SIAM J. IMAGING SCIENCES
Vol. 4, No. 3, pp. 954–979

Contour Stencils: Total Variation along Curves for Adaptive Image Interpolation[*]

Pascal Getreuer[†]

Abstract. Image interpolation is the problem of increasing the resolution of a given image. An important aspect of interpolation is accurate estimation of edge orientations. This work introduces contour stencils, a new method for estimating the image contours based on total variation along curves. Contour stencils are able to distinguish lines of different orientations, curves, corners, and other geometric features with a computationally efficient formula. Contour stencils are applied in designing an edge directed color interpolation method. The method incorporates an efficient approximation of deconvolution. Although most interpolation methods that involve deconvolution require either solving a large linear system or running many iterations, this method has linear complexity in the number of pixels and can be computed in one or a small number of passes through the image. Comparisons show that the proposed interpolation is competitive with existing methods.

Key words. contour stencils, image interpolation, total variation

AMS subject classifications. 68U10, 65J22, 65D05, 65K10, 65T60

DOI. 10.1137/100802785

1. Introduction.

Linear image interpolation is a thoroughly studied subject with the B-splines usually considered to be the best linear method in terms of optimal theoretic properties and cost-performance tradeoff [39, 54]. Yet due to the limitations inherent in linearity, all linear methods must have unsatisfactory effects such as blurring, jagged edges, and overshoot (halo) artifacts. To overcome this limitation, more recent works have turned to designing nonlinear methods.

Background. There are three prominent approaches for nonlinear interpolation: edge directed, diffusion, and patch-based methods.

Edge directed interpolation methods attempt to detect and adapt to the image edges. The common idea among these methods is to *interpolate along the edges, not across them*. Early works such as Ho and Gersho [28] and Allebach and Wong [2] discovered that estimating the edge orientations accurately is critical to the success of this approach. Methods proposed for estimating edge orientations for interpolation include estimating the gradient [1, 31], the Marr–Hildreth edge detector [2], data-dependent triangulation [14, 62], fitting a bilevel oriented step function [30], local covariance [34], ridge detection [56], nonparametric estimation or kernel regression [46, 52, 53], and sparse linear mixtures [37]. Additionally, some remarkably computationally efficient methods have been designed from low-cost ad hoc estimation strategies [28, 41, 48, 50, 51].

[*] Received by the editors July 19, 2010; accepted for publication (in revised form) August 4, 2011; published electronically September 29, 2011. Preliminary versions of this work have been published in [22, 23, 24]. This material is based upon work supported by the National Science Foundation under award DMS-1004694.
http://www.siam.org/journals/siims/4-3/80278.html

[†] CMLA, ENS Cachan 61, av. du Président Wilson, 94235 Cachan, France (getreuer@gmail.com).

Another approach to nonlinear interpolation is to design a diffusion process $\partial_t u = D(u)$, where D is a differential operator and $u|_{t=0}$ is an initial interpolation. Analogous to the "interpolate along edges" idea, D is typically an anisotropic operator such that u flows faster along edges than it does across them [7, 36, 49]. Many such diffusion-based methods solve the more general problem of *structure inpainting*. Inpainting is the problem of restoring an image where some region is unknown [8, 12, 38, 55, 58].

While edge directed and diffusion-based methods focus on recovering the geometric features of an image, patch-based methods also attempt to recover texture. An early patch-based technique is fractal zooming, where a fractal representation of the input image is found as an iterated function system [18]. A related approach is that of Freeman, Jones, and Pasztor [20], where interpolation is performed with a large database of image patches of low- and high-resolution pairs. It is also useful to compare patches of the same resolution, as proposed by Efros and Leung [15] for texture synthesis and Buades, Coll, and Morel [10] and Awate and Whitaker [4] for image denoising. Recently, these works have inspired several nonlocal interpolation methods [16, 25, 35, 45, 47].

Preliminary versions of this work have been published in [22, 23, 24]. This work improves substantially upon them with a rigorous analysis of contour stencil discretization, a consistent discretization formula, extension of contour stencils beyond lines to other geometric features, and analysis of an improved interpolation method that efficiently incorporates deconvolution. This work was published jointly with the Image Processing On Line article [21], which provides implementation details, source code, and an online demonstration.[1]

Outline. The rest of the paper is organized as follows. First, contour stencils are introduced for estimating image contours in section 2. Contour stencils are then applied in designing an edge directed interpolation method in section 3.2. The interpolation method is compared with several high-quality existing methods in section 3.3.

