Efficient Synthesis of Gaussian Filters by Cascaded Uniform Filters

WILLIAM M. WELLS, III, MEMBER, IEEE

Abstract—Gaussian filtering is an important tool in image processing and computer vision. In this paper we discuss the background of Gaussian filtering and look at some methods for implementing it. Consideration of the central limit theorem suggests using a cascade of "simple" filters as a means of computing Gaussian filters. Among "simple" filters, uniform-coefficient finite-impulse-response digital filters are especially economical to implement. The idea of cascaded uniform filters has been around for a while [13], [16]. We show that this method is economical to implement, has good filtering characteristics, and is appropriate for hardware implementation. We point out an equivalence to one of Burt's methods [1], [3] under certain circumstances. As an extension, we describe an approach to implementing a Gaussian Pyramid which requires approximately two addition operations per pixel, per level, per dimension. We examine tradeoffs in choosing an algorithm for Gaussian filtering, and finally discuss an implementation.

Index Terms—Binomial convolution, Gaussian convolution, Gaussian Pyramid, image processing, Laplacian Pyramid, uniform filter convolutions.

I. Introduction

In this section, we look at the motivations for using Gaussian filtering in image processing and computer vision.

A filter that has a Gaussian-like impulse response function is essentially a low-pass filter. Gaussian filters are useful in the synthesis of high-pass and bandpass filters.

In their work on edge detection, Marr and Hildreth [11] argue that meaningful intensity changes in a two-dimensional image occur at a variety of spatial scales. These changes, they say, are best located by maxima in the first spatial derivative, or by zeros in the second spatial derivative of the low-pass filtered image. Desire for a radially symmetric, second-derivative operator led Marr and Hildreth to choose the Laplacian operator, applied to the two-dimensional Gaussian, as the impulse response of their filter. The size, or standard deviation, of the Gaussian may be adjusted according to the spatial scale desired. Significant intensity changes in the image of the chosen scale are marked where the filtered result crosses zero.

Effective stop band rejection and spatial (temporal) localization are characteristics of good low-pass filters. The Gaussian is interesting as it is the function with the minimum product of second moments in space (time) and frequency. The Gaussian is also separable (discussed in next section). This separability is important in the realization of multidimensional filters.

The Laplacian applied to the Gaussian is a "Mexican hat" or "center-surround" operator. This operator is often efficiently approximated as the difference of Gaussians of two scales (DOG).

Marr and Hildreth propose that a suitable representation for visual imagery is a set of zero-crossing maps that are derived from the output of a multiple scale set of Laplacian of Gaussian filters [11]. Such a set of filter outputs, or Laplacian Pyramid, may be approximated, as mentioned above, by subtracting neighboring levels of a Gaussian Pyramid (the output of a set of multiscale Gaussian filters). Further, they suggest the construction of a symbolic description of the edges, bars, blobs, and terminations evident in the set of zero crossings, which they call the "raw primal sketch."

Witkin [17] further develops the concept of multiple-scale low-pass filtering. He proposes a representation for signals in which continuously varying the scale parameter causes the filtered signal to sweep out a surface in what he calls the "scale-space-image." In Witkin's words:

> "In this representation, it is possible to track extrema as they move continuously with scale changes, and to identify the singular points at which new extrema appear. The scale-space image is then collapsed into a tree, providing a concise but complete qualitative description of the signal over all scales of observation." [17]

The notion of describing signals by tracking their derivative extrema in multiple-scale filterings has become popular in computer vision research. Crowley and Parker [6] apply a difference of low-pass (DOLP) transform (a type of Laplacian Pyramid) to two-dimensional image data. The steps in the scale parameter are the square root of two. They link peaks and ridges in adjacent scales to form a graph which describes shapes in the image data. Correspondence matching between pairs of such descriptions is facilitated by a coarse–fine strategy. Crowley and Sanderson [7] explore a pattern description, derived from the DOLP transform, which is a tree of symbols with probabilistic attributes. They describe a correspondence algorithm which maximizes a similarity measure, and an algorithm which derives pattern models from observation sets.

