda_N = \lambda$, for N odd, and $\lambda_N = \mu$ for N even. Note that this algorithm requires minimum storage, only one array of dimension n to store the Laplacian of a signal if computed in place, and two arrays of dimension n in general.

To maintain the minimum storage property and the same simple algorithmic structure, one could try to generalize by changing the scale factors λ_N from iteration to iteration in a different way. But if we start with a given pass-band frequency $k_{\text{PB}} = 1/\lambda + 1/\mu$, as it is usually the case when one wants to *design* the filter, there are many values of λ and μ such that $0 < \lambda < -\mu$, that define a filter with the same pass-band frequency. In order for the polynomial $f(k) = (1 - \lambda k)(1 - \mu k)$ to define a low-pass filter in the interval $[0,2]$ it is necessary that $|f(k)| < 1$ in the stop-band region, so that $f_N(k) = f(k)^N \rightarrow 0$ when N grows. Since $f(k_{\text{PB}}) = 1$ and $f(k)$ is strictly decreasing for $k > k_{\text{PB}}$, this condition is equivalent to $f(2) > -1$, which translates into the following constraint on λ

$$\lambda < \frac{-k_{\text{PB}} + \sqrt{(2-k_{\text{PB}})^2 + 4}}{2(2-k_{\text{PB}})} . \tag{9}$$

As λ increases, the slope of the filter immediately after the pass-band frequency increases, i.e., the filter becomes sharper, but at the same time instability starts to develop at the other end of the spectrum, close to $k = 2$. If the maximum eigenvalue k_n of the matrix K is significantly less than 2 (which is not usually the case) we only need the filter to be stable in the interval $[0, k_n]$ (i.e., $1 > f(k_n) > -1$), and larger values of λ are acceptable. A good estimate of the maximum eigenvalue of K can be obtained with the Lanczos method [3]. Even if the maximum eigenvalue k_n is not known, the signal x to be smoothed may be band-limited, i.e., the coefficients $\hat{x}_i$ in equation (5) associated with high frequencies are all zero, or very close to zero. This condition may be difficult to determine in practice for a particular signal, but if we apply the algorithm with small λ for a certain number of iterations, the resulting signal becomes in effect band-limited. At this point λ can be increased keeping the pass-band frequency constant, maybe even making the filter unstable, and the algorithm can be applied again with the new values of λ and μ for more iterations. This process of increasing λ keeping the pass-band frequency constant can now be repeated again and again. A moderate speed-up is obtained in this way.

5 Filter Design with Windows

The most straightforward approach to traditional digital filter design is to obtain a trigonometric polynomial approximation of the ideal filter transfer function by truncating its Fourier series. The resulting trigonometric polynomial minimizes the L_2 distance to the ideal filter transfer function among all the trigonometric polynomials of the same degree.

To obtain regular polynomials, not trigonometric ones, we first apply the change of variable $k = 2(1 - \cos(\theta))$. This change of variable is a $1 - 1$ mapping $[0, \pi/2] \rightarrow [0, 2]$. Then we extend the resulting function to the interval $[-\pi, \pi]$ as follows

$$h_{\text{LP}}(\theta) = \begin{cases} 0 & \pi/2 \le \theta \le \pi \\ f_{\text{LP}}(2(1-\cos(\theta))) & 0 \le \theta \le \pi/2 \\ h(-\theta) & -\pi \le \theta \le 0\,. \end{cases}$$

Note that this function, periodic of period 2π and even, is also an ideal low-pass filter as a function of θ

$$h_{\text{LP}}(\theta) = \begin{cases} 1 & \text{if } |\theta| < \theta_{\text{PB}} \\ 0 & \text{otherwise} \end{cases},$$

where θ_{PB} is the unique solution of $k_{\text{PB}} = 2(1-\cos(\theta_{\text{PB}}))$ in $[0,\pi/2]$. Since $h(\theta)$ is an even function, it has a Fourier series expansion in terms of cosines only

$$h_{\text{LP}}(\theta) = h_0 + 2\sum_{n=0}^{\infty} h_n \cos(n\theta)\ .$$

Now, it is well known that $\cos(n\,\theta) = T_n(\cos(\theta))$, where T_n is the n-th. Chebyshev polynomial [2], defined by the three term recursion

$$T_n(w) = \begin{cases} 1 & n=0 \\ w & n=1 \\ 2\,w\,T_{n-1}(w) - T_{n-2}(w) & n>1 \end{cases}$$

The N-th. polynomial approximation of f_{LP} for $k \in [0,2]$ is then

$$f_N(k) = \frac{\theta_{\text{PB}}}{\pi}\,T_0(1-k/2) + \sum_{n=1}^{N} \frac{2\sin(n\,\theta_{\text{PB}})}{n\,\pi}\,T_n(1-k/2)\,. \tag{10}$$

Direct truncation of the series leads to the well-known Gibbs phenomenon, i.e., a fixed percentage overshoot and ripple before and after the discontinuity. As it is shown in section 8, this is one of the problems that makes this technique unsatisfactory. The other problem is that the resulting polynomial approximation does not necessarily satisfy the constraint $f_N(0) = 1$, which is required to preserve the average value of the signal (DC level in classical signal processing, centroid in the case of surfaces). Our experiments show that a desirable surface smoothing filter transfer function should be as close as possible to 1 within the pass-band, and then decrease to zero in the stop-band ($[k_{\text{PB}}, 2]$).

