A Simple Nonlinear Filter for Edge Detection in Images

Mladen Victor Wickerhauser

Department of Mathematics, Washington University, St. Louis, M0 63130 USA

and

Wojciech Czaja

Instytut Matematyczny, Uniwersytet Wrocławski, 50-384 Wrocław, Poland

ABSTRACT

We specialize to two simple cases the algorithm for singularity detection in images from eigenvalues of the dual local autocovariance matrix. The eigenvalue difference, or "edginess" at a point, then reduces to a simple nonlinear function. We discuss the derivation of these functions, which provide low-complexity nonlinear edge filters with parameters for customization, and obtain formulas in the two simplest special cases. We also provide an implementation and exhibit its output on six sample images.

Keywords: edge detection, nonlinear filtering

1. INTRODUCTION

In,[1] we built an edge detector for images using the localized two-dimensional Fourier transform. For theoretical analysis, we suppose that an image is a square-integrable function $I : \mathbf{R}^2 \to \mathbf{R}$. We localize with a smooth, radial, nonzero but compactly-supported bump function $g : \mathbf{R}^2 \to \mathbf{R}$, and define the "edginess" $e(I, z)$ of I at a point $z \in \mathbf{R}^2$ to be

$$\liminf_{\epsilon \to 0+} |\lambda_1(\epsilon, z) - \lambda_2(\epsilon, z)|,$$

where $\lambda_1(\epsilon, z)$ and $\lambda_2(\epsilon, z)$ are the eigenvalues of the *scale ϵ dual local autocovariance matrix of I at point z*, which we denote by $E_{\epsilon,g}(I; z)$. This is a 2×2 second moment matrix for the unnormalized density $\Phi \stackrel{\text{def}}{=} |\widehat{g_{\epsilon,z} I}|^2$ given by the squared magnitude of the Fourier transform of I multiplied by $g_{\epsilon,z}(x) \stackrel{\text{def}}{=} g(\frac{x-z}{\epsilon})$. The four coordinates of the matrix are

$$E_{\epsilon,g}(I; z)_{ij} = \int_{B(0,1/\epsilon)} \xi_i \xi_j \Phi(\xi)\, d\xi, \qquad i, j \in \{1, 2\}. \tag{1}$$

By the assumptions, $g_{\epsilon,z} I$ is integrable, so Φ is bounded and continuous and the matrix coefficients are well defined. The matrix is symmetric and positive semidefinite by construction, so its eigenvalues are real and nonnegative and we may speak of the larger one λ_1 and smaller one λ_2. Thus $e(I, z) = \lim_{\epsilon \to 0+} [\lambda_1(\epsilon, z) - \lambda_2(\epsilon, z)] \geq 0$.

Our earlier results may be combined into the following theorem:

THEOREM 1.1. *Let $D \subset \mathbf{R}^2$ be a bounded domain with a smooth boundary, and put $I = S + \mathbf{1}_D$, where $S : \mathbf{R}^2 \to \mathbf{R}$ is differentiable as well as locally square integrable. Then $z \in \partial D$ if and only if $e(I, z) > 0$.* The boundary smoothness assumption may be replaced by the weaker Morrey-Campanato regularity assumption[2] in generalized form.[3]

The first author was supported in part by NSF grant DMS-0072234.

2. DISCRETIZATION AND IMPLEMENTATION

In practice, an image I is supported on some bounded rectangle $[0,M]\times[0,N]\subset\mathbf{R}^2$, where M and N are positive integers, and has piecewise constant *intensity* on squares, or *pixels*, of the form $[m,m+1)\times[n,n+1)$, $0\le m<M$, $0\le n<N$. In this case, the scale parameter ϵ should indicate the number of pixels near a point z used to compute edginess there, and the radial bump function g should indicate how to weight the intensities of pixels according to their distance from z. The localized image $g_{\epsilon,z}I(x)$ is then completely described by the finite list of samples $f(m,n)\stackrel{\text{def}}{=}g_{\epsilon,z}I(x)$ indexed by $m\in\{0,1,\ldots,M-1\}$ and $n\in\{0,1,\ldots,N-1\}$. Each evaluation locus $x=(x_1,x_2)=(m,n)$ is called a *gridpoint*.