Manuscript received March 8, 1985; revised August 19, 1985. Recommended for acceptance by W.E.L. Grimson. This work was supported by the Air Force Office of Scientific Research under Contract F49620-82-C-0092 at the Stanford Artificial Intelligence Laboratory.

The author is with SRI International, Menlo Park, CA 94025.

IEEE Log Number 8406216.

The derivation of symbolic graph representations from scale-space has been applied in other areas of computer vision as well. Brady and Asada [4] apply a Gaussian Pyramid to image contour curves (in the form of tangent angle as a function of arc length). In the resulting scale-space they identify instances of parameterized models of curve features such as corners. They call the resulting representation the "curvature primal sketch." The identified curve features are used in the computation of a higher level representation called "smoothed local symmetries." Ponce and Brady [15] apply similar techniques to dense depth maps in order to locate important surface intersections.

Others who have used Gaussian filtering include Canny, who in his work on edge detection [5], computes optimal edge-operators that can be approximated by derivatives of Gaussians, and Burt and Adelson [2], who employ a Laplacian Pyramid in their image data compression scheme.

As the above researchers have discovered, Gaussian filtering can be an important tool in image processing and computer vision.

II. Computing Gaussian Convolutions

In this section, we examine ways to implement Gaussian filters and look at the attractive features of cascaded uniform filters.

A normalized, radially symmetric, central two-dimensional Gaussian function is defined by

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}.$$

Note that this function may be separated into a function of x multiplied (or convolved) by a function of y. For an operator limited to an N by N nonzero region of support, this reduces the cost of the convolution from N^2 to $2N$ multiplies and adds.

The Gaussian function is also well known in probability and statistics as the normal density function. It is the subject of the central limit theorem, which we will describe.

If a random variable is defined to be the sum of a set of independent random variables, then the probability density of that random variable is obtained by the convolution of the probability densities of the random variables in the set. The central limit theorem states that under quite general conditions the resulting probabilty density tends to the normal density function as the number of members of the set grows [9].

An example of this used in texts on probability and statistics is when several random variables having uniform (rectangular) densities are combined. When three such densities are convolved, the visual resemblance of the result to an approximating Gaussian is striking [10].

The central limit theorem illustrates an interesting property of the repeated convolution of simple nonnegative functions—the result tends to a Gaussian function. This property provides an economical means for computing Gaussian convolutions.

Canny [5] exploits this property to improve his recursive filter approximation of Gaussian convolutions. The cascaded uniform filters method we discuss below is also based on this approach. A simple example of this would be to successively convolve by a one-dimensional function consisting to two adjacent ones, zeros elsewhere. This is equivalent to convolving once by the binomial coefficients. These have long been approximated with Gaussian functions by statisticians [9].

This binomial algorithm [14] might seem appealing since there is no multiplication in the computation. But compared to the methods below, it may not be the best approach, since the standard deviation of the resulting Gaussian approximation grows only as the square root of the number of successive convolutions of the simple kernel.

Burt and Crowley

Burt and Crowley present similar approaches to computing a Gaussian Pyramid. Both employ a "generating kernel" that has a small group of nonzero elements, (typically 5 by 5 for 2-D applications). Burt [1], [3] describes two methods, "Hierarchical Discrete Correlation," and "Reduced Hierarchical Discrete Correlation." The second is applied to efficient signal encoding [2].

Burt's two methods may be described recursively as follows:

1)
- The first member of the pyramid is the input image.
- The n + 1st member of the pyramid is computed by convolving the nth member of the pyramid by the generating kernel after the generating kernel has been "expanded" by a factor of s.

2)
- The first member of the pyramid is the input image.
- The n + 1st member of the pyramid is computed by "reducing" the nth member of the pyramid by a factor of s and then convolving it by the generating kernel.

In the case where $s = 2$, the expansion consists of inserting a zero between each element of the kernel, such as transforming the kernel (1, 2, 1) into (1, 0, 2, 0, 1). When $s = 2$, the reduction amounts to throwing out every other element of the distribution, meaning (1, 2, 3, 4) could be transformed into (1, 3).

