Edge Orientation Using Contour Stencils

Pascal Getreuer [(1)]

(1) Department of Mathematics, University of California Los Angeles
getreuer@math.ucla.edu

Abstract:

Many image processing applications require estimating the orientation of the image edges. This estimation is often done with a finite difference approximation of the orthogonal gradient. As an alternative, we apply contour stencils, a method for detecting contours from total variation along curves, and show it more robustly estimates the edge orientations than several finite difference approximations. Contour stencils are demonstrated in image enhancement and zooming applications.

1. Introduction

A fundamental and challenging problem in image processing is estimating edge orientations. Accurate edge orientations are important for example in edge-oriented inpainting methods [2], and optical character recognition features [8].

1.1 $\nabla \mathbf{u}^\perp$ for Estimating Edge Orientation

A starting point to edge orientation estimation is to approximate $\nabla u^\perp$ with finite differences. Finite difference estimation alone is typically too noisy to be reliable, especially near edges, so the gradient is often regularized by a convolution $\nabla u \approx \nabla(G * u)$ where G is for example a Gaussian. However, there is a serious problem in that $\nabla u^\perp$ and $-\nabla u^\perp$ both describe the same edge orientation, so linear smoothing tends to *cancel* the desired edge information.

Introduced by Bigün and Granlund [1] and Forstner and Gulch [3], a better approach is to use the 2×2 *structure tensor* $J(\nabla u) = \nabla u \otimes \nabla u$. The structure tensor satisfies $J(-\nabla u) = J(\nabla u)$ and ∇u is an eigenvector of $J(\nabla u)$. The structure tensor takes into account the orientation but not the sign of the direction, thus solving the antipodal cancellation problem.

As developed by Weickert [9], let

$$J_\rho(\nabla u_\sigma) = G_\rho * J(G_\sigma * u) \qquad (1)$$

where G_σ and G_ρ are Gaussians with standard deviations σ and ρ. The eigenvector of $J_\rho(\nabla u_\sigma)$ associated with the smaller eigenvalue is called the *coherence direction*, and is an effective approximation of edge orientation.

2. Contour Stencils

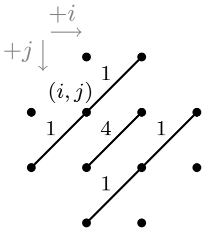
Numerical implementation of $J(\nabla u)$ yet involves estimating ∇u. Since numerical estimates of ∇u are sensitive to noise and unreliable near edges, significant amounts of smoothing is still needed for acceptable results. We abandon $\nabla u^\perp$ and approach the estimation of edge orientation from an entirely different principle.

Given a smooth curve C and a parameterization $\gamma : [0, T] \to C$, consider measuring the total variation of u along C,

$$\mathrm{TV}(C) = \int_0^T \big|\partial_t u\big(\gamma(t)\big)\big|\, dt. \qquad (2)$$

Edge orientations can be estimated by comparing $\mathrm{TV}(C)$ with various candidate curves. *Contour stencils* [4, 5] is a numerical implementation of this idea.

Let $u : \Lambda \to \mathbb{R}$ be a discrete image. Denote by $u_{i,j}$, $(i, j) \in \Lambda$, the value of u at the (i, j)th pixel, and let $x_{i,j} \in \mathbb{R}^2$ denote its spatial location.



$$\mathcal{S}(\alpha,\beta) = \begin{cases} 1 & \alpha = (i,j), \beta = (i-1,j+1), \\ 1 & \alpha = (i,j), \beta = (i+1,j-1), \\ 4 & \alpha = (i,j+1), \beta = (i+1,j), \\ 1 & \alpha = (i+1,j+1), \beta = (i,j+2), \\ 1 & \alpha = (i+1,j+1), \beta = (i+2,j), \\ 0 & \text{otherwise} \end{cases}$$

Figure 1: An example contour stencil $\mathcal{S}$ for detecting a $45°$ orientation.