We may therefore calculate the autocovariance matrix with the separable, two-dimensional discrete Fourier transform, or 2D-DFT:

$$\hat f(p,q)\stackrel{\text{def}}{=}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\exp\left(-2\pi i\frac{pm}{M}\right)\exp\left(-2\pi i\frac{qn}{N}\right)f(m,n),\qquad (p,q)\in\mathbf{Z}^2. \tag{2}$$

This expression is M-periodic in p and N-periodic in q, so it suffices to compute it for $p\in B_M\stackrel{\text{def}}{=}\left[-\frac{M}{2},\frac{M}{2}\right]$ and $q\in B_N\stackrel{\text{def}}{=}\left[-\frac{N}{2},\frac{N}{2}\right]$. That will cost $O(MN\log M\log N)$ arithmetic operations, using the "fast" factored implementation of 2D-DFT, or 2D-FFT.

If $g_{\epsilon,z}I$ is supported in the smaller subrectangle $[M_0,M_1]\times[N_0,N_1]$ around z, with $0\le M_0<M_1\le M$ and $0\le N_0<N_1\le N$, then the 2D-DFT sum for f reduces to the smaller range $\{M_0,\ldots,M_1-1\}\times\{N_0,\ldots,N_1-1\}$. If $M_1-M_0\ll M$ and $N_1-N_0\ll N$, then instead of 2D-FFT we may rearrange the sum to avoid complex arithmetic and still obtain the same low computational cost.

For this other rearrangement, we begin by expanding the squared absolute value of $\hat f(p,q)$ for real-valued f as follows:

$$\begin{aligned}|\hat f(p,q)|^2 &= \sum_{m=M_0}^{M_1-1}\sum_{m'=M_0}^{M_1-1}\sum_{n=N_0}^{N_1-1}\sum_{n'=N_0}^{N_1-1}f(m,n)f(m',n')\times\\ &\qquad\times\exp\left(-2\pi i\frac{p(m-m')}{M}\right)\exp\left(-2\pi i\frac{q(n-n')}{N}\right),\end{aligned}$$

with $(p,q)\in B_M\times B_N$ giving a complete set of representative values. Now $|\hat f(p,q)|^2$ approximates $\left|\widehat{g_{\epsilon,z}I}(\xi)\right|^2$ evaluated at the discrete frequency $\xi=(\xi_1,\xi_2)=(p,q)$. Hence the moments of Equation 1 have the following discrete approximations:

$$E_{\epsilon,g}(I;z)_{ij}\approx E_{ij}\stackrel{\text{def}}{=}\sum_{p\in B_M}\sum_{q\in B_N}\xi_i\xi_j|\hat f(p,q)|^2,\qquad i,j\in\{1,2\}. \tag{3}$$

In particular, E_{11} has a factor $\xi_1\xi_1=\xi_1^2=p^2=p^2q^0$, while E_{22} has a factor $\xi_2^2=p^0q^2$, and $E_{12}=E_{21}$ has the factor $\xi_1\xi_2=p^1q^1$ inside the sum. Let r be the degree of the $p=\xi_1$ term and s the degree of the $q=\xi_2$ term in these expressions, so that we may speak of the (r,s) moment. Then E_{11} is the $(2,0)$ moment, E_{22} is the $(0,2)$ moment, and $E_{12}=E_{21}$ is the $(1,1)$ moment of $|\hat f(p,q)|^2$.