Burt examines the case where $s = 2$. Crowley's approach is similar to taking s to be the square root of 2, while combining the above two methods in an alternating fashion. One of Burt's approaches is discussed further in the Appendix.

These methods provide good approximations to Gaussian Pyramids. A good approximation to a Laplacian Pyramid is obtained by subtracting neighboring levels in the Gaussian Pyramid.

As we discuss in the following section, when the power spectra of the resulting filters are considered, the binomial coefficients turn out to be a suitable choice for Burt's generating kernel. Interestingly, under these circumstances there is an equivalence between one of Burt's methods and the cascaded uniform filters approach. This is discussed in the Appendix.

Canny

Canny [5] discusses several interesting approaches to computing approximations of Gaussian convolutions. One is based on recursive filtering, while another is an adaptation of Strassen's algorithm for fast multiplication.

The recursive filtering method uses a two pole, two zero (four term) recursive filter, whose response is a damped exponential cosine which approximates the Gaussian. Such filters have asymmetrical responses. To overcome this, he applies the filter to the input twice, one in each direction (for each dimension) and adds the results, obtaining a symmetrical response. In order to improve the approximation to the Gaussian, the procedure may be repeated, exploiting the property of repeated convolutions as discussed above in connection with the central limit theorem.

A major attraction of this approach is that its computational complexity is independent of kernel size. For two dimensions with passes in four directions, twelve multiplications and additions are required per pixel. Two passes would lead to 24 multiplications and additions.

Canny also examines Strassen's methods for fast multiplication, with convolution replacing multiplication. One advantage is that this works for general kernels, not just those which are separable. For some kernel sizes, computational complexity is reduced by a factor of 5 or 6 compared to direct methods.

We have discussed some of the current methods used to implement Gaussian filters. In the next section, we look at another economical method.

III. Cascaded Uniform Filters

Cascaded uniform filters, as a means of implementing Gaussian filters, is economical to implement and has good filtering characteristics. We point out an equivalence to one of Burt's methods [1], [3] under certain circumstances. This is discussed more fully in the Appendix. We examine an approach to implementing a Gaussian Pyramid which beyond the first level, requires only two addition operations per pixel, per level, per dimension. The scales of this filter are related by roughly half an octave.

The repeated convolution property discussed in the previous section suggests cascading "simple" (i.e., cheap to compute) filters in order to achieve Gaussian filtering. Finite-impulse-response digital filters which have uniform coefficients are particularly economical to implement. As we will show, the convolution of a signal by a uniform finite-impulse-response function may be computed with only one addition and one subtraction per pixel, per dimension (no multiplies are needed). This is independent of the width of the impulse response. This method has been around for some time [13], [16]. McDonnell [13] discusses the synthesis of low-pass, high-pass, and Gaussian filters using this approach.

Let us define the impulse responses of a uniform-coefficient digital filter having width n:

$$b_{\mathfrak{N}}(k) = \begin{cases} 1 & 0 \le k < \mathfrak{N} \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

The response of such a filter is given by the convolution of the above with the input signal, and may be written as follows:

$$y(k) = \sum_{i=0}^{\mathfrak{N}-1} x(k-i)$$

where $x(k)$ is the input signal and $y(k)$ is the output signal. We may write

$$y(k) - y(k-1) = \sum_{i=0}^{\mathfrak{N}-1} x(k-i) - \sum_{i=0}^{\mathfrak{N}-1} x(k-i-1)$$
$$= x(k) - x(k-\mathfrak{N})$$

or

$$y(k) = y(k-1) + x(k) - x(k-\mathfrak{N}).$$

This describes a means of implementing a uniform coefficient filter with a single addition and subtraction per pixel.

Let $b_{\mathfrak{N}}^{\mathfrak{M}}(k)$ denote the impulse response of the filter consisting of M cascaded uniform coefficient filters having response $b_{\mathfrak{N}}(k)$. Then $b_{\mathfrak{N}}^{\mathfrak{M}}(k)$ is $b_{\mathfrak{N}}(k)$ convolved with itself $\mathfrak{M} - 1$ times.

