

## Analog 2-D video processing and its application to wavelet filter banks

M. S. Moore, B. Amrine, and J. Langan

Computational Sensors Corporation  
201 N. Calle Caesar Chavez, Suite 203  
Santa Barbara, CA 93103

### ABSTRACT

In this paper we introduce a biologically-inspired spatial video filtering chip and discuss its application in wavelet filter banks. Two types of spatial filtering chips have been developed – the thin film analog image processor (TAIP) and the switched-capacitor analog image processor (SCAIP). Each chip can filter video at high frame rates with Gaussian-like filters having adjustable widths. By linearly combining the outputs of a bank of spatial filter chips we can create a large variety of filters.

Effective use of the filtering chips requires two things. First, an assessment of the filters that can be realized within the constraints of the hardware is necessary. Although any function within reasonable constraints can be decomposed into a combination of Gaussian functions, an efficient method to do so is an open problem. We have restricted ourselves to a simpler problem – given a limited number of Gaussian-like functions, what useful classes of filters can be generated? Second, given an image-processing application, a method to organize a choice of filters is needed. We are currently investigating these problems in the context of feature analysis/discrimination, and have found a useful organizing principle in the continuous wavelet transform (CWT).

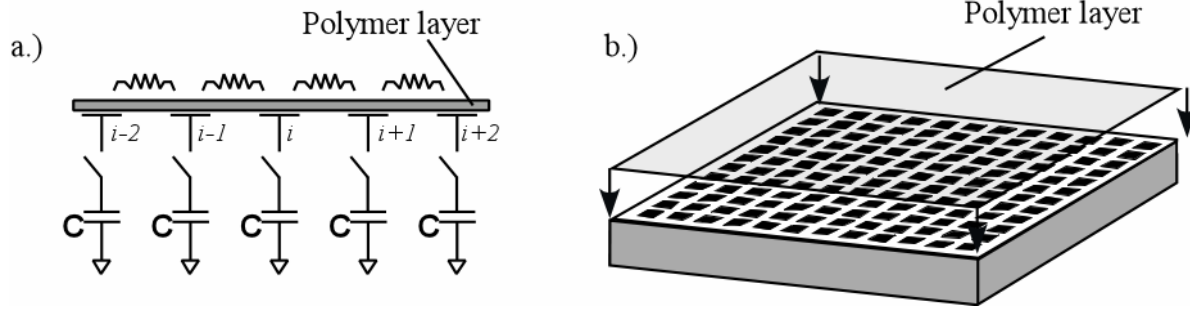
**Keywords:** massively parallel VLSI image processor, continuous wavelet transform, discrimination

### 1. INTRODUCTION

The most common way to perform image and video processing is to use digital processors such as general-purpose computers or digital signal processors (DSPs). The primary advantage of this approach is its flexibility. Given enough resources, almost any algorithm can be implemented digitally. These devices are programmable, so the algorithms can be adjusted after the initial implementation. The most significant source of error results from the limited precision of digital devices. To some extent, increasing the resources of the system can alleviate this problem.

Spatial filtering of video using digital processors is challenging for two reasons. First, the amount of memory required is staggering. One second of normal NTSC color video requires more than 20 megabytes (Mb) of data storage. If this video is input to an un-decimated filter bank with five outputs, the 20 Mb gets expanded into 100 Mb. Therefore, practical digital image processing systems normally use algorithms with reduced memory requirements. If only a spatial filter is required and a three-by-three filter is satisfactory, only just over two lines of video (a few kilobytes) needs to be stored to implement the filter. This small amount of data is satisfactory for DSP implementations. However, as the filter size increases, the number of lines required increases.

As memory becomes cheaper, numbers that were large a few years ago, such as 100 Mb of data, are no longer impractical. However, a second disadvantage of digital processing still arises. Digital signal processors generally process pixel values *serially*. Although it can be stored, processing 100 Mb of data still requires significant time. The processing time also depends on the size of the filter. As the filter impulse response size increases, the number of memory accesses required also increases. Using multiple processors running in parallel can reduce the processing time. In this arrangement, each processor processes its own portion of the video data. However, dividing the data may not be practical in some applications. In addition, the amount of power required increases as processors are added. The ultimate limitation may simply be that there is inadequate power available to support a bank of digital processors.



**Figure 1.** (a) One-dimensional schematic of an analog VLSI image processor. The physical two-dimensional device is shown in (b) where a CMOS 256x256 array of capacitor unit cells is overlain with a continuous, isotropic or anisotropic, doped polyaniline/regioregular poly(3-alkylthiophenes) electroactive polymer layer.

In contrast, analog VLSI image processors can perform filtering operations with all pixel values computed in parallel. This approach was originally inspired by the biology of the retina<sup>1</sup>. The retina performs many image processing operations, including contrast enhancement, edge detection, and other filtering operations. These filters are implemented using a large, 2-D arrays of neurons. The interconnections among the neurons share information in an excitation/inhibition network that is thought to create several versions, or channels, of the input image with different spatial frequency and orientation sensitivities<sup>2</sup>.

Several VLSI image processing solutions have been proposed that include retina-like models<sup>1,3,4,5</sup>. These chips are similar in that the incoming data is organized in a 2-D array of pixels and the processing is performed by the pixel interconnecting circuitry. Beyond that, the different approaches vary in the complexity of the interconnections and the form of the data (analog or digital). At Computational Sensors Corporation (CSC), we have been working with two kinds of Analog Image Processors (AIPs). These chips work with analog signals and use simple methods of interconnecting pixels. This approach allows us to build fast chips capable of processing large format images. The different analog processor chips are discussed in more detail in the following section.