But the (r,s) moment of $|\hat f(p,q)|^2$ may be computed by a sum in the original pixel coordinates:

$$\begin{aligned}\sum_{p\in B_M}\sum_{q\in B_N}p^rq^s|\hat f(p,q)|^2 &= \sum_{p\in B_M}\sum_{q\in B_N}\sum_{m=M_0}^{M_1-1}\sum_{m'=M_0}^{M_1-1}\sum_{n=N_0}^{N_1-1}\sum_{n'=N_0}^{N_1-1}p^rq^sf(m)f(m')f(n)f(n')\times\\ &\qquad\times\exp\left(-2\pi i\frac{p(m-m')}{M}\right)\exp\left(-2\pi i\frac{q(n-n')}{N}\right)\end{aligned}$$

$$
\begin{aligned}
&= \sum_{m=M_0}^{M_1-1} \sum_{m'=M_0}^{M_1-1} \sum_{n=N_0}^{N_1-1} \sum_{n'=N_0}^{N_1-1} f(m)f(m')f(n)f(n') \times \\
&\qquad \times \left[\sum_{p \in B_M} p^r \exp\left(-2\pi i \frac{p(m-m')}{M}\right)\right] \left[\sum_{q \in B_N} q^s \exp\left(-2\pi i \frac{q(n-n')}{N}\right)\right] \\
&\approx \sum_{m=M_0}^{M_1-1} \sum_{m'=M_0}^{M_1-1} \sum_{n=N_0}^{N_1-1} \sum_{n'=N_0}^{N_1-1} f(m)f(m')f(n)f(n')\mu_r(m-m')\mu_s(n-n').
\end{aligned}
$$

As in,[1] we have replaced the sums in square brackets with the Riemann integrals they approximate:

$$
\mu_j(k) \stackrel{\text{def}}{=} \int_{-\frac{1}{2}}^{\frac{1}{2}} x^j \exp(-2\pi i k x)\, dx, \qquad j = 0, 1, 2, \ldots; \quad k \in \mathbf{Z}, \tag{4}
$$

These are needed with $k \leftarrow n - n'$ or $k \leftarrow m - m'$ for $j = 0, 1, 2$.

Evidently $\mu_0(k) = 1$ if $k = 0$, but is zero otherwise. Hence, the unnormalized coefficients of the dual local autocovariance matrix may be computed from the following sums:

$$
\begin{aligned}
E_{11} &= \sum_{m=M_0}^{M_1-1} \sum_{m'=M_0}^{M_1-1} \sum_{n=N_0}^{N_1-1} \sum_{n'=N_0}^{N_1-1} f(m,n)f(m',n')\mu_2(m-m')\mu_0(n-n') \qquad (5) \\
&= \sum_{m=M_0}^{M_1-1} \sum_{m'=M_0}^{M_1-1} \sum_{n=N_0}^{N_1-1} f(m,n)f(m',n)\mu_2(m-m'); \\
E_{22} &= \sum_{m=M_0}^{M_1-1} \sum_{n=N_0}^{N_1-1} \sum_{n'=N_0}^{N_1-1} f(m,n)f(m,n')\mu_2(n-n'); \qquad (6) \\
E_{12} = E_{21} &= \sum_{m=M_0}^{M_1-1} \sum_{m'=M_0}^{M_1-1} \sum_{n=N_0}^{N_1-1} \sum_{n'=N_0}^{N_1-1} f(m,n)f(m',n')\mu_1(m-m')\mu_1(n-n'). \qquad (7)
\end{aligned}
$$

An elementary calculation gives us formulas for the other two moment functions that we need:

$$
\mu_1(k) = \begin{cases} 0, & \text{if } k = 0, \\ i\frac{(-1)^k}{2\pi k}, & \text{otherwise;} \end{cases} \qquad \mu_2(k) = \begin{cases} 1/12, & \text{if } k = 0, \\ \frac{(-1)^k}{2\pi^2 k^2}, & \text{otherwise.} \end{cases} \tag{8}
$$

3. ALGORITHM

Since translations of f have no effect on $|\hat{f}|^2$, we may reindex to have $M_0 = N_0 = 0$ in Equations 5, 6, and 7. Namely, we make the simultaneous index substitutions $m \leftarrow m + M_0$, $m' \leftarrow m' + M_0$, $n \leftarrow n + N_0$ and $n' \leftarrow n' + N_0$, which have no effect on $m - m'$ and $n - n'$ and result in the localized portion of the image being translated to the subgrid $\{(m, n) : 0 \leq m < M_1;\ 0 \leq n < N_1\}$. This simplifies implementation by making the summation formula independent of the point z after f is defined.