Continuity

The continuity of the impulse responses of cascades of uniform filters, $b_{\mathfrak{N}}^{\mathfrak{M}}(k)$, may be studied in the context of their continuous analogs, that is, in the limit where the sample spacing goes to zero, with the width being held constant. The continuous analogs of the impulse responses are described by the following recurrence relation:

$$b_w'^{\mathfrak{M}}(x) = b_w'^{1}(x) * b_w'^{\mathfrak{M}-1}(x)$$

$$b_w'^{1}(x) = \begin{cases} 1 & 0 \le x < w \\ 0 & \text{otherwise} \end{cases}$$

where * indicates convolution.

These functions have the following properties:

function	form	continuity
$b_w'^1(x)$	piecewise constant	not continuous
$b_w'^2(x)$	piecewise linear	C^1
$b_w'^3(x)$	piecewise quadratic	C^2
$b_w'^4(x)$	piecewise cubic	C^3
	etc.	

As discussed in the previous section, the degree to which these functions approximate the Gaussian function improves with order.

Goodness of Fit to Gaussians

The following table summarizes a measure of how well the impulse responses of cascades of uniform-coefficient filters approximate Gaussian functions of the same weight and variance. The quantity shown is the rms difference computed over the nonzero region of the impulse response, divided by the weight, or sum, of the impulse response.

$\mathfrak{N}$	Number of Convolutions	rms Difference/Weight
4	1	0.180
4	2	0.043
4	3	0.021
4	4	0.014
8	1	0.127
8	2	0.026
8	3	0.013
8	4	0.009
16	1	0.090
16	2	0.018
16	3	0.009
16	4	0.006
32	1	0.064
32	2	0.012
32	3	0.006
32	4	0.004

Variance and Weight of Impulse Responses

The weight of $b_{\mathfrak{N}}(k)$ is easily seen to be $\mathfrak{N}$. Since $b_{\mathfrak{N}}(k)$ are nonnegative,

$$\text{weight } (b_{\mathfrak{N}}^{\mathfrak{M}}(k)) = \mathfrak{M}\mathfrak{N}.$$

The variance of $b_{\mathfrak{N}}(k)$ may be directly calculated and is

$$\text{var } (b_{\mathfrak{N}}(k)) = \frac{\mathfrak{N}^2 - 1}{12}.$$

As is familiar from the study of probability densities, the variance of the convolution of these functions is the sum of the variances of the individual functions. Thus,

$$\text{var } (b_{\mathfrak{N}}^{\mathfrak{M}}(k)) = \mathfrak{M}\, \frac{\mathfrak{N}^2 - 1}{12}.$$

Power Spectra

An important characteristic of low-pass filters is their rejection of frequencies outside of the designed passband. A measure of this criterion is the ratio of the highest sidelobe level to the dc response (HSSL). Continuity of the impulse response is a major determiner of the HSSL of a filter.

The power spectra of cascaded identical uniform filters may also be studied in the continuous case. The impulse responses of this case were discussed above. The HSSL here is approximately $n \cdot 13.3$ dB, where n is the number of cascaded filters. If n is chosen to be 4, then the HSSL is -53 dB.

In the Appendix we discuss a correspondence to one of Burt's techniques [3] when n is 4 and his free parameter "a" is 3/8. In Burt's scheme, "a" may be chosen instead to minimize the HSSL. Empirically carrying out this minimization reduces the HSSL to -61 dB [3].

When $n = 4$, as above, cascaded uniform filters compare favorably to Burt's approach [3].

Extension—Computation of Gaussian Pyramid

It is not necessary to keep the width of the uniform filters constant in order to obtain good low-pass filtering. To illustrate this we describe an method of implementing a Gaussian Pyramid having approximately half-octave scale steps. This filter has the recursive style of Burt and Crowley [3], [8] in that each level is obtained by applying a filter to the previous level—in this case a uniform-coefficient finite-impulse-response filter.

The following table summarizes the characteristics of the first few levels. We list the impulse response, the half-power bandwidth, and the highest sidelobe level.