A classical technique to control the convergence of the Fourier series is to use a weighting function to modify the Fourier coefficients. In our case the polynomial approximation of equation (10) is modified as follows

$$f_N(k) = w_0\,\frac{\theta_{\text{PB}}}{\pi}\,T_0(1-k/2) + w_n \sum_{n=1}^{N} \frac{2\sin(n\,\theta_{\text{PB}})}{n\,\pi}\,T_n(1-k/2)\,, \tag{11}$$

where $w_0, w_1, \ldots, w_N$ are the weights that constitute a so called *window*. The polynomial approximation of equation (10) is a particular case of (11), where the weights are all equal to 1. This is called the *Rectangular window*. Other popular windows are, the *Hanning window*, the *Hamming window*, and the *Blackman window*.

$$w_n = \begin{cases} 1.0 & \text{Rectangular} \\ 0.5 + 0.5\,\cos(n\pi/(N+1)) & \text{Hanning} \\ 0.54 + 0.46\,\cos(n\pi/(N+1)) & \text{Hamming} \\ 0.42 + 0.5\,\cos(n\pi/(N+1)) + 0.08\,\cos(2n\pi/(N+1)) & \text{Blackman}\,. \end{cases} \tag{12}$$

If the low-pass filter must have a very narrow pass-band region, which is usually the case in the surface smoothing application, then a high degree polynomial is necessary to obtain a reasonable approximation. This is in fact a consequence of the uncertainty principle. The phenomenon can be observed even in the case of the rectangular window. The problem is even worse for the other windows, because they have wider main lobes. To obtain a reasonably good approximation of degree N, the pass-band must be significantly wider than the width of the main lobe of the window. If σ is the width of the main lobe of the window, the resulting filter will be approximately equal to one for $\theta \in [0, \theta_{\text{PB}} - \sigma]$, approximately equal to zero for $\theta \in [\theta_{\text{PB}} + \sigma, \pi]$, and approximately decreasing for $\theta \in [\theta_{\text{PB}} - \sigma, \theta_{\text{PB}} + \sigma]$. Our solution in this case of narrow pass-band frequency, is to design the filter for a small value of N, but with the pass-band frequency increased by σ (no longer the width of the main lobe of the window)

$$f_N(k) = w_0\,\frac{(\theta_{\text{PB}} + \sigma)}{\pi}\,T_0(1 - k/2) + w_n \sum_{n=1}^{N} \frac{2\sin(n(\theta_{\text{PB}} + \sigma))}{n\,\pi}\,T_n(1 - k/2)\;, \tag{13}$$

and then, eventually iterate this filter ($f(k) = f_N(k)^M$). The value of σ can be determined numerically by maximizing $f(k_{\text{PB}})$ under the constraints $|f(k)| < 1$ for $k_{\text{PB}} < k \leq 2$. In our implementation, we compute the optimal σ with a local root finding algorithm (a few Newton iterations) so that $f_N(k_{\text{PB}}) = 1$, starting from an interactively chosen initial value. Figure 1 shows some examples of filters designed in this way, compared with filters of the same degree and $\sigma = 0$, and with $\lambda - \mu$ filters of the same degree.

6 How to Choose The Pass-Band Frequency

In this section we are concerned with how to choose the pass-band frequency k_{PB} to prevent shrinkage. As in the classical case, since the DFT $\hat{x}$ of a signal x satisfies Parseval's formula, the value of $\hat{x}_i^2$ can be interpreted as the *energy content* of x in the frequency k_i. Similarly, the sum

$$\sum_{k_i \leq k_{\text{PB}}} \hat{x}_i^2$$

measures the energy content of x in the pass-band. Our criterion is to choose the minimum pass-band frequency such that most of the energy of the signal falls