The edginess computation is performed by the following steps:

1. **Localization.** Extract the pixel values on the square subgrid contained in $z + [-\epsilon, \epsilon] \times [-\epsilon, \epsilon]$, where a small positive integer plays the role of ϵ in $g_{\epsilon,z}$. The pixel value at a gridpoint $z + y$ is multiplied by the radial bump function $g(y)$ and stored at the index corresponding to y in the subgrid. If z is within ϵ grid points of the boundary, then we simply pad any missing samples in the subgrid with zeros.

   Note that z need not be a gridpoint with integer coordinates. In particular, z may have half-integer coordinates, but then y will have half-integer coordinates as well and the extracted subgrid indices will be "corrected" from y. We may assume that the subgrid indices are $\{(m,n) : 0 \le m < M_1,\ 0 \le n < N_1\}$ as previously discussed.

This step costs $O(\epsilon^2)$ operations per pixel.

2. Dual Autocovariance. Compute the 2×2 matrix $E = (E_{ij})$ using Equations 5, 6, and 7. This always yields a real-valued, symmetric, positive semidefinite matrix. The computational complexity of the quadruple sum in Equation 7 dominates the triple sums of Equations 5 and 6, so this step costs $O(\epsilon^4)$ operations per pixel.

3. Eigenvalues. For symmetric 2×2 matrices E, the exact formula for eigenvalues is:

$$\lambda = \frac{1}{2}\left(E_{11} + E_{22} \pm \sqrt{(E_{11} - E_{22})^2 + 4E_{12}^2}\right), \tag{9}$$

where we take $+$ for λ_1 and $-$ for λ_2. These will satisfy $\lambda_1 \ge \lambda_2 \ge 0$

4. Edginess. We define this to be the difference $\lambda_1 - \lambda_2$. This reduces to a simple expression in terms of the matrix coefficients Eij:

$$\lambda_1 - \lambda_2 = \sqrt{(E_{11} - E_{22})^2 + 4E_{12}E_{21}} \tag{10}$$

Alternatively, we may normalize to get an intensity-independent and bounded quantity such as $(\lambda_1 - \lambda_2)/\lambda_1 \in [0, 1]$. Also, the value may be transformed in other ways to facilitate display. For example, we may use the bounded reciprocal λ_2/λ_1, which is related to the normalized difference by $\frac{\lambda_2}{\lambda_1} = 1 - \frac{\lambda_1 - \lambda_2}{\lambda_1}$. This may amplified to fill the grayscale range of a write-black display device. That way, the darkest marks indicate the greatest edginess.

The dual autocovariance step dominates the total computational complexity. It is therefore $O(\epsilon^4)$ operations per pixel, if we localize to subgrids of $O(\epsilon \times \epsilon)$ points. The edginess calculation regardless has linear complexity in the number of pixels. The $O(\epsilon^4)$ proportionality constant is controlled by choosing a small integer ϵ such as 2 or 3.

4. TWO SPECIAL CASES

As before, assume that the image is represented as a list of pixel values $f(m, n)$ indexed by $M \times N$ integer pairs $\{(m, n) : 0 \le m < M, 0 \le n < N\}$.

No edge information can be gleaned from a single pixel, so the smallest useful ϵ in the edginess algorithm is $\epsilon = 1$. We will consider the corresponding special cases of the edginess formula and compute them explicitly.

The radial bump function $g_{\epsilon,z}$ centered at z will weight all equidistant pixels equally, so the set of weights will be least complicated if z is most centered among the gridpoints. We achieve this centering in two ways: at midpoints and at gridpoints. In both centerings, edginess will be computed at $M \times N$ points. Zero padding will be required at some or all of the boundaries.

Midpoints. z is of the form $(m + \frac{1}{2}, n + \frac{1}{2})$. With $\epsilon = 1$, the dual autocovariance matrix is computed from the pixels immediately NE, SE, SW, and NW of z, with indices $(m+1, n+1)$, $(m+1, n)$, (m, n) and $(m, n+1)$, respectively:

$$\begin{matrix} (m, n+1) & & (m+1, n+1) \\ & z & \\ (m, n) & & (m+1, n) \end{matrix} \quad \longleftrightarrow \quad \begin{matrix} NW \overset{\text{def}}{=} f(m, n+1) & & NE \overset{\text{def}}{=} f(m+1, n+1) \\ & [z] & \\ SW \overset{\text{def}}{=} f(m, n) & & SE \overset{\text{def}}{=} f(m+1, n) \end{matrix}$$

By using the direction labels SW, SE, NW, NE to represent the pixel values, we hope to make the forthcoming calculations less cumbersome.