Impulse Response	−3 dB BW (*2 pi rad/pixel)	Highest Sidelobe Level (dB)
$g_1(k) = b_2(k) * b_3(k) * b_4(k)$	0.078	−35.9
$g_2(k) = g_1(k) * b_6(k)$	0.116	−34.5
$g_3(k) = g_2(k) * b_8(k)$	0.083	−42.0
$g_4(k) = g_3(k) * b_{12}(k)$	0.055	−37.7
$g_5(k) = g_4(k) * b_{16}(k)$	0.0395	−43.7
$g_6(k) = g_5(k) * b_{24}(k)$	0.0275	−38.6
$g_7(k) = g_6(k) * b_{32}(k)$	0.0195	−44.3
$g_8(k) = g_7(k) * b_{48}(k)$	0.0135	−38.9
etc.		

(* denotes convolution)

Notice that beyond the first level this filter may be implemented with only two addition operations per pixel, per level, per dimension. A characteristic of this type of filter (increasing width cascade) is that as the number of levels per octave is increased (by slowing the growth rate of the size of the uniform filters' impulse responses), the HSSL figure improves.

IV. Application Issues

An important concern in choosing an algorithm for Gaussian filtering is whether results at multiple scales are desired. If full size (i.e., not reduced) results are desired at only one particular scale, then Burt and Crowley's methods [1], [8] may be more costly, since they require computation proportional to the log of the size of the kernel edge. The cascaded uniform filter we discuss (the case of cascaded identical filters) and Canny's recursive filter approach [5], have complexity that is independent of the kernel size. Which method is best would depend upon the application.

Canny's recursive filter method [5] requires 12 or 24 multiplies and adds (24 or 48 operations) per pixel for two dimensions. A cascade of four uniform coefficient filters requires 16 adds per pixel in two dimensions.

In applications where multiplication is substantially more expensive than addition, cascaded uniform-filters would be better. This applies to some computers (e.g., Motorola 68000 performing integer arithmetic). Additionally, adders have less hardware complexity than multipliers in high-speed processor implementations.

Locality and sequencing of addressing are issues to consider in addition to number and type of operations. Canny's approach requires access to a small set of nearby pixels and previous results. The size of the set is independent of the kernel size; address sequencing is two directions (per dimension) is also required.

The cascaded uniform filters approach accesses data separated by the size of the uniform filters' impulse response. This is comparable in size to the approximating Gaussian, in each dimension. It may be implemented using simple sequential addressing. In this case, for two-dimensional convolutions, temporary storage would be needed for n pixels and n rows for a filter of size n. In situations where more flexibility in addressing is available, the storage requirement can be reduced. An example of this would be a processor where the addressing sequence can be changed from row major to column major. Here, the convolution may be carried out in one dimension with one mode of addressing, then in the other dimension with the other mode of addressing.

In contexts where a full (nonreduced) pyramid is desired, any of the methods discussed above may be considered, since all (except Canny's adaptation of Strassen's algorithm) have comparable complexity to generate the next level of the pyramid. Burt's methods [1]–[3] use octave scale spacing. Crowley's approach [8], and the method we described at the end of the previous section, (cascaded increasing size uniform filters) are designed for half-octave scale spacing. Cascaded uniform filter methods, as well as Canny's [5], could generate nearly arbitrary scale spacings. Here again, a choice would depend on the relative cost of addition and multiplication, as well as addressing issues.

V. Implementation

We have implemented Gaussian filtering using cascaded uniform filters in a straightforward fashion on a VAX 11/780 in the C language. Approximately 80 seconds are required for a four pass implementation on 512 by 512 by 8 bit images.

VI. Summary and Conclusion

Gaussian convolutions have proved to be a useful operation in computer vision, data compression, and image processing.

Cascaded uniform filters, each of which may be implemented with a single addition and subtraction per dimension, have been shown to provide an economical means for computing Gaussian-like convolutions.

We have described an efficient method of computing a Gaussian Pyramid which uses one uniform-coefficient filter to implement each level beyond the first.

These methods are appropriate for implementation in pipelined processors. Real-time processing of video imagery is a natural application. These methods are also suitable for general purpose computers. Using a high level language, execution time on the order of one minute is practical on video image data.

Appendix
Correspondence to Burt's Method

The following demonstrates a correspondence between a cascade of uniform coefficient filters and Burt's filtering techniques [1], [3] when binomial coefficients are used as his generating kernel.