## 2. SPATIAL FILTERING CHIP

The TAIP performs spatial filtering by loading each successive frame into an array of capacitors. The array is laid out similarly to a focal plane array. However, in each cell, instead of a photo-detector, there is a connection through a switch to a resistive polymer layer. An illustration of the TAIP is shown in Figure 1. With the charge on each capacitor corresponding to the image level at a pixel, the image thus represented will be blurred, or low-pass filtered, when the switches are closed for a time, allowing charge to transfer among the capacitors. Starting from Kirchoff's laws, one can develop a mathematical analysis of this diffusion. In one dimension, and with edge-effects ignored, the modulation transfer function of this blurring filter is approximately

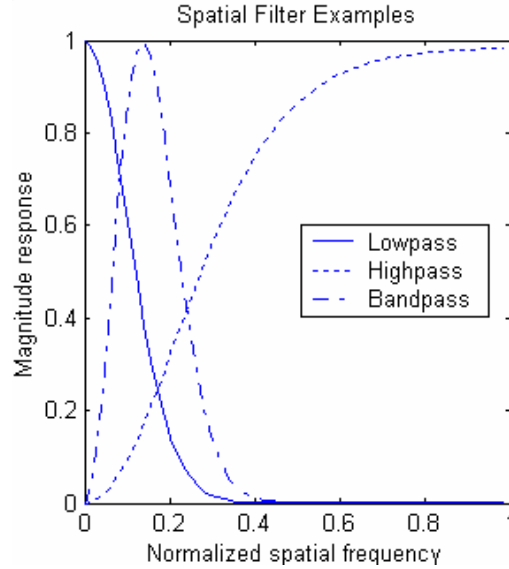
$$MTF(k) = \exp\left(-\frac{t}{RC} 4 \sin^2\left(\frac{1}{2} k\right)\right), \quad (1)$$

where  $k$  is the discrete spatial frequency along one axis,  $t$  is the time during which the charge flows, and  $R$  and  $C$  are characteristics of the chip<sup>6,7</sup>. Alternatively, the filter can be described in terms of its impulse response, or convolution kernel, which can be approximated by the function

$$h(x) = \exp\left(-\frac{2t}{RC}\right) I_x\left(\frac{2t}{RC}\right) \quad (2)$$

Here  $x$  is pixel number,  $t$ ,  $R$ , and  $C$  are as before,  $I_x(\bullet)$  is the modified Bessel function of order  $x$ , and the approximation is, again, for the one-dimensional case and far from an edge<sup>7</sup>.

If the resistive layer has equal resistance in all directions, the resulting filter will be isotropic and Equations 1 and 2 can be generalized to two dimensions. If the resistive layer resistance depends on orientation, the resulting filter will be



**Figure 2.** Magnitude response example for the spatial TAIP chip. With the exception of the spatial bandpass filter, all of these responses were generated using a single chip. The spatial bandpass was generated by cascading a lowpass and a highpass filter (two chips).

anisotropic and the 2-D response is more complicated. Polymer layers can be made to be deliberately anisotropic. This may be useful in some applications where orientation-dependent filtering is desired.

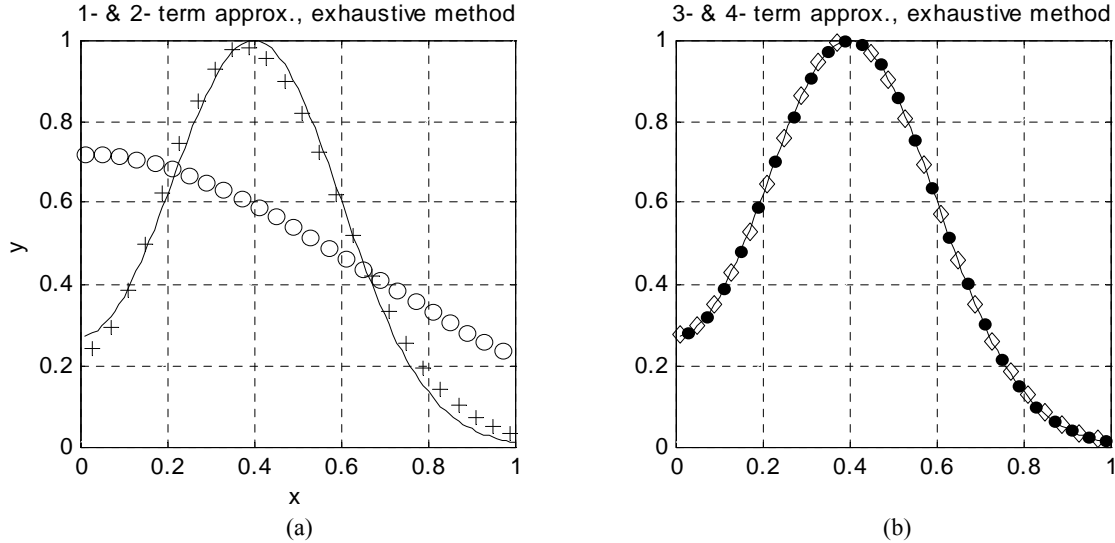
Varying  $t$  controls the characteristics of the TAIP lowpass filter. The value of  $t$  is typically small, ranging from less than a microsecond to about a millisecond. As a result, video frames can be processed at high frame rates and the rate does not depend on the size of a frame. However, some flexibility has been lost. A digital processor can implement any arbitrary filter. The resistive layer can only be used to directly implement the linear lowpass filter specified in Equation (1). To provide more flexibility, the TAIP has been designed to include an extra capacitor per pixel. This capacitor is also charged with a value proportional to the original pixel value. However, it is not connected to the resistive layer. Therefore, these capacitors retain the original frame even after a filtering operation. The TAIP also includes comparators that allow the difference between the original and lowpass-filtered values to be calculated for each pixel. The result is a highpass filter that is the delay complement of the lowpass filter in Equation (1). More complex filters can be implemented by combining variably-scaled versions of the outputs of multiple TAIPs in cascade, parallel, or other structures. Figure 2 contains a plot of some simple spatial filter realizations.