All pixel values receive the same weight, as all pixels are equidistant from z, so we may assume the weight from g is 1. We evaluate $\mu_1(0) = 0$, $\mu_1(\pm 1) = \frac{\pm 1}{2\pi i}$, $\mu_2(0) = \frac{1}{12}$,and $\mu_2(\pm 1) = \frac{-1}{2\pi^2}$ from Equation 8. Equations 5, 6, and 7 then give:

$$\begin{aligned}
E_{11} &= (NW^2 + NE^2)\mu_2(0-0) + (SW^2 + +SE^2)\mu_2(1-1) + \\
&\qquad +(NW \cdot SW + NE \cdot SE)\left(\mu_2(0-1) + \mu_2(1-0)\right) \\
&= \frac{1}{12}\left(NW^2 + NE^2 + SW^2 + +SE^2\right) - \frac{1}{\pi^2}\left(NW \cdot SW + NE \cdot SE\right); \\
E_{22} &= \frac{1}{12}\left(NW^2 + NE^2 + SW^2 + +SE^2\right) - \frac{1}{\pi^2}\left(NW \cdot NE + SW \cdot SE\right); \\
E_{12} = E_{21} &= -\frac{1}{2\pi^2}\left(NW \cdot SE - NE \cdot SW\right).
\end{aligned}$$

Only the first calculation is shown. But then

$$E_{11} - E_{22} = \frac{1}{\pi^2}\left(NW \cdot NE + SW \cdot SE - NW \cdot SW - NE \cdot SE\right) = \frac{1}{\pi^2}\left(NW - SE\right)\left(NE - SW\right).$$

Substituting these expressions into Equation 10 yields the following formula:

$$\lambda_1 - \lambda_2 = \frac{1}{\pi^2}\sqrt{\left(NW - SE\right)^2\left(NE - SW\right)^2 + \left(NW \cdot SE - NE \cdot SW\right)^2}. \tag{11}$$

This expression defines a nonlinear four-tap filter with a square 2×2 pixel mask.

Gridpoints. z is of the form (m, n). With $\epsilon = 1$, the dual autocovariance matrix is computed from the pixel at $z = (m, n)$, which has one weight, and the pixels immediately N, E, S, and W of z, with indices $(m, n+1)$, $(m+1, n)$, $(m, n-1)$ and $(m-1, n)$ respectively, which all have the same second weight:

$$\begin{array}{ccc}
 & (m, n+1) & \\
(m-1, n) & z = (m, n) & (m+1, n) \\
 & (m, n-1) &
\end{array}
\quad \longleftrightarrow \quad
\begin{array}{ccc}
 & N \stackrel{\text{def}}{=} f(m, n+1) & \\
W \stackrel{\text{def}}{=} f(m-1, n) & C \stackrel{\text{def}}{=} f(m, n) & E \stackrel{\text{def}}{=} f(m+1, n) \\
 & S \stackrel{\text{def}}{=} f(m, n-1) &
\end{array}$$

We may normalize g so that the weight at z is 1 and let $t > 0$ be the weight for the four neighboring pixels. As before, we use the direction labels S, E, N, W to represent the pixel values so as to simplify the notation a bit. We evaluate $\mu_1(0) = 0$, $\mu_1(\pm 1) = \frac{\pm 1}{2\pi i}$, $\mu_1(\pm 2) = \frac{\pm i}{4\pi}$, $\mu_2(0) = \frac{1}{12}$, $\mu_2(\pm 1) = \frac{-1}{2\pi^2}$, and $\mu_2(\pm 2) = \frac{1}{8\pi^2}$ from Equation 8, and substitute everything into Equations 5, 6, and 7:

$$\begin{aligned}
E_{11} &= \frac{1}{12}(C^2 + t^2[N^2 + E^2 + S^2 + W^2]) - \frac{t}{\pi^2}(N \cdot C + C \cdot S) + \frac{t^2}{4\pi^2}\left(N \cdot S\right); \\
E_{22} &= \frac{1}{12}(C^2 + t^2[N^2 + E^2 + S^2 + W^2]) - \frac{t}{\pi^2}(E \cdot C + C \cdot W) + \frac{t^2}{4\pi^2}\left(E \cdot W\right); \\
E_{12} = E_{21} &= \frac{t^2}{2\pi^2}\left(N \cdot W - N \cdot E + S \cdot E - S \cdot W\right) \quad = \quad \frac{t^2}{2\pi^2}\left(N - S\right)\left(E - W\right).
\end{aligned}$$

But then

$$E_{11} - E_{22} = \frac{t}{\pi^2}C\left(E + W - N - S\right) + \frac{t^2}{4\pi^2}\left(N \cdot S - E \cdot W\right).$$

Substituting these expressions into Equation 10 yields the following formula:

$$\lambda_1 - \lambda_2 = \frac{t}{\pi^2}\sqrt{\left[C\left(E + W - N - S\right) + \frac{t}{4}\left(N \cdot S - E \cdot W\right)\right]^2 + t^2\left(N - S\right)^2\left(E - W\right)^2}. \tag{12}$$

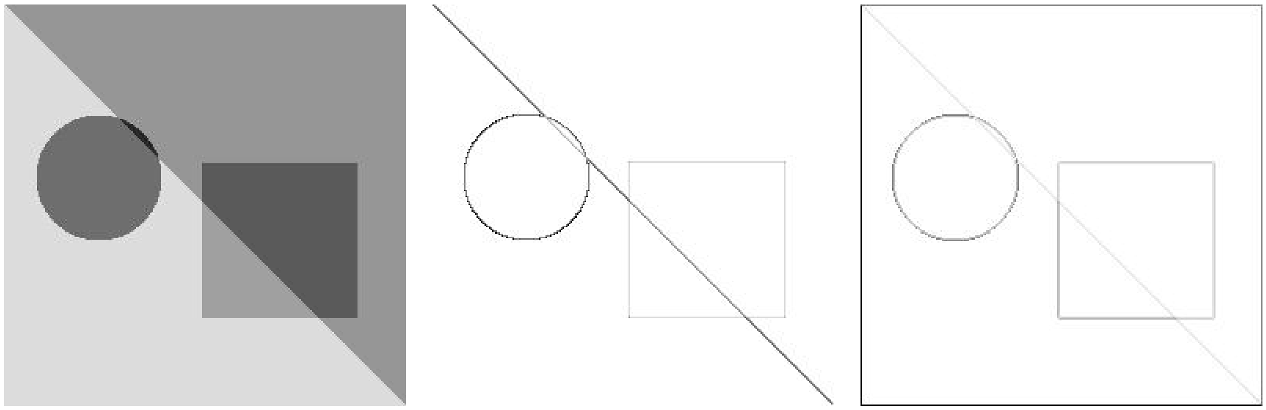

Figure 1: Figure: image, midpoint edginess, and gridpoint edginess.