A *contour stencil* is a function $\mathcal{S} : \Lambda \times \Lambda \to \mathbb{R}^+$ describing weighted edges between pixels (see Figure 1). These edges approximate several parallel curves localized over a small neighborhood. As a discretization of (2), the total variation of $\mathcal{S}$ is

$$\mathrm{TV}(\mathcal{S}) := \tfrac{1}{|\mathcal{S}|} \sum_{\alpha,\beta\in\Lambda} \mathcal{S}(\alpha,\beta)\,|u_\alpha - u_\beta|, \qquad (3)$$

and $|\mathcal{S}| := \sum_{\alpha,\beta} \mathcal{S}(\alpha,\beta)\,|x_\alpha - x_\beta|$. For the contour stencil in Figure 1, $|\mathcal{S}| = (1+1+4+1+1)\sqrt{2}$ and

$$\begin{aligned} \mathrm{TV}(\mathcal{S}) = \tfrac{1}{|\mathcal{S}|}\big(&|u_{i,j} - u_{i-1,j+1}| + |u_{i,j} - u_{i+1,j-1}| \\ &+ 4\,|u_{i,j+1} - u_{i+1,j}| \\ &+ |u_{i+1,j+1} - u_{i,j+2}| + |u_{i+1,j+1} - u_{i+2,j}|\big). \end{aligned}$$

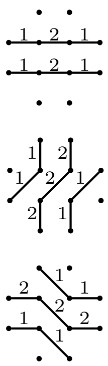


Figure 2: The proposed cell-centered contour stencils.

The contours of u are estimated by finding a stencil with low total variation,

$$\mathcal{S}^* = \arg\min_{\mathcal{S}\in\Sigma} \mathrm{TV}(\mathcal{S}) \qquad (4)$$

where Σ is a set of candidate stencils (see Figures 2 and 3). The best-fitting stencil $\mathcal{S}^*$ provides a model of the underlying contours.

In summary, contour stencil orientation estimation is done by first computing the TV estimates (3) for each candidate stencil, and then determining the best-fitting stencil $\mathcal{S}^*$. For efficient implementation, define

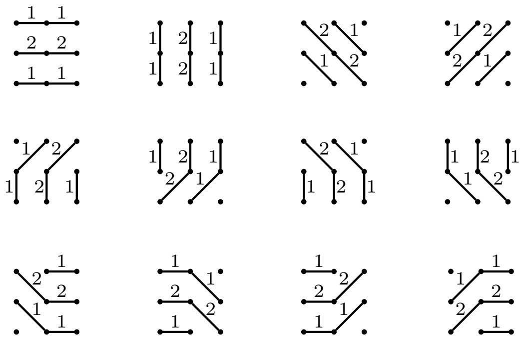


Figure 3: A node-centered stencil set.

$$D^H_{i,j} = |v_{i,j} - v_{i+1,j}|, \qquad D^A_{i,j} = |v_{i,j} - v_{i+1,j+1}|,$$
$$D^V_{i,j} = |v_{i,j} - v_{i,j+1}|, \qquad D^B_{i,j} = |v_{i,j+1} - v_{i+1,j}|,$$

then the $\mathrm{TV}(\mathcal{S})$ can be computed as sums of these differences, and the differences may be reused between successive cells. For the proposed stencil sets, contour stencils cost a few dozen operations per pixel [4].

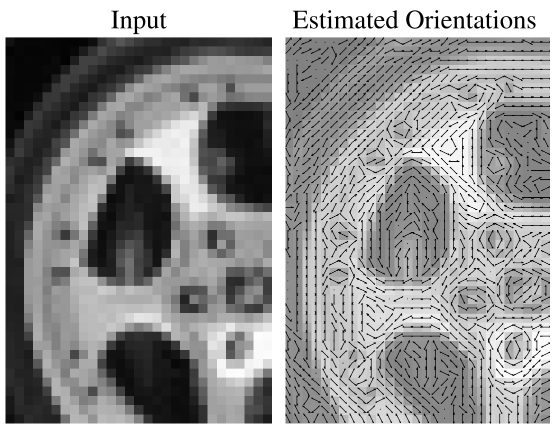


Figure 4: Edge orientation estimation with contour stencils (using the cell-centered stencils in Figure 2).