The TAIP effectively performs a massive, parallel, convolution operation. One significant parameter – the spatial extent of the filter – is directly controllable. However, the degree of anisotropy of the filter is fixed by the manufacturing process. In addition, variability in the process of adding the polymer leads to variability in the chip characteristics. To reduce this variability, another image processor is being developed that replaces the polymer layer with several simple switched-capacitor interconnections. This processor is called the switched-capacitor analog image processor (SCAIP). The resulting MTF has a similar form to Equation (1), except that the blur in each of the interconnection directions can be controlled separately. Therefore, the anisotropy of the chip can be controlled. This increases the flexibility of the video processing circuitry.

### 3. APPROXIMATION OF IDEAL FILTERS

As mentioned above, CSC has image-processing hardware that allows for the combination of several analog image processor outputs. Of the system architectures that have been implemented, perhaps the most versatile and amenable to analysis is linear combination, wherein the results of filtering an image by several analog processors are multiplied by some choice of coefficients and then added. In general, the filters implemented by the chips will have differing spatial extent. Also, an allpass filter mode, which will delay but otherwise pass the input image along unaltered, is assumed.





**Figure 4.** One- through four-term approximations by zero-mean Gaussians to a smooth band-pass function as found by an exhaustive search. For clarity, the results are shown in two plots: (a) one- and two-term, (b) three- and four-term. The function to be approximated is  $f_{BP}$  as defined in Equation 5. In the plot, the function  $f_{BP}$ , the one-term, two-term, three-term, and four-term approximants are shown by the solid curve, the circles, the plus signs, the diamonds, and the dots, respectively. The approximants with parameters and approximation errors are listed in Table 1.

Each of our approximations will correspond to a choice of a positive integer  $n$  and a pair of real  $n$ -vectors  $\alpha = [\alpha(1), \dots, \alpha(n)]$  and  $s = [s(1), \dots, s(n)]$ , where the components of  $s$  belong to  $S$ . The linear combination of Gaussians determined this way will be denoted  $G(\alpha, s; \bullet)$  and defined by

$$G(\alpha, s; x) = \sum_{i=1}^n \alpha(i)g(s(i); x). \quad (4)$$

For the results reported here, we measured the error of approximating  $f$  by  $G$  as the root mean square of the values of  $(f - G)$ , evaluated at a hundred evenly-spaced points between zero and one.

Among the various algorithms used to find Gaussian approximations of the type represented in Equation 4, there are two that illustrate some issues in rather extreme forms: (1) an exhaustive search, and (2) an incremental, “greedy” procedure that we have specialized from a general method presented by Kurkova and Beliczynski<sup>8</sup> (KB). Here we shall summarize the results of applying both algorithms to the approximation of a smooth band-pass type function  $f_{BP}$  defined by

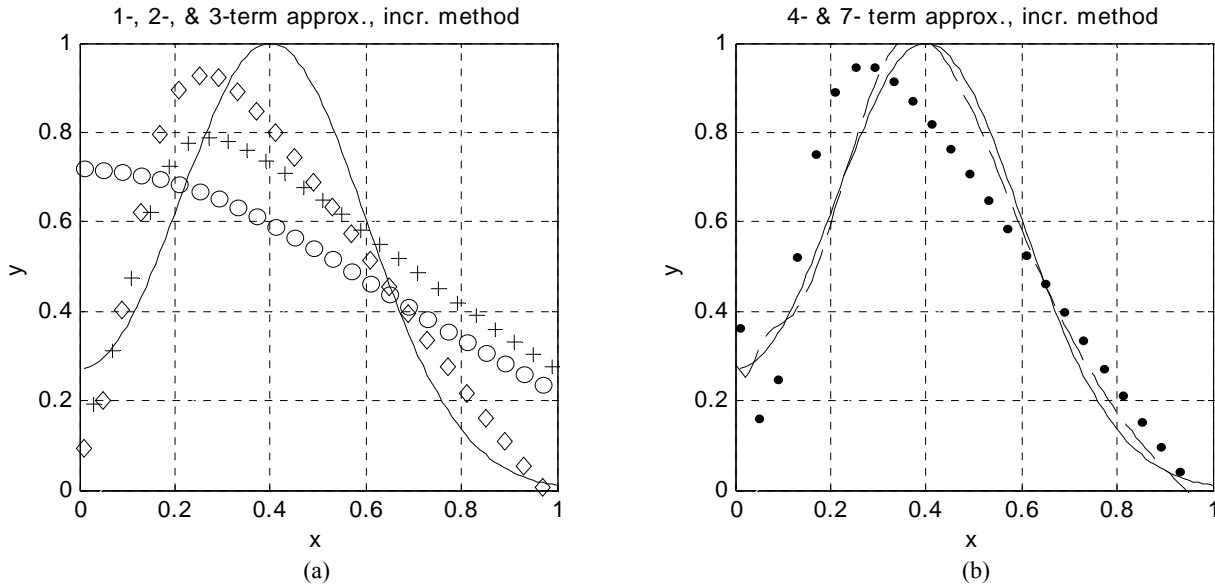
$$f_{BP}(x) = \sum_{i=1}^2 \exp\left(-\frac{1}{2}\left(\frac{x-m_i}{s}\right)^2\right). \quad (5)$$

with  $s$  equal to 0.2,  $m_1$  equal to -0.4, and  $m_2$  equal to 0.4. Note that while this is a linear combination of Gaussians, they are not zero-mean Gaussians, as are the basis functions for our approximation. Therefore, the function can not be simply implemented using two of the basic analog processors.