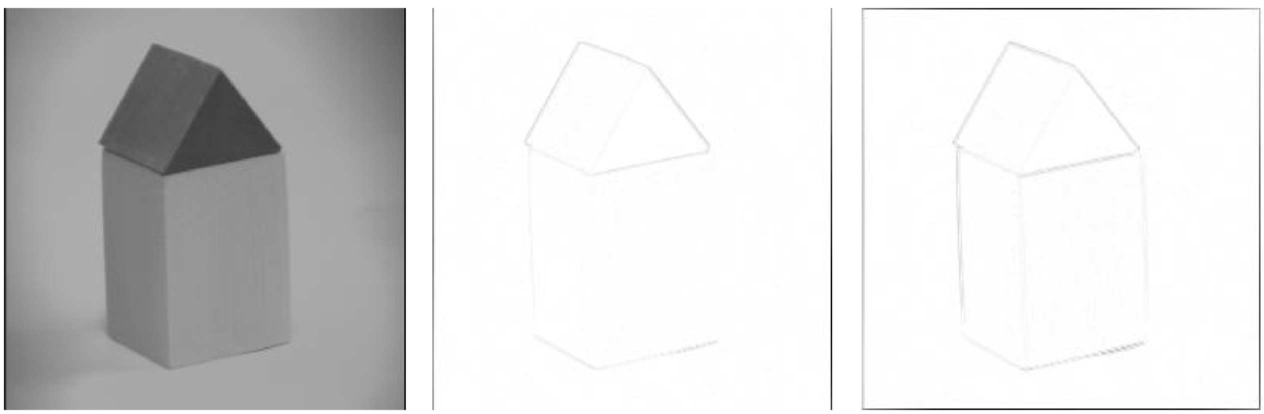

Figure 2: House: image, midpoint edginess, and gridpoint edginess.

This expression defines a nonlinear five-tap filter with a mask shaped like a plus sign. We obtain a different filter for each weight t. Notice that the choice $t = 0$, which suppresses all but the center pixel value C, results in zero edginess everywhere as expected.

A value $t \in (0, 1)$ such as $t = \frac{3}{4}$ may be chosen to tune the edginess function to a particular class of images. We note that as ϵ increases, the number of different weights in this algorithm will increase as well, giving extra degrees of freedom for customization.

5. EXAMPLE IMAGES

We prepared six examples by the two algorithms described above, localizing with Gaussian bumps restricted to 7×7 subgrids centered at z.

"Figures" are piecewise constant functions with jump discontinuities along various rectifiable, mostly smooth curves. "House" is the demonstration image `house_gray.pgm` from the XHoughTool website, `http://www.lut.fi/dep/tite/XHoughtool`. "Fingerprint" was obtained from NIST; it is part of a compliance test suite for the FBI's WSQ compression standard. "Lena" is the famous image from.[4] "Cone" is a synthetic ray-traced image provided by Craig Kolb. "Truck," provided by Peng Li, is one frame from a video. All six are available from the second author's web site, `http://www.math.wustl.edu/~victor`.

The gridpoint edginess figures all used the weight $t = \frac{3}{4}$.

Figure 3: Fingerprint: image, midpoint edginess, and gridpoint edginess.

Figure 4: Lena: image, midpoint edginess, and gridpoint edginess.

Figure 5: Cone: image, midpoint edginess, and gridpoint edginess.

Figure 6: Truck: image, midpoint edginess, and gridpoint edginess.

6. CONCLUSION

The localized two-dimensional Fourier transform, used to find edge-like singularities, leads to nonlinear filters for edge detection. These were implemented as transformations of a grayscale image into an edginess function of the same domain. We showed that these filter transformations have low computational complexity: they are linear in the number of pixels, with a small constant controlled by a scale parameter of the algorithm. As the scale parameter grows, the edge filter acquires addtional degrees of freedom that may be adjusted to suit different classes of images. We assert that this method permits the construction of new custom nonlinear edge detectors.

We illustrated the action of the simplest of these filter transformations on six grayscale images. The implementation in source code form, together with the images, is available from the authors' web site.

REFERENCES

1. W. K. Czaja and M. V. Wickerhauser, "Singularity detection in images using dual local autocovariance," *Applied and Computational Harmonic Analysis* **13**, pp. 77–88, July 2002.
2. S. Janson, M. Taibleson, and G. Weiss, "Elementary characterizations of the Morrey-Campanato spaces," in *Proceedings of the July, 1982 Harmonic Analysis Conference Held in Cortona, Italy*, G. Weiss, G. Mauceri, and F. Ricci, eds., *Lecture Notes in Mathematics* **992**, pp. 101–114, Springer-Verlag, (New York), 1983.
3. W. K. Czaja, *Applications of Local Autocovariance Matrices.* PhD thesis, Washington University, Saint Louis, Missouri, 2000.
4. L. Sjooblum, "Miss November." Playboy Magazine, Playboy Enterprises, Inc., 1972.