International Journal of Pure and Applied Mathematics

Volume 46 No. 2 2008, 199-209

Invited Lecture Delivered at
Forth International Conference of Applied Mathematics
and Computing (Plovdiv, Bulgaria, August 12–18, 2007)

TRANSFORMS, FILTERS AND EDGE DETECTORS IN IMAGE PROCESSING

John Schmeelk

Department of Mathematical Sciences
Virginia Commonwealth University
P.O. Box 8095, Doha, QATAR
e-mail: jschmeelk@qatar.vcu.edu

Abstract: This is an expansion of my previous presentation in our Third International Conference of Applied Mathematics and Computing in 2006 in Plovdiv. Again, filter design is an integral component to enhance desired aspects within an image such as clarity of edges. This paper provides an introduction to filter design and image edge detection using matrices, partial derivatives, convolutions and frequencies that enter into identifying components in the problem. We are especially addressing the notion of edge detection, which has far reaching applications in all areas of research, including medical research. For example a patient can be diagnosed as having an aneurysm by studying an angiogram. An angiogram is the visual view of the blood vessels whereby the edges are highlighted. This process is completed through convolution, filters and special frequency techniques using the software, *MATLAB 7.2* (2006). Some illustrations included will be vertical, horizontal and Sobel Edge Detectors together with some wavelet transforms to locate the edges in an image. We also include the Fast Fourier Transform to obtain the frequency space together with filtering such as the use of the Ideal Filter, Butterworth Filter and Bessel Function Filter.

AMS Subject Classification: 68U10
Key Words: edge detection in images, filter design, wavelet transforms

Received: August 17, 2007

1. Introduction

To motivate this paper, we provide an introduction to some interesting problems in image processing implementing matrix techniques, partial derivatives and convolutions. Section 2 provides an introduction to matrix and partial derivatives and how they are applied to the pixels to obtain the gray level value. Section 3 introduces a few specific examples such as the vertical, horizontal and Sobel edge detectors. Section 4 provides the reader with a series of illustrations that demonstrate edging techniques in three-dimensional image processing. Section 5 illustrates several filters together with their impact on the frequencies transferring the results to two images.

2. Some Notions and Notations

A current laptop in advertisements displays an image using 1680 x1050 pixels. The number of pixels continues to increase everyday as technology progresses. Therefore the images continue to become clearer as technology improves. Each pixel location designated by the coordinates, (x_1, y_1), contains a gray level value indicating the shade of gray within the image at that point. The values are usually on a scale of 0 to 255 whereby 0 corresponds to pure white and 255 correspond to black. The value of the gray level at this lattice point, (x_1, y_1), will be designated by f(x_1, y_1).

However before we continue with the edge detection analysis, we first review a few matrix and calculus techniques. We first recall the familiar dot product for two vectors, **x**, **y**, to be $\mathbf{x}\bullet\mathbf{y}=\sum\limits_{i=1}^{2} x_i y_i$. From this dot or inner product we define the norm to be $\|x\|^2 = \sum\limits_{i=1}^{2} x_i y_i$. Then we obtain the familiar and very important result to many applications that the cosine of the angle between the two vectors, **x** and **y**, satisfies the equation that $\cos(\theta)=x\bullet y/(\|x\|\,\|y\|)$. We know the maximum value for the cosine occurs when the two vectors coincide giving a value, cosine(0)=1. This is an important observation in edge detection and will latter be brought forward.

We now introduce the partial derivative formulas,

$$\frac{\partial f(x,y)}{\partial x} = \lim_{\Delta x \to 0} \frac{f(x+\Delta x, y) - f(x,y)}{\Delta x},$$

and

$$\frac{\partial f(x,y)}{\partial y} = \lim_{\Delta y \to 0} \frac{f(x,y+\Delta y) - f(x,y)}{\Delta y}.$$

The distance between pixel locations will be defined to be 1 so all of the increments in the partial derivative formulae will be equal to one. This then gives,

$$\frac{\partial f(x,y)}{\partial x} \approx \frac{f(x+1,y) - f(x,y)}{1},$$

and

$$\frac{\partial f(x,y)}{\partial y} \approx \frac{f(x,y+1) - f(x,y)}{1}.$$

We now denote the function, f(x, y), to be the gray level values between neighboring pixels in the horizontal and vertical directions respectively giving us the formulas, f(x_1+1, y_1) –f(x_1, y_1) and f(x_1, y_1+1) –f(x_1, y_1). The spatial locations, x_i and y_i can only take on integer values given by their integer locations.