In the exhaustive search, for a given  $n$ , we run through all choices of  $n$ -vectors  $s$  with components in  $S$ , and for each of these, we choose the coefficient list  $\alpha$  that minimizes the RMS value of  $(f-G)$ . Since the number of choices is given by the binomial coefficient (choose  $n$  from 132), the time to complete this algorithm grows very rapidly with increasing  $n$ . For the tests described in this paper, the 1- and 2-term approximations took a few seconds or fractions thereof, the 3-term approximation took several minutes, and the 4-term approximation took hours. The search for the 5-term approximation by this method would have taken a few days. While this method will probably not be fast enough for our purposes when  $n > 2$ , it does produce the best possible approximations, and can provide a standard against which to compare other algorithms we expect to develop in the future. The 1-, 2-, 3-, and 4-term Gaussian approximants this method produced for our test function  $f_{BP}$  are shown in Figure 4. For the actual parameters of the approximations, see Table 1.

**Table 1.** Gaussian approximants to  $f_{BP}$  found using exhaustive search. Also see Figure 4 and Equations 3—5.

Fit	$\alpha$	$s$	RMS error
1-term	0.720	0.650	2.6
2-term	19.6, -19.3	0.291, 0.273	0.31
3-term	102, -160, 58.0	0.252, 0.240, 0.225	0.035
4-term	90.2, -297, 353, -146	0.246, 0.225, 0.213, 0.205	0.0052



**Figure 5.** One- through four-term and seven-term approximations by zero-mean Gaussians to a smooth band-pass function as found by an iterative, incremental method. For clarity, the results are shown in two plots: (a) one- and two-, and three-term, (b) four-, and seven-term. The function to be approximated is  $f_{BP}$  as defined in Equation 5. In the plot, the function  $f_{BP}$ , the one-term, two-term, three-term, four-term, and seven-term approximations are shown by the solid curve, the circles, the plus signs, the diamonds, the dots, and the dashed line, respectively. The approximants with parameters and approximation errors are listed in Table 2.

In this and other test cases, the exhaustive method showed some properties that are undesirable in our intended applications: (1) it achieves its close approximations by employing pairs of Gaussians with similar standard deviations whose corresponding coefficients have opposite sign and absolute values of about the same order of magnitude, and (2) it uses coefficients with high absolute values. The first property is a drawback because it promotes instability – the evaluations of linear combinations are liable to be too sensitive to changes in the values of their determining parameters. The second characteristic is problematic for many reasons, such as power consumption and noise amplification.

In the second algorithm, a variant of the KB method<sup>8</sup>, we iteratively and incrementally improve an existing approximation to some function  $f$  by adding another Gaussian function to the set currently used to make the approximation. The search for the new Gaussian  $g(s, \bullet)$  consists of finding the  $s$  in  $S$  which maximizes the correlation between  $g(s, \bullet)$  and the current error function. At the beginning, we have no Gaussians, our approximation is the function constantly zero, and our error function is  $f$ . Each time the set of Gaussians is augmented, coefficients are adjusted to minimize the energy of the difference between  $f$  and a linear combination of the new set. The process is continued until the set of Gaussians has size equal to the given  $n$ . This method is fast, and the time is approximately a linear function of  $n$ . In our example, the computation of the 4-term approximations was done in under a tenth of a second (compared with several hours on the same computer system for the  $n = 4$  case by the exhaustive method). For graphs of the 1-, 2-, 3-, 4-, and 7- term approximations to our test function, as found by this iterative method, consult Figure 5. For the parameters of the approximations and the related errors, see Table 2.

Comparing the approximants found by the two methods, one observes that the errors are larger for this method as compared with the exhaustive method, for the same number of Gaussians,  $n$ . Indeed, a value of  $n = 7$  was required for

**Table 2.** Gaussian approximants to  $f_{BP}$  found using incremental optimization. Also see Figure 5 and Equations 3—5.

Fit	$\alpha$	$s$	RMS error
1-term	0.720	0.650	2.6
2-term	0.880, -0.718	0.650, 0.101	2.0
3-term	1.61, -1.00, -0.524	0.650, 0.101, 100	1.4
4-term	1.69, -1.32, -0.563, 0.569	0.650, 0.101, 100, 0.0349	1.3
7-term	0.929, 0.718, -0.344, -0.196, -3.11, 2.21, 0.081	0.650, 0.101, 100, 0.0349, 0.196, 0.312, 0.0125	0.30

the incremental method to match or surpass the results produced with  $n = 2$  by the exhaustive method. To beat the exhaustive method with  $n = 4$ , one had to go to  $n = 16$  with the iterative method. This tendency to require large values of  $n$  would be a disadvantage in applications with our hardware, since it corresponds to use of more of the basic image-blurring chips. On the other hand, in the examples examined, the iterative method showed the advantage of finding approximants with small coefficients. In none of these cases did the coefficients have absolute value in excess of 32, which is the bound imposed by our current system architecture.