Consider the result $y(k)$ of applying a cascade of $\mathfrak{M}$ uniform coefficient filters having impulse responses of width 2^m to an input function $x(k)$.

$$y(k) = \underbrace{b_{2^m}(k) * b_{2^m}(k) * \cdots b_{2^m}(k)}_{\mathfrak{M}} * x(k)$$

where $*$ denotes convolution and $b_n(k)$ is defined in Section III, (1).

Taking the Z transformation:

$$Y(Z) = B_{2^m}^{\mathfrak{M}}(Z)\, X(Z);\; B_N(Z) = \sum_{i=0}^{N-1} Z^{-i}$$

(here $B_{2^m}^{\mathfrak{M}}$ is B_{2^m}, the transform of b_{2^m}, raised to the $\mathfrak{M}$ power).

Define the auxiliary functions

$$h_s(k) = \delta(k) + \delta(k - 2^s)$$

with transformations

$$H_s(Z) = 1 + Z^{-2^s}. \tag{2}$$

The following recurrence relation holds among the B_N and H_N:

$$B_{2^m}(Z) = H_{m-1}(Z)\, B_{2^{m-1}}(Z). \tag{3}$$

Another useful property of these functions is that

$$h_{m-1}^{\mathfrak{M}}(k) = \sum_j \binom{\mathfrak{M}}{j} \delta(k - j2^{m-1}). \tag{4}$$

This is demonstrated by examining

$$H_{m-1}^{\mathfrak{M}}(Z) = (1 + Z^{-2^{m-1}})^{\mathfrak{M}}$$

from (2), using the binomial theorem

$$H_{m-1}^{\mathfrak{M}}(Z) = \sum_j \binom{\mathfrak{M}}{j} Z^{-(j2^{m-1})}$$

and taking the inverse Z transform.

Note that in (4) $h_{m-1}^{\mathfrak{M}}$ refers to h_{m-1} convolved $\mathfrak{M}$ times, not raised to the $\mathfrak{M}$ power as with $H_{m-1}^{\mathfrak{M}}$.

From (3) we may write

$$B_{2^m}^{\mathfrak{M}}(Z) = H_{m-1}^{\mathfrak{M}}(Z)\, B_{2^{m-1}}^{\mathfrak{M}}(Z).$$

Taking the inverse Z transform and using the convolution summation property of the Z transform results in

$$b_{2^m}^{\mathfrak{M}}(k) = \sum_{i=0}^{\infty} h_{m-1}^{\mathfrak{M}}(i)\, b_{2^{m-1}}^{\mathfrak{M}}(k - i).$$

Substituting (4) leads to

$$b_{2^m}^{\mathfrak{M}}(k) = \sum_{i=0}^{\infty} \sum_{j} \binom{\mathfrak{M}}{j} \delta(i - j2^{m-1})\, b_{2^{m-1}}^{\mathfrak{M}}(k - i)$$

or

$$b_{2^m}^{\mathfrak{M}}(k) = \sum_{j} \binom{\mathfrak{M}}{j} b_{2^{m-1}}^{\mathfrak{M}}(k - j2^{m-1})$$

Let $f_m^{\mathfrak{M}} = b_{2^m}^{\mathfrak{M}}$. Then

$$f_0^{\mathfrak{M}}(k) = b_1^{\mathfrak{M}}(k) = \delta(k)$$

$$f_m^{\mathfrak{M}}(k) = \sum_{j} \binom{\mathfrak{M}}{j} f_{m-1}^{\mathfrak{M}}(k - j2^{m-1}). \tag{5}$$

This recurrence relation is a property of the impulse responses which are equivalent to $\mathfrak{M}$ successive convolutions by a uniform-coefficient finite-impulse-response function of width 2^m.

One formulation of Burt's approach, hierarchical discrete correlation [1], [3], has equivalent kernels described by the following recurrence relation (for one dimension):

$$g_0(k) = \delta(k)$$

$$g_m(k) = \sum_{j=-n}^{n} w(j)\, g_{m-1}(k - j2^{m-1}) \tag{6}$$

where $w(i)$ is a symmetrical generating kernel.