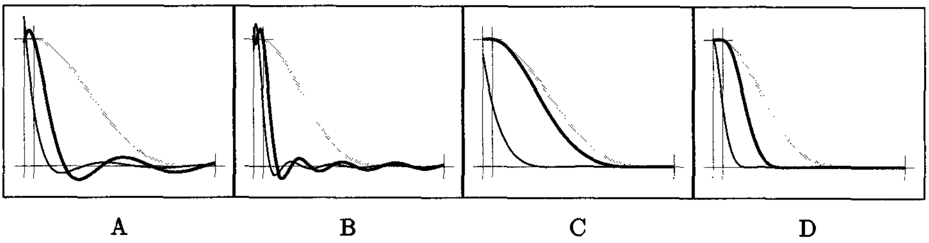

Fig. 1. Filters $f_N(k)$ for $k_{\text{PB}} = 0.1$ and $\sigma > 0.0$. (A) Rectangular window, $N = 10$, $\sigma = 0.1353$. (B) Rectangular window, $N = 20$, $\sigma = 0.0637$. (C) Hamming window, $N = 10$, $\sigma = 0.5313$. (D) Hamming window, $N = 20$, $\sigma = 0.2327$. In each of the four cases the thick black line corresponds to the filter described above, the thin black line to the same filter with $\sigma = 0.0$, and the gray line is a $\lambda - \mu$ filter of the same degree and $\lambda = 0.5$.

in the pass-band, i.e., we choose k_{PB} such that

$$\sum_{k_i \le k_{\text{PB}}} \hat{x}_i^2 \ge (1 - \epsilon) \|x\|_D^2 ,$$

where ϵ is a very small number. Of course, since we cannot compute the DFT of x, we cannot minimize this expression exactly. We can only get a rough estimate of the minimizer using the power spectral estimator described in the next section. What value of ϵ to use, and how accurate the estimation should be is application dependent, but in general it should be determined experimentally for a set of typical signals.

7 Power Spectrum Estimation

Ideally, to evaluate the performance of the different low-pass filter algorithms we should measure the DFT of the filter outputs, and check that the high frequency energy content is very small. Since we do not have any practical way of computing the DFT, we estimate the power spectrum of a signal as follows. We partition the interval $[0, 2]$ into a small number of non-overlapping intervals $I^1, \ldots, I^M$, and for each one of this intervals we estimate the energy content of the signal within the interval. We do so by designing a very sharp (high degree) pass-band filter $f^j(k)$ for each interval I^j. The energy content of the signal x within the interval I^j can be estimated by measuring the total energy of the output of corresponding filter applied to the signal

$$\|f^j(K)x\|_D^2 \approx \sum_{k_i \in I^j} \hat{x}_i^2 .$$

By designing all these FIR filters of the same degree, a filter-bank, we can evaluate all of them simultaneously at a greatly reduced computational cost. The only disadvantage is that we need M arrays of the same dimension as the input signal x to accumulate the filter outputs before their norms are evaluated. If the pass-band filters were ideal, Parseval's formula implies that the sum of the total energies of the filter outputs must be equal to the total energy of the input signal. Since the transfer functions of the filters overlap, this condition is only approximately satisfied. But the error can be made arbitrarily small by increasing the degree of the polynomials. We recommend using filters designed with the Hanning or Hamming windows of a degree at lest ten times the number of spectrum bands.

8 Experimental Results

Figure 2 shows the result of applying the filters of figure 1 to the same input surface. The spectrum estimate for the input surface yields the 99.88% of the energy in the band $[0, 0.1]$. This is a typical result for relatively large surfaces, and we have found that a default value $k_{PB} = 0.1$ produces very good results. But as we pointed out before, the appropriate value for a family of similar signals must be determined experimentally by estimating the spectrum of a typical sample.

The ideal transfer function should be as flat as possible in the pass-band region ($f(k) \approx 1$ for $k \in [0, k_{PB}]$), and then decrease as fast as possible in the stop-band region ($k \in [k_{PB}, 2]$). The transfer function of the $\lambda - \mu$ algorithm has this shape, but does not decrease fast enough in the stop-band. The results obtained with rectangular filters are unsatisfactory. The filters designed with the other three windows (Hanning, Hamming, and Blackman), and with increased σ produce transfer functions of similar shape. The Blackman window produces transfer functions that are much flatter in the pass-band, but at the expense of a slower rate of decrease in the stop-band. Hanning and Hamming windows produce similar results, but the Hamming window produces transfer functions with less oscillations. As figure 2 shows, filters designed with the Hamming window produce filters of similar quality as the $\lambda - \mu$ algorithm, but much faster.

References

1. R. Courant and D. Hilbert. *Methods of Mathematical Physics*, volume 1. Interscience, 1953.
2. P.J. Davis. *Interpolation and Approximation*. Dover Publications, Inc., 1975.
3. G. Golub and C.F. Van Loan. *Matrix Computations*. John Hopkins University Press, 2nd. edition, 1989.
4. R.W. Hamming. *Digital Filters*. Prentice Hall, 1989.
5. A.V. Oppenheim and R.W. Schafer. *Digital Signal Processing*. Prentice Hall, Englewood Cliffs, NJ, 1975.
6. E. Seneta. *Non-Negative Matrices, An Introduction to Theory and Applications*. John Wiley & Sons, New York, 1973.
7. G. Taubin. Curve and surface smoothing without shrinkage. In *Proceedings, Fifth International Conference on Computer Vision*, pages 852–857, June 1995.
8. G. Taubin. A signal processing approach to fair surface design. *Computer Graphics*, pages 351–358, August 1995. (Proceedings SIGGRAPH'95).