We have investigated two methods for finding approximations to a function by linear combinations of a finite family of zero-mean Gaussians. While each method has drawbacks which make it unsuitable for our purposes, each also has advantages. It seems possible that features of each of these methods may be included in some algorithm yet to be developed. In future studies, we expect to abandon the generalized Gaussian model for models more explicitly representative of the hardware realizations of the basic spatial filters. Also, where appropriate, we intend to measure the approximation errors by metrics adapted to the character of the filter being approximated. We are presently considering some version of a simulated annealing type of algorithm<sup>9</sup> in hopes of getting faster, more robust results.

#### 4. APPLICATION OF THE CWT TO OBJECT DISCRIMINATION / RECOGNITION

The image processors we have developed are suitable for a variety of applications. The focus in this paper is on image analysis for the discriminating amongst similar targets embedded in a cluttered background. The objects-of-interest may appear at different locations, scales, and orientations. Because our focus is on analysis for discrimination, we are not interested in the reconstruction properties of a particular transformation. In addition, we have a powerful spatial filtering solution, capable of large-scale spatial filtering operations that would normally be undesirable for computational reasons. Therefore, we have been investigating over-complete wavelet transformations. In particular, the system architecture will be capable of implementing a version of the continuous wavelet transform (CWT).

Our initial ideas for discrimination algorithms have been inspired by a paper of Antoine, Vandergheynst, Bouyoucef, and Murenzi<sup>10</sup>, in which the authors present a strategy for automatic target detection and recognition using the CWT in two dimensions. To explain this we first summarize the CWT in the terms used by these authors and show how our image processors can be used to compute it. In the context of our application, a wavelet is a function with zero mean and defined on the same domain as our image function. Given a real- (or complex-) valued function  $\psi$  defined on the plane, a position vector  $p$ , a scalar  $a > 0$ , and an angle  $\theta$ , we can transform  $\psi$  by translation, dilation, and rotation into another such function  $\psi(a, \theta, p; \bullet)$ , defined by

$$\psi(a, \theta, p; x) = \frac{1}{a} \psi \left( r_{\theta} \left( \frac{x - p}{a} \right) \right) \quad (6)$$

where  $r_{\theta}$  denotes rotation by the angle  $\theta$ . If  $f$  is an image, then its CWT with respect to a wavelet  $\psi$  is the function  $CWT_{\psi, f}$  defined by

$$CWT_{\psi, f}(a, \theta, p) = \langle \psi(a, \theta, p; \bullet) \mid f \rangle \quad (7)$$

Here  $\langle \bullet | \bullet \rangle$  denotes the scalar product of functions. By working with the definition of the scalar product as an integral, one can see that

$$CWT_{\psi,f}(a, \theta, p) = ((\psi(a, \theta + \pi; \bullet) \otimes f)(p)) \quad (8)$$

using  $\psi(a, \theta; \bullet)$  to mean  $\psi(a, \theta, 0; \bullet)$ , and  $\otimes$  to represent convolution of functions. Equation 8 shows that we can approximate values of  $CWT_{\psi,f}$  using one of our image processors if we can use linear combinations of its basic convolution kernels to approximate dilated and rotated versions of the wavelet  $\psi$ .

To get information about a particular object in the image  $f$ , Antoine et al suggest fixing  $p$  at some location in the object, such as its centroid, and looking at  $CWT_{\psi,f}(a, \theta, p)$  as a function of  $(a, \theta)$ , for an appropriately chosen wavelet  $\psi$ . If there is a maximum at some coordinate pair  $(a_M, \theta_M)$ , one hopes that it can serve as a measurement of the size and orientation of the object. More generally, the authors suggest that the function  $(a, \theta) \rightarrow CWT_{\psi,f}(a, \theta, p)$ , could serve as a signature of the object centered at  $p$ , serving to distinguish it from other objects. In the next section, we report some results from an initial implementation of this scheme using our analog image processors.

## 5. EXAMPLE: OBJECT-SIZE DETECTION

As a test case for using our hardware to recognize or distinguish image objects, we pursued the ideas discussed in the preceding section with simplified inputs and a simple goal: estimate the size, in pixels, of various regular objects embedded in a noisy background. For a wavelet to implement the CWT, we worked with a family constructed by taking the difference of Gaussian functions, with coefficients adjusted to make the mean equal zero. More specifically, for positive constants  $s_1$  and  $s_2$ , we obtained a wavelet  $\psi$  by defining

$$\psi(x) = s_1^{-2} \exp(-0.5\langle x | x \rangle / s_1^2) - s_2^{-2} \exp(-0.5\langle x | x \rangle / s_2^2) \quad (9)$$

for every position vector  $x$ . For convenience and clarity, we restrict ourselves here to the case where  $s_1 = 0.5$  and  $s_2 = 1.0$ . Since this  $\psi$  is invariant under rotation, Equation 8 simplifies to