3. Convolution and Edge Detectors

We first introduce the usual calculus definition for convolution given by the formula,

$$h(x,y) * f(x,y) = \int\limits_{-\infty}^{+\infty} \int\limits_{-\infty}^{+\infty} h(k_1,k_2) f(x-k_1, y-k_2) dk_1 dk_2,$$

and its discrete version by the formula,

$$h(n_1,n_2) * f(n_1,n_2) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} h(k_1,k_2) f(n_1-k_1, n_2-k_2).$$

We now reduce the discrete convolution to be a 3 by 3 matrix, which will play the role of a convolute and select our function, h(n_1, n_2), to have the matrix values,

$$\mathbf{h} = \begin{pmatrix} h(-1,1) & h(0,1) & h(1,1) \\ h(-1,0) & h(0,0) & h(1,0) \\ h(-1,-1) & h(0,-1) & h(1,-1) \end{pmatrix} = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}.$$

The arguments (n_1, n_2) in h(n_1, n_2) of the first array are easily remembered by noting that they are the needed lattice point coordinates referred to as a Cartesian coordinate system. This is illustrated in Figure 1. Clearly the reduced array for h(n_1, n_2) is part of the complete array, where h(n_1, n_2) is part of the complete array and is equal to zero whenever $|n_1|$ or $|n_2| > 2$. Next we convolve the function, h(n$_1$, n$_2$) with the function, f(n$_1$, n$_2$), and obtain

$$\begin{aligned} h(n_1, n_2) * f(n_1, n_2) &= \sum_{k_1=-1}^{1} \sum_{k_2=-1}^{1} h(k_1, k_2) f(n_1 - k_1, n_2 - k_2) \\ &= f(n_2 - 1, n_2 + 1) - f(n_1 + 1, n_2 + 1) + f(n_1 - 1, n_2) - f(n_1 + 1, n_2 + 1) \\ &\qquad + f(n_1 - 1, n_2 + 1) - f(n_1 + 1, n_2 - 1). \end{aligned}$$

We now investigate this last result only to find that it gives the difference of three columns of pixel values in the horizontal direction. If one checks the literature (see [6], [7]), we find that this is the approximation used in the horizontal direction in several leading software image-processing packages. The function, h(n$_1$, n$_2$), is called the kernel of the convolution and when we change its values, we obtain different edger's. The edge is the portion of the image, where there is a sudden change in gray levels. The edger implemented selects a particular feature in the image, which is beneficial to the particular application. The kernel for vertical edging is given by

$$\mathbf{h} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}.$$

A more sophisticated edger is the Sobel edger, which uses the gradient to approximate the edges. Since the gradient includes both horizontal and vertical components, two kernels are employed given by the matrices,

$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, \quad \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}.$$

4. Illustrations using Edge Detectors

Figure 2 and 8 illustrate the alphabets O and N respectively in three dimensions. We then employ a vertical edge detector on the alphabets O and N shown in Figures 3 and 9 respectively. Again horizontal edge detectors are applied and illustrated on O and N in Figures 4 and 10 respectively. The Sobel edge detector is then applied to O and N and illustrated in Figures 5 and 11. We include in Appendix A brief discussion of the two dimensional wavelet transform. We conclude the illustrations with a wavelet constructed using the Gaussian illustrated in Figure 6 together with its application on the letters O