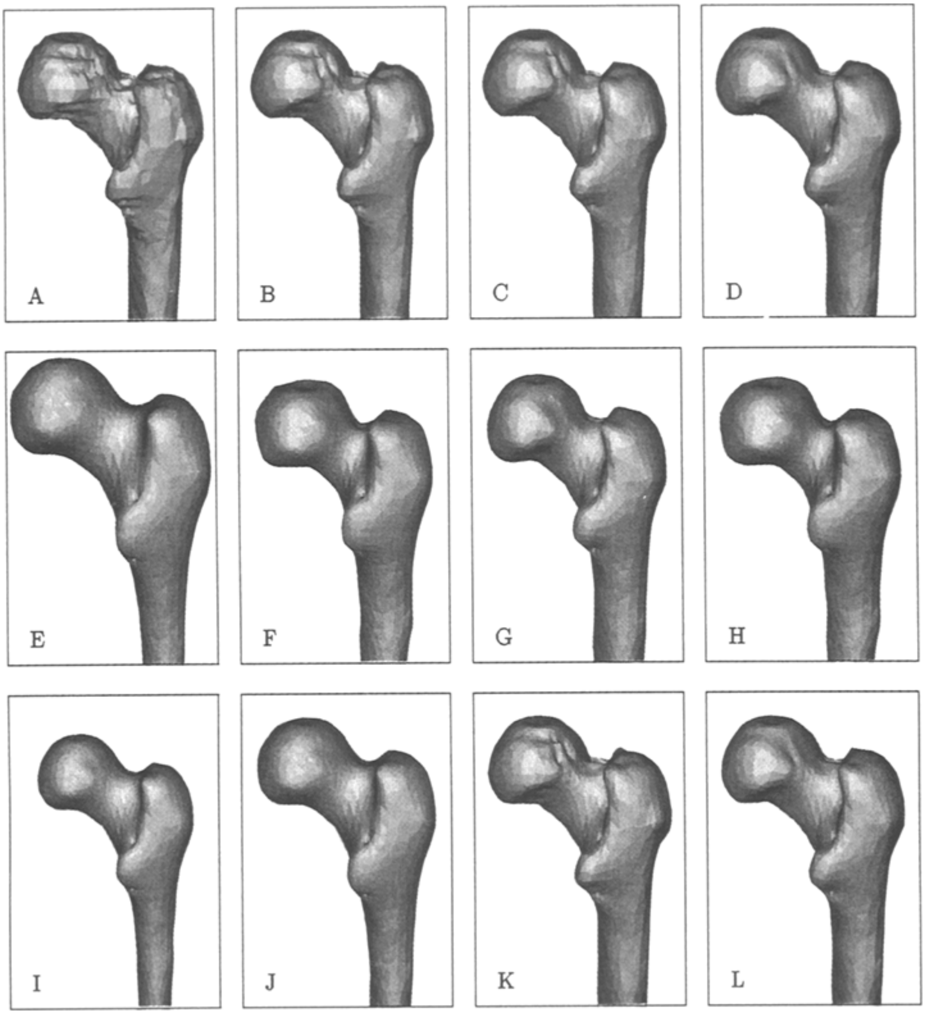

Fig. 2. Filters of Figure 1 applied to the same surface. In all these examples $k_{PB} = 0.1$. (A) Input surface (2565 vertices, 5138 triangles). (B) $\lambda - \mu$ filter $\lambda = 0.5$ $n = 10$. (C) $\lambda - \mu$ filter $\lambda = 0.5$ $n = 20$. (D) $\lambda - \mu$ filter $\lambda = 0.5$ $n = 60$. (E) Rectangular window $\sigma = 0.0$ $n = 10$. (F) Rectangular window $\sigma = 0.0$ $n = 20$. (G) Rectangular window $\sigma = 0.01353$ $n = 10$. (H) Rectangular window $\sigma = 0.06374$ $n = 20$. (I) Hamming window $\sigma = 0.0$ $n = 10$. (J) Hamming window $\sigma = 0.0$ $n = 20$. (K) Hamming window $\sigma = 0.5313$ $n = 10$. (L) Hamming window $\sigma = 0.2327$ $n = 20$.

9. C.T. Zahn and R.Z. Roskies. Fourier descriptors for plane closed curves. *IEEE Transactions on Computers*, 21(3):269–281, March 1972.