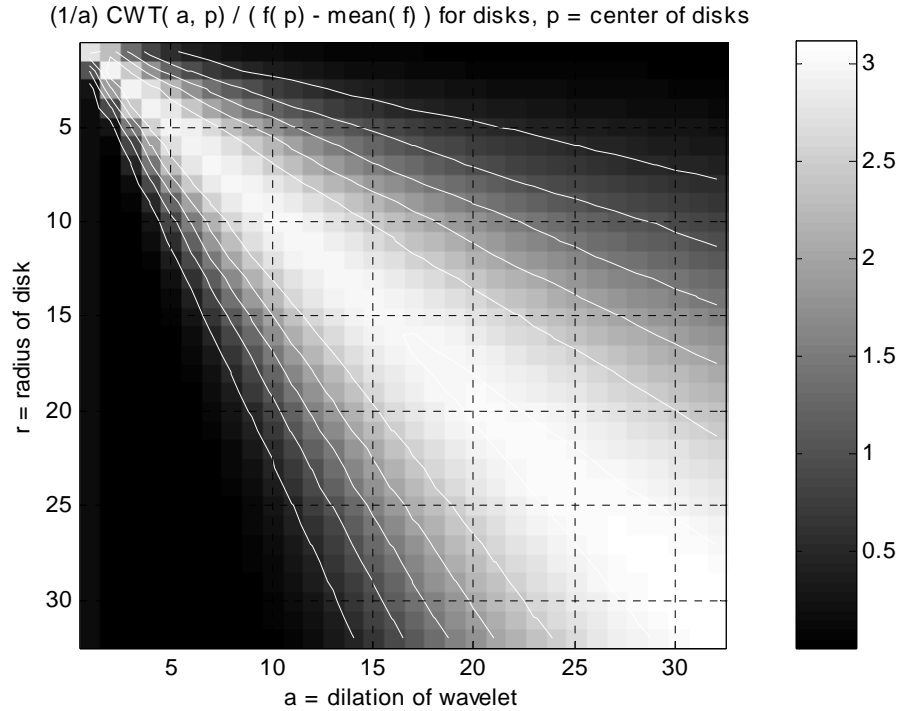
$$CWT_{\psi,f}(a, p) = (\psi(a; \bullet) \otimes f)(p) \quad (10)$$

with the definitions  $CWT_{\psi,f}(a, p) = CWT_{\psi,f}(a, 0, p)$  and  $\psi(a; \bullet) = \psi(a, 0; \bullet)$ .

With this particular choice of  $\psi$  (Equation 9,  $s_1 = 0.5$ ,  $s_2 = 1.0$ ), we performed a series of experiments, wherein we computed  $CWT_{\psi,f}(a, p)$  using Equation 10 with  $p$  located within various explicitly designed objects in synthetic images  $f$ . These confirmed a systematic relationship between the value of  $a$  and the size, or scale, of a family of objects with the same shape. With  $p$  fixed at the centroid of an object of a particular shape,  $CWT_{\psi,f}(a, p)$  plotted against the dilation  $a$  of the wavelet and the scale  $r$  of the object showed a clear tuning effect between the two coordinates. Division of  $CWT_{\psi,f}(a, p)$  by  $a$  gives the plot more symmetry between  $a$  and  $r$ ; while division by  $f(p) - \text{mean}(f)$  makes it invariant under constant scaling or offsetting of the image levels. Figure 6 shows a plot of  $(1/a) CWT_{\psi,f}(a, p) / (f(p) - \text{mean}(f))$  versus wavelet dilation  $a$  and object scale  $r$  for our special choice of  $\psi$ , in the case where  $p$  is the center of disks, and  $r$  is disk radius. In this case, the values of this normalized CWT are constant along the lines where  $a/r$  is constant. In particular, the maximum value occurs for  $a/r = 1.03$ , approximately. Such plots had a similar character for objects of other “compact” shapes, such as squares, equilateral triangles, and so on.

Contemplation of Figure 6 has led to various methods for size-detection schemes. For example, if we suppose that  $p$  is at the center of a disk, then we could estimate its radius by computing the CWT with our special wavelet for many choices of dilation  $a$ , since the radius would have a known ratio to the value of  $a$  that maximizes the CWT. We would like an efficient way to use one of our image processors to estimate object size. Although we are developing a method using disk inputs, ideally the algorithm should not depend too much upon the assumption that the objects are disks. To this end we carried out an experiment.

For  $a = 1.0, \dots, 19.0$  we were able to find a good approximation to  $\psi(a; \bullet)$  by some two-term linear combination  $h_a$  of basic TAIP convolution kernels, using an exhaustive search as discussed in Section 3. This was done in simulation,



**Figure 6.** Specially normalized  $CWT(a, p)$  for several images containing disks, with  $p$  at their centers, computed using the special wavelet defined in Section 5. Contour lines have been superimposed on the plot to improve legibility. Shows the tuning effect between  $a$  and  $r$ . Value is constant along lines of the form  $a/r = \text{constant}$ . Maximum value occurs on the line for which the constant is approximately 1.03.

using a model for the kernels analogous to the one given in Equation (2) for the one-dimensional case. (Here,  $\psi$  is the special wavelet defined in Equation 9 with  $s_1 = 0.5$  and  $s_2 = 1.0$ .) Next we computed data similar to that plotted in Figure 6, but in place of  $\psi(a; \bullet)$ , we used its approximant  $h_a$ , resulting in a matrix

$$\mathbf{M}(a, r) = a^{-1}(h_a \otimes f)(p) / (f(p) - \text{mean}(f)) , \quad (11)$$

where  $p$  is the center of a disk of radius  $r$  in the image  $f$ ,  $a = 1, \dots, 19$ , and  $r = 1, \dots, 32$ . This matrix  $\mathbf{M}$  holds our “training” data.