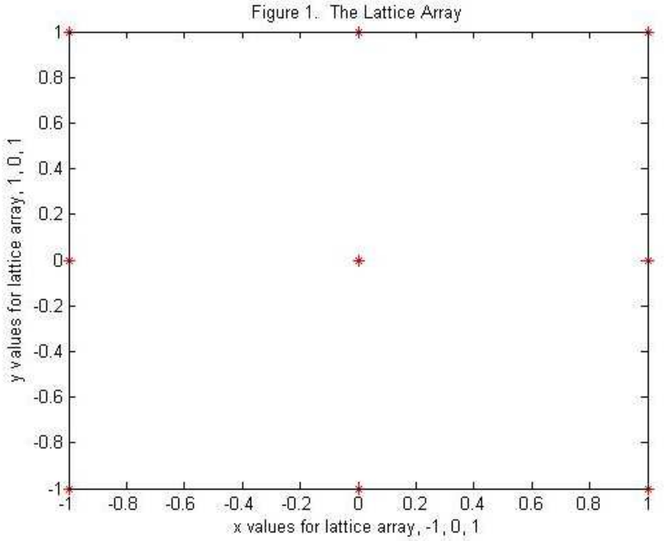

Figure 1. The Lattice Array

and N illustrated in Figures 7 and 12 respectively.

We conclude the previous applications with the realization that the type of image can yield results that are clearly visible for human sight. The Sobel edge detector on the letters O and N are clearly visible to our eyesight. However adjustments not readily apparent to human sight such as the wavelet transforms on the letter O and N can have far reaching consequences when compared to “normal” vs. “abnormal” physical phenomenon such as the aneurysm situation. We continue with filter designs.

5. Filter Designs

Filters designed and implemented in signal analysis are an integral component to enhance desired aspects of the application. In image processing filters are designed to select frequencies to add or subtract much-needed visual components of the image. For example the edges are significantly destroyed because of noise. The noise component contributes heavily to the high-frequency content of the Fourier transform of the image. To remove some of the unwanted noise can be achieved in a low pass filter passing the low frequencies. However we view an “ideal low pass filter” which has a sharp drop off of higher frequencies. A sharp drop off frequency can in many situations not remove the unwarranted noise. The drop off must be “gradual” thus making filter designs a mathematical “art