Notice the similarity of (5) and (6). If Burt's generating kernel is taken to be a shifted binomial coefficient distribution, then his result at level m is equivalent to $\mathfrak{M}$ successive convolutions by a boxcar of width 2^m, up to a displacement of the kernel.

In the method described above, Burt suggests a five-element generating kernel of the form

$$w(0) = a$$

$$w(1) = w(-1) = \tfrac{1}{4}$$

$$w(2) = w(-2) = \tfrac{1}{4} - a/2.$$

This is constrained to be normalized, symmetrical, and to have a property called "equal contribution."

If the free parameter is chosen as follows:

$$a = \tfrac{3}{8}$$

then

$$w(j) = \binom{4}{j+2} \cdot \frac{1}{16}.$$

This case is equivalent to a cascade of four uniform-coefficient finite-impulse-response filters.

References

[1] P. J. Burt, "Fast, hierarchical correlations with Gaussian-like kernels," Comput. Vision Lab., Univ. Maryland, Tech. Rep. TR-860, Jan. 1980.

[2] P. J. Burt and E. H. Adelson, "The Laplacian Pyramid as a compact image code," *IEEE Trans. Commun.*, vol. COM-31, pp. 532–540, Apr. 1983.

[3] P. J. Burt, "Fast algorithms for estimating local image properties," *Comput. Vision, Graphics, Image Processing*, vol. 21, pp. 368–382, Mar. 1983.

[4] M. Brady and H. Asada, "Smoothed local symmetries and their implementation," *Int. J. Rob. Res.*, vol. 3, pp. 36–61, Fall 1984.

[5] J. F. Canny, "Finding edges and lines in images," Master's thesis, Massachusetts Inst. Technol., A.I. Lab. Tech. Rep. 720, June 1983.

[6] J. L. Crowley and A. C. Parker, "A representation for shape based on peaks and ridges in the difference of low-pass transform," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, pp. 156–169, Mar. 1984.

[7] J. L. Crowley and A. C. Sanderson, "Multiple resolution representation and probabilistic matching of 2-D gray scale shape," in *Proc. IEEE Workshop on Vision, Representation and Control*, Mar. 1984, pp. 95–105.

[8] J. L. Crowley and R. M. Stern, "Fast computation of the difference of low-pass transform," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, pp. 212–222, Mar. 1984.

[9] W. Feller, *An Introduction to Probability Theory and Its Applications.* New York: Wiley, 1957.

[10] B. R. Frieden, *Probability, Statistical Optics, and Data Testing.* Berlin: Springer-Verlag, 1983.

[11] D. Marr and E. Hildreth, "Theory of edge detection," *Proc. Roy. Soc. London*, vol. B 207, pp. 187–217, 1980.

[12] D. Marr, *Vision.* San Francisco, CA: W. H. Freeman, 1982.

[13] M. J. McDonnell, "Box filtering techniques," Comput. Vision Lab., Univ. Maryland, Tech. Rep. TR-966, Nov. 1980.

[14] W. Nicholson and K. Davis, "The binomial window," in *Proc. SPIE Image Processing for Missile Guidance*, vol. 238, 1980, pp. 467–479.

[15] J. Ponce and M. Brady, "Toward a surface primal sketch," in *Proc. IEEE Int. Conf. Rob. Automation*, 1985, pp. 420–425.

[16] K. E. Price, "Change detection in multi-spectral images," Ph.D. dissertation, Carnegie-Mellon Univ., Pittsburgh, PA, Dec. 1976.

[17] A. P. Witkin, "Scale-space filtering," in *Proc. 7th Int. Joint Conf. Artificial Intell.* Palo Alto, CA: Kaufmann, 1983, pp. 1019–1021.

William M. Wells, III (M'85) received the B.A. degree in physics from the University of California, Santa Cruz, in 1976, and the M.S. degree in computer science/computer engineering from Stanford University, Stanford, CA, in 1984.

He is currently with SRI International, Menlo Park, CA. His research interests include computer vision, robotics, and parallel processing.