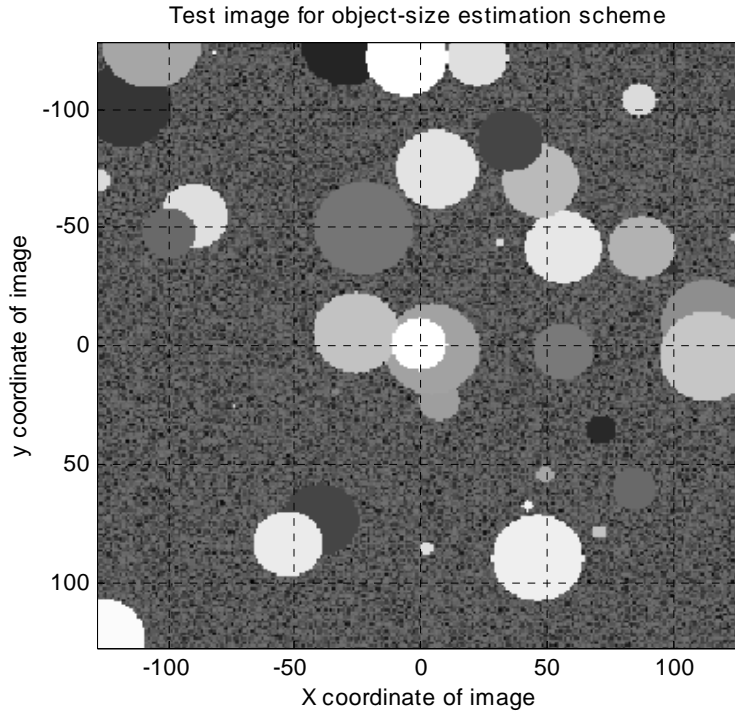
For a positive integer  $n$  (not less than 3), we chose  $n$  of the values of  $a$  for which  $\mathbf{M}(a, r)$  is defined, say  $a_1, \dots, a_n$ . (This choice was made empirically, to improve the procedure described below.) Given some general point  $q$  in a general image  $g$ , we applied this method to determine an object-size estimate  $\rho(q, g)$ : (1) Compute

$$\mathbf{V}(a) = a^{-1}(h_a \otimes g)(q) / (g(p) - \text{mean}(g)) , \quad (12)$$

for  $a = a_1, \dots, a_n$ . (2) Choose the  $r$  that minimizes the Euclidean norm of the difference between the  $n$ -vectors  $(V(a_1), \dots, V(a_n))$  and  $(M(a_1, r), \dots, M(a_n, r))$ . This  $r$  is the object-size estimate  $\rho(q, g)$ :

The set of dilations  $a_1, \dots, a_n$  should be as small as possible to minimize computation. Using  $n = 3$  and choosing  $(a_1, a_2, a_3) = (1, 10, 19)$  seems plausible and works as well or better than any other choice of 3. We have yet to study whether making  $n$  larger significantly improves the accuracy of size estimates.

With this choice we applied our object-size estimation scheme to several test images including the one illustrated in Figure 7. The image shown there was made by creating a background of random noise and overwriting it with disks whose radii, centers, and constant image level were randomly chosen within some set bounds. The typical image level on the disks is fairly high compared with the noise in the surround. The radii of these disks are not necessarily integral.



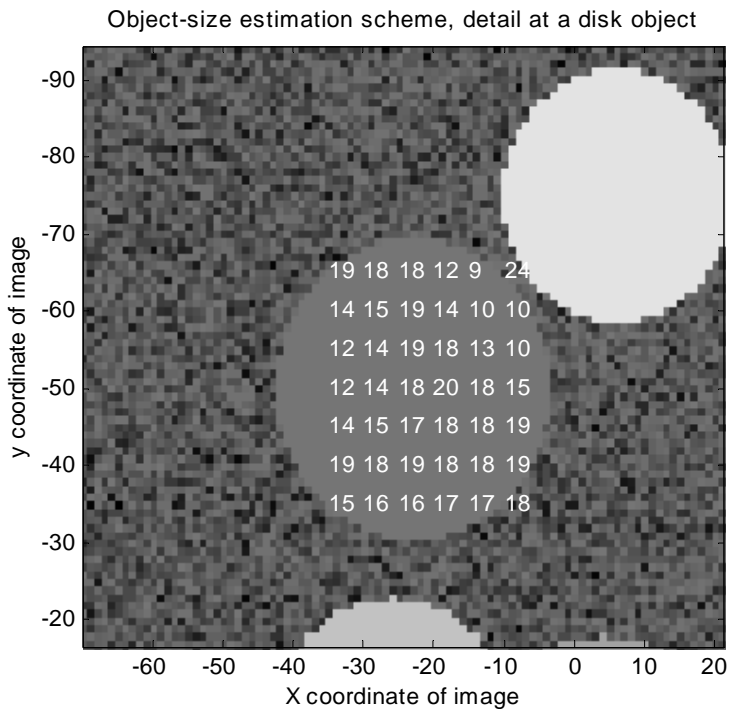
**Figure 7.** A test image for the object-size estimation scheme discussed in Section 6. It has a randomly selected choice of constant-level disks on a background of random noise. A tone curve has been applied to the original image to improve the legibility of the illustration. Figures 8 and 9 show expanded views of sections this image.

The object-size estimates  $\rho(q, g)$  computed using this method tend to correspond well with the radius of a disk-shaped object in the image  $g$  if  $q$  is the center of the disk, but the dependence on position is not impractically sensitive, i.e., the value of  $\rho(q, g)$  does not change very rapidly as  $q$  moves in a neighborhood of the center of the disk. Figure 8 and Figure 9 illustrate these points.