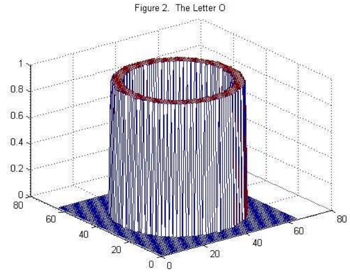

Figure 2

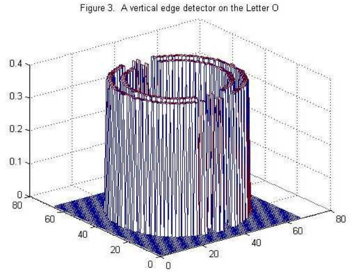

Figure 3

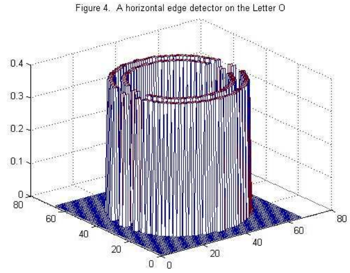

Figure 4

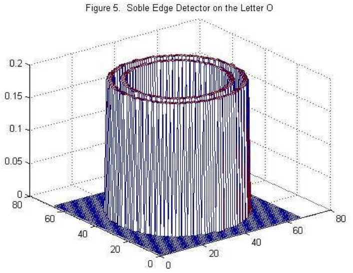

Figure 5

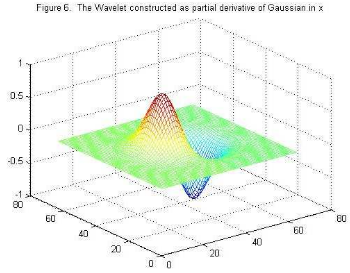

Figure 6

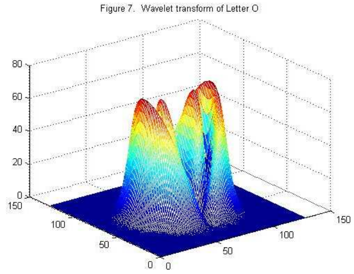

Figure 7

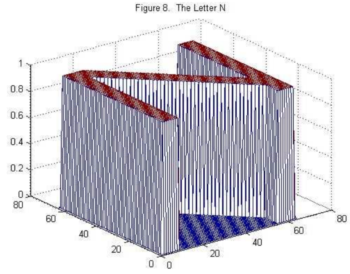

Figure 8

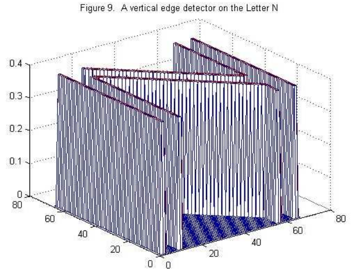

Figure 9

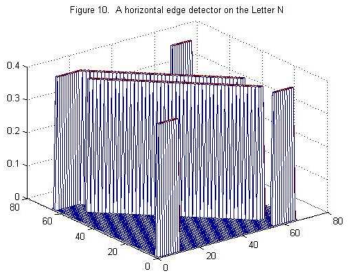

Figure 10

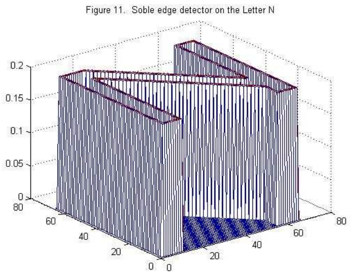

Figure 11

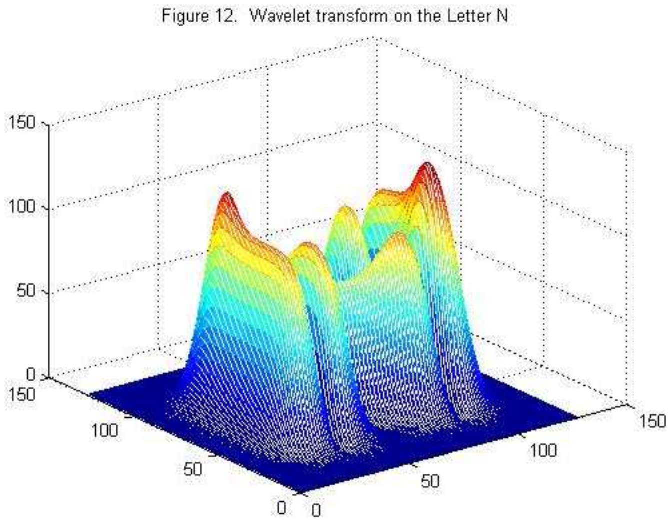

Figure 12

form". We illustrate a sharp drop off with a cylinder ideal low pass filter in Figure 13. An excellent low pass filter designed by Butterworth is illustrated in Figure 14 and a low pass Bessel function filter illustrated in Figure 15. When high frequencies are needed in applications we have developed the high pass Butterworth and Bessel function filters in Figures 23 and 24 respectively.

The filters are then applies to the Letter O and N.

The high pass Ideal, Butterworth and Bessel Function filters are illustrated in Figures 22, 23 and 24 respectively.

We now apply the high pass filters to the Letter O.