Contour stencils extend naturally to nonscalar data by replacing the absolute value in (3) with a metric. On color images for example, a suitable choice is the ℓ^1 vector norm in $\mathrm{YC_bC_r}$ color space.

3. Comparison

Here we compare contour stencils and several finite difference methods for estimating edge orientation.

As a test image with fine orientations, we use a small image of straw (Figure 5).

u

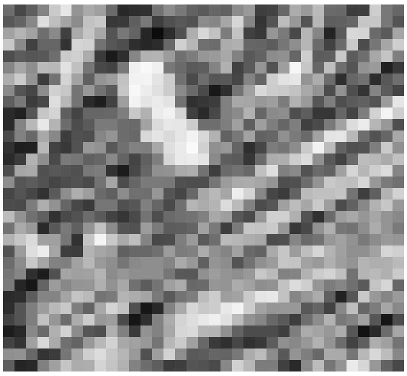
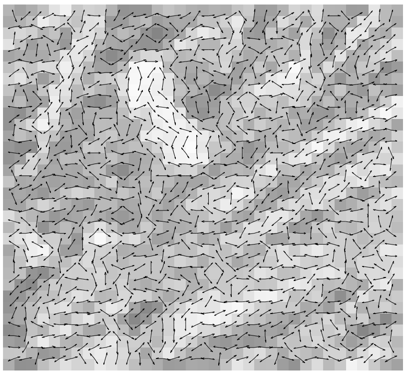


Figure 5: The test image.

As is done with coherence direction (1), any orientation field $\vec{\theta}$ can be smoothed by filtering its tensor product: $G_\rho * (\vec{\theta} \times \vec{\theta})$. But for easier comparison, all methods are shown *without* smoothing.

$\nabla u^\perp$ with (5)



Contour Stencils (Σ as in Figure 2)

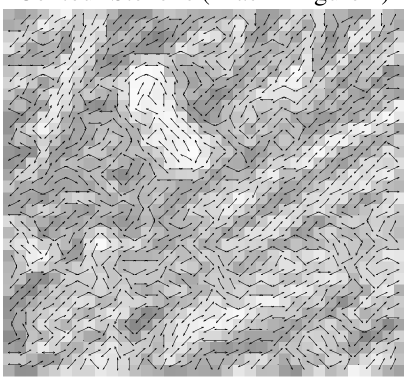


Figure 6: Comparison of cell-centered methods.

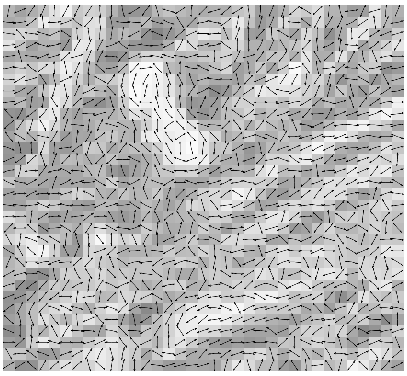
We consider two categories of methods: cell-centered and node-centered. Define the (i, j)th cell as the square whose corners correspond to $u_{i,j}$, $u_{i+1,j}$, $u_{i,j+1}$, $u_{i+1,j+1}$. Cell-centered methods compute orientation estimates logically located in the center of the cells. With node-centered methods, the edge orientation estimates are centered on the pixels.

Let D_x^+ denote the forward difference operator $D_x^+ u_{i,j} = u_{i+1,j} - u_{i,j}$ and similarly in the other coordinate D_y^+. An estimate of ∇u symmetric over the cell is

$$\nabla u_{i,j} \approx \begin{pmatrix} (D_x^+ u_{i,j} + D_x^+ u_{i,j+1})/2 \\ (D_y^+ u_{i,j} + D_y^+ u_{i+1,j})/2 \end{pmatrix}. \tag{5}$$

Figure 6 compares $\nabla u^\perp$ estimated using (5) with contour stencils using the cell-centered stencil set shown in Figure 2.