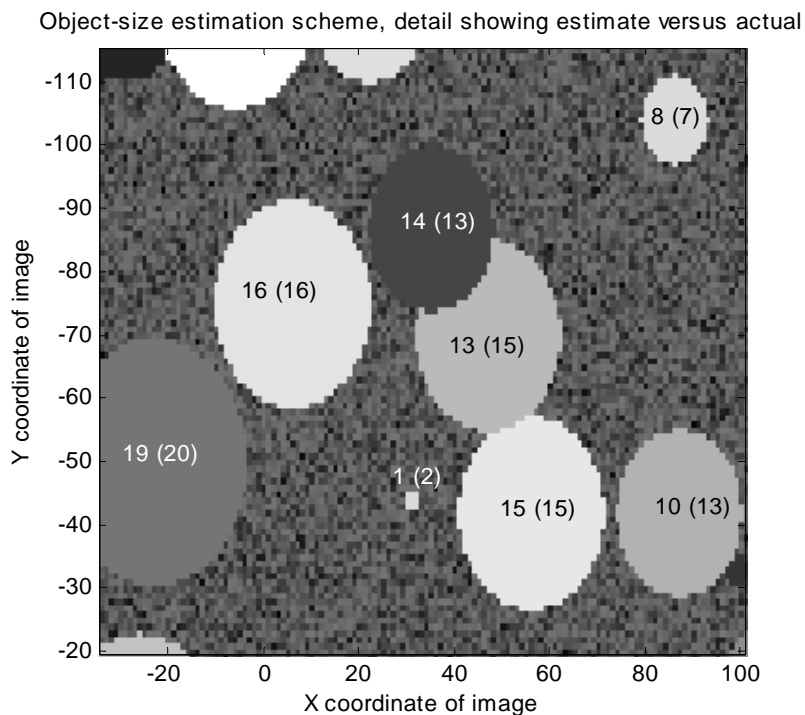
Experiment has shown that this size-estimation scheme also has application to objects that are not necessarily disk-shaped nor constant in image level, provided they are close to convex and not too elongated. For such an object centered at  $q$  in an image  $g$ ,  $\rho(q, g)$  has a good correspondence to the “equivalent radius” of the object, which is the radius of the disk-shaped cylinder that has the same volume of the object when regarded as bump in a three-dimensional graph of the image..

## 6. CONCLUSION

We have outlined some of our initial work on the application of novel analog image-processing devices to problems where their ability to perform massive, parallel convolution operations gives them an advantage over digital devices in terms of economy of time and power. Inspired by an idea in the literature<sup>10</sup> for application of the CWT, we constructed a filter bank composed of linear combinations of models of basic device-native filters and used it to estimate the size of objects in an image. The ideal wavelet used here was of the simplest form, as were our approximations. By using wavelets of more complex structure, and using improved techniques to find approximations, we hope to extend the procedure illustrated here to design filter banks applicable to the extraction of other, more descriptive, object features.



**Figure 8.** This shows an expanded view of a section of the image  $g$  of Figure 7, with values of the object size estimate  $\rho(q, g)$  written at some of the positions  $q$ . The actual radius of the disk in the center of the figure is about 20.



**Figure 9.** This shows an expanded (and distorted) view of a section of the image Figure 7. Each disk is labeled with the object-size estimate computed at its center and, in parentheses, the actual radius of the disk, rounded.

## 7. ACKNOWLEDGEMENTS

A substantial portion of the work discussed in this paper was supported by Naval Surface Warfare Center, Dahlgren Division, contracts N00178-02-C-3116 and N00178-02-C-3114.

## 8. REFERENCES

1. C. Mead, *Analog VLSI and Neural Systems*, Addison-Wesley, New York, 1989.
2. N. Graham, *Visual Pattern Analyzers*, Oxford University Press, Oxford, 1989.
3. L. O. Chua, L. Yang, "Cellular neural networks: theory and applications", *IEE Trans. on Circuits and Systems* **35**, pp. 1257-1290, 1988.
4. F. Werblin, T. Roska, L. O. Chua, "The analogic cellular neural network as a bionic eye", *Int. J. Circuit Theory and Applications* **23**, pp. 541-549, 1995.
5. L. O. Chua, T. Roska, *Cellular Neural Networks and Visual Computing*, Cambridge University Press, Cambridge, 2001.
6. J. McElvain, J. D. Langan, R. Behm, M. Costolo, A. J. Heeger, "Spatial frequency filtering using hybrid polymer/VLSI technology", *Proc. Smart Structures And Materials 1999: Electroactive Polymer Actuators and Devices (Proc. of SPIE, Vol. 3669)*, SPIE Press, 1999.
7. J. McElvain, "1-D and 2-D analysis of the thin film analog image processor (TAIP)", UNIAX Corporation, unpublished manuscript.
8. V. Kurkova, B. Beliczynski, "An incremental learning algorithm for Gaussian radial-basis-function approximation" (1995). <http://citeseer.nj.nec.com/kurkova95incremental.html>
9. D.B. Fogel, *Evolutionary Computation*, SPIE Press, Bellingham Washington, USA, 2000, p.71.
10. J.-P. Antoine, P. Vanderghenst, K. Bouyoucef, and R. Murenzi, "Target detection and recognition using two-dimensional continuous isotropic and anisotropic wavelets", *Proc. Automatic Object Recognition V*, pp. 20-31, SPIE Press, Orlando, Florida, 1995.