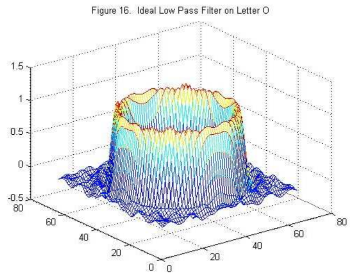

Figure 13

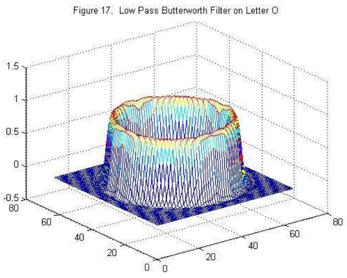

Figure 14

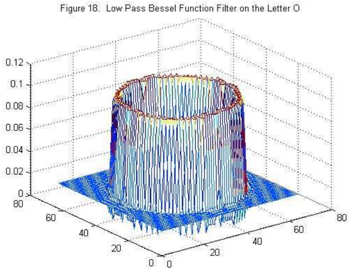

Figure 15

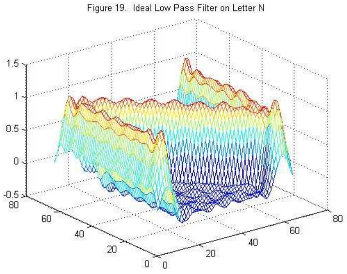

Figure 16

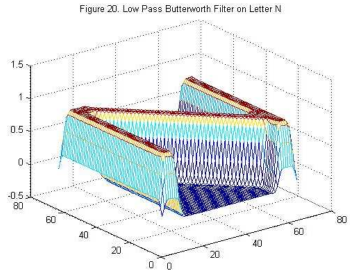

Figure 17

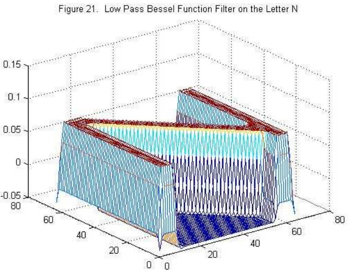

Figure 18

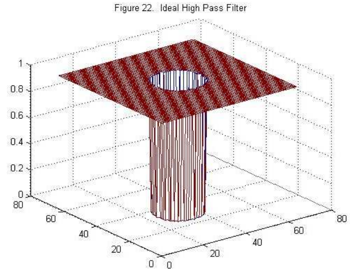

Figure 19

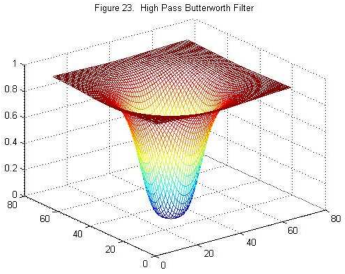

Figure 20

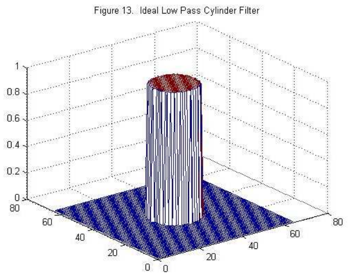

Figure 21

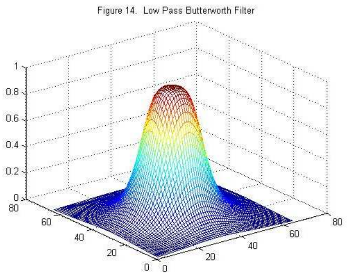

Figure 22

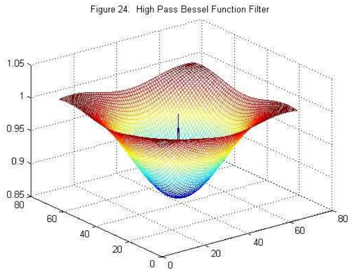

Figure 23

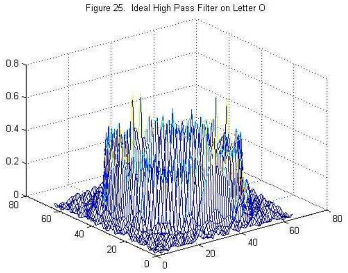

Figure 24

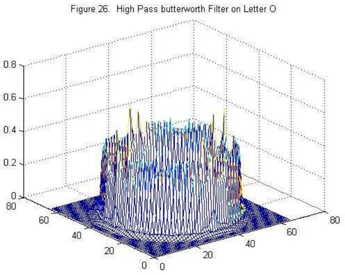

Figure 25

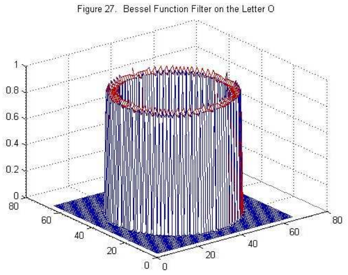

Figure 26

References

[1] H.C. Andrews, B.R. Hunt, *Digital Image Restoration*, Prentice Hall, N.J. (1977).

[2] D.H. Ballard, Parameter Nets, *Artificial Intelligence*, **22** (1984), 235-267.

[3] D.H. Ballard, C.M. Brown, *Computer Vision*, Prentice Hall, N.J. (1982).

[4] B.G. Batchelor, *Pattern Recognition*, Plenum Press, N.Y. (1978).

[5] F.W. Campbell, J.G. Robson, Application of Fourier analysis to the visibility of gratings, *J. Physiol.*, **197** (1968), 551-566.

[6] R.C. Gonzalez, P. Wintz, *Digital Image Processing*, Addison-Wesley Publ. Co., MA (1987).

[7] A.K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, NJ (1989).

[8] J.S. Lim, *Two-Dimensional Signal and Image Processing*, Prentice Hall, NJ (1990).