Sobel filter (6)



Contour Stencils (Σ as in Figure 3)

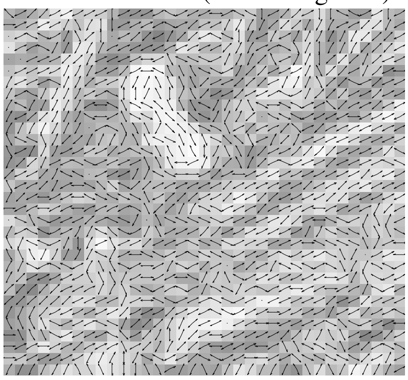


Figure 7: Comparison of node-centered methods.

The Sobel filter [7] is a node-centered approximation of ∇u,

$$\partial_x u \approx \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * u \tag{6}$$

and similarly for $\partial_y u$. Figure 7 compares the Sobel filter with contour stencils using the node-centered stencil set from Figure 3.

4. Applications

Contour stencils are useful in applications where edges are significant.

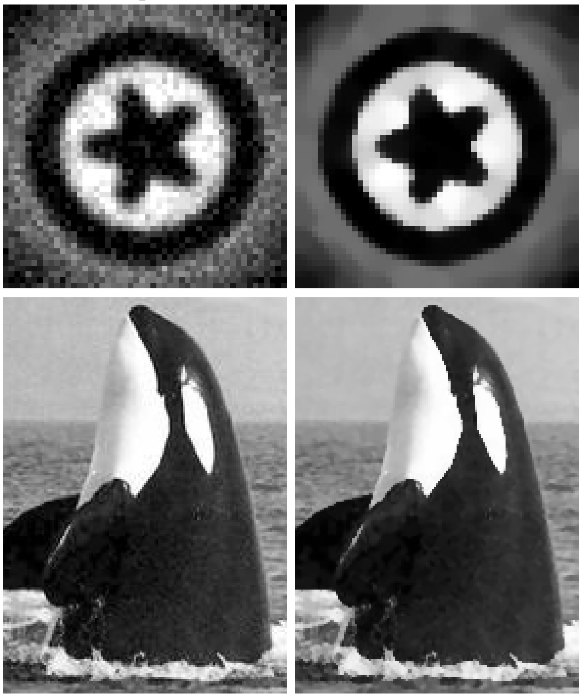


Figure 8: Simultaneous sharpening and denoising using contour stencils [4].

Contour stencils can be useful in discretizing image diffusion processes. Figure 8 demonstrates image enhancement using a combination of the Rudin-Osher shock filter [6] and TV-flow that has been discretized with contour stencils.

As another application, Figure 9 shows an image zooming result using contour stencils. The method approaches zooming as an inverse problem using a least-squares graph regularization. The regularization is adapted according to the edge orientations estimated from the contour stencils.

5. Conclusions

Contour stencils provide reliable orientation estimates at low computational cost, enabling better results in image processing applications.

Figure 9: (This is a color image.) Edge-adaptive zooming using contour stencils [5].

References:

[1] J. Bigün and G. H. Granlund. Optimal orientation detection of linear symmetry. In *IEEE First International Conference on Computer Vision*, pages 433–438, London, Great Britain, June 1987.

[2] Folkmar Bornemann and Tom März. Fast image inpainting based on coherence transport. *J. Math. Imaging Vis.*, 28(3):259–278, 2007.

[3] W. Förstner and E. Gulch. A fast operator for detection and precise location of distinct points, corners, and centers of circular features. pages 281–305, 1987.

[4] Pascal Getreuer. Contour stencils for edge-adaptive image interpolation. volume 7257, 2009.

[5] Pascal Getreuer. Image zooming with contour stencils. volume 7246, 2009.

[6] S. J. Osher and L. I. Rudin. Feature-oriented image enhancement using shock filters. *SIAM Journal on Numerical Analysis*, 27:919–940, 1990.

[7] Irwin Sobel and Jerome A. Feldman. A 3x3 isotropic gradient operator for image processing. Presented at a talk at the Stanford Artificial Project in 1968.

[8] Øivind Due Trier, Anil K. Jain, and Torfinn Taxt. Feature-extraction methods for character-recognition: A survey. *Pattern Recognition*, 29(4):641–662, April 1996.

[9] Joachim Weickert. *Anisotropic diffusion in image processing*. ECMI Series, Teubner-Verlag, Stuttgart, Germany, 1998.