[9] G. Nagy, State of the art in pattern recognition, *Proc. IEEE*, **56** (1968), 836-862.

[10] W. Pedrycz, Fuzzy sets in pattern recognition; methodology and methods, *Pattern Recognition*, **20** No. 1-2 (1990), 121-146.

[11] W.K. Pratt, *Digital Image Processing*, John Wiley, Sons, NY (1991).

[12] R.J. Schalkoff, *Digital Image Processing and Computer Vision*, John Wiley, Sons, NY (1989).

[13] Y.Y. Tang, L.H. Yang, J. Liu, H. Ma, Wavelet theory and its application to pattern recognition, *World Scientific*, NJ (2000).

Appendix A. Two Dimensional Wavelet Transforms

We will not include a presentation regarding multiresolution analysis leading to a scaling function and then to a "mother" wavelet. The references [8, 13] are but a few resources for this remarkable analysis.

We begin with a two-dimensional mother wavelet, w(x, y), having dilation and translation parameters, (a_1, a_2) and (b_1, b_2) respectively each varying over R^2. The dilated and translated "mother" wavelet then becomes

$$w^{(a_1,a_2)(b_1,b_2)}(x,y) = \frac{1}{\sqrt{a_1 a_2}} w\left(\frac{x-b_1}{a_1}, \frac{y-b_2}{a_2}\right),$$

$a_1 \neq 0$ and $a_2 \neq 0$. The Fourier transform of this wavelet then becomes

$$\widehat{w}^{(a_1,a_2)(b_1,b_2)}(u,v) = \frac{2\pi}{\sqrt{a_1 a_2}} \int\limits_{-\infty}^{\infty} \int\limits_{-\infty}^{\infty} e^{-j\pi(ux+vy)} w\left(\frac{x-b_1}{a_1}, \frac{y-b_2}{a_2}\right)$$

$$= \frac{1}{\sqrt{a_1 a_2}} e^{-j\pi(ub_1+vb_2)} \widehat{w}(ua_1, va_2).$$

Furthermore Parsevals' formula in R^2 becomes

$$\int\limits_{-\infty}^{\infty}\int\limits_{-\infty}^{\infty} f(x,y)\overline{g(x,y)}dxdy = \frac{1}{4\pi^2}\int\limits_{-\infty}^{\infty}\int\limits_{-\infty}^{\infty} \hat{f}(u,v)\overline{\hat{g}(u,v)}dudv.$$

Definition A1.1. The two-dimensional wavelet transform on f(x, y) is then defined by the formula,

$$(w^{wav}f)\left((a_{,1},a_2),(b_1,b_2)\right) = \left(f, w^{(a_1,a_2)(b_1,b_2)}\right)$$

$$= \int\limits_{-\infty}^{\infty}\int\limits_{-\infty}^{\infty} \frac{1}{\sqrt{a_1}a_2} f(x,y)\overline{w\left(\frac{x-b_1}{a_1}, \frac{y-b_2}{a_2}\right)}dxdy.$$

The resolution of the identity an important inversion tool for the wavelet transform is given by the following theorem.

Theorem A1.2. *For all f, g $\in L^2\left(R^2\right)$ there holds*

$$\iiiint \frac{da_1da_2db_1db_2}{(a_1a_2)^2}\left\{(T^{wav}f)(a_1,a_2)(b_1,b_2)\right\}\left\{(T^{wav}g)(a_1,a_2)(b_1,b_2)\right\}$$

$$= C_{\varpi}(f,g).$$

Proof. See reference [24].

The C$_{\varpi}$ in Theorem A1.2 equals

$$C\varpi = \iint \frac{ds_1ds_2}{|s_1s_2|}\left|\omega(s_1,s_2\right|^2, \qquad (A1.3)$$

leading to the inversion formula,

$$f(x)$$

$$= C_{\varpi}^{-1}\iiint \frac{da_1da_2db_1db_2}{(a_1a_2)^2}\left\{(T^{wav}f)(a_1,a_2)(b_1,b_2)\right\}\left\{\omega^{(a_1,a_2)(b_1,b_2)}\right\}. \qquad \text{(A1.4)}$$

Expression (A1.4) requires the “mother” wavelet to satisfy the necessary condition,

$$\iint \omega(x,y)dxdy = 0\,. \qquad \square$$