Notation. Pixels of a discrete image v are denoted by subscript as $v_{i,j}$, $i, j \in \mathbb{Z}$, and also as v_k, $k \in \mathbb{Z}^2$. The horizontal index i increases to the right, and the vertical index j increases upwards. When referencing a pixel beyond the boundaries of the image, symmetric extension or another boundary extension should be employed. If a function f is vector-valued, then $\|f\|_p$ is defined as the L^p-norm of the Euclidean norm of f.

2. Contour estimation.

A fundamental problem of importance to image interpolation is accurate estimation of edge orientations and local image structure. This section introduces contour stencils as a new approach to this problem.

2.1. Total variation along curves.

Contour stencils estimate image contours by measuring total variation (TV) along curves. As first introduced by Jordan [32], the *total variation* of a function $f : [a, b] \to \mathbb{R}$ is

$$\|f\|_{\mathrm{TV}[a,b]} := \sup \sum_{n=1}^{N} |f(t_n) - f(t_{n-1})| , \tag{2.1}$$

[1] See http://www.ipol.im/pub/algo/g_interpolation_geometric_contour_stencils.

where the supremum is over all finite sequences (t_n), $a = t_0 < t_1 < \cdots < t_N = b$. For a two-dimensional function $u : \mathbb{R}^2 \to \mathbb{R}$, define the *TV along curve C* as

$$\|u\|_{\mathrm{TV}(C)} := \|u \circ \gamma\|_{\mathrm{TV}[0,T]}\,, \tag{2.2}$$

where C is a smooth simple curve and $\gamma : [0, T] \to C$ parameterizes C. If u and γ are differentiable, then

$$\|u\|_{\mathrm{TV}(C)} = \int_0^T \left|\tfrac{\partial}{\partial t} u\big(\gamma(t)\big)\right| dt. \tag{2.3}$$

TV along a curve is different from the usual TV, $\|u\|_{\mathrm{TV}} = \int_{\mathbb{R}^2} |\nabla u|\, dx$. To see that $\|u\|_{\mathrm{TV}(C)}$ is independent of the parameterization γ, substitute $s = g(t)$, where g is differentiable and strictly increasing. Since $\left|\frac{ds}{dt}\right| \frac{dt}{ds} = 1$, we have with $\tilde{\gamma} = \gamma \circ g^{-1}$

$$\int_0^T \left|\tfrac{\partial}{\partial t} u\big(\gamma(t)\big)\right| dt = \int_{g(0)}^{g(T)} \left|\tfrac{ds}{dt} \tfrac{\partial}{\partial s} u\big(\gamma(g^{-1}(s))\big)\right| \tfrac{dt}{ds}\, ds = \int_{g(0)}^{g(T)} \left|\tfrac{\partial}{\partial s} u\big(\tilde{\gamma}(s)\big)\right| ds. \tag{2.4}$$

The independence from the parameterization implies that TV along curves is invariant under diffeomorphisms of space; i.e., $\|u\|_{\mathrm{TV}(C)} = \|u \circ \varphi^{-1}\|_{\mathrm{TV}(\varphi(C))}$ for φ a diffeomorphism on $\mathbb{R}^2$. For these reasons, we prefer TV instead of other seminorms along C (e.g., the H^1 seminorm), which do generally depend on the parameterization.

A key result in studying TV is the coarea formula [17, 42]. The coarea formula is a kind of curvilinear version of Fubini's theorem, decomposing an integral into integrals over the contours of a function.

Theorem 2.1 (see coarea formula [17]). *Suppose that Ω is an open subset of $\mathbb{R}^n$, f is Lipschitz on Ω, and $g \in L^1(\Omega)$. Then*

$$\int_\Omega g(x)\, |\nabla f(x)|\, dx = \int_{\mathbb{R}} \left(\int_{f^{-1}(\lambda)} g(x)\, d\mathcal{H}^{n-1}(x) \right) d\lambda, \tag{2.5}$$

where $\mathcal{H}^{n-1}$ is the $(n-1)$-dimensional Hausdorff measure.

The motivation for our contour estimation strategy is a consequence of the coarea formula: if $u \circ \gamma$ is Lipschitz, then $\|u\|_{\mathrm{TV}(C)}$ counts in a sense the number of times C crosses the contour lines,

$$\begin{aligned} \int_0^T \left|\tfrac{\partial}{\partial t} u\big(\gamma(t)\big)\right| dt &= \int_{\mathbb{R}} \mathcal{H}^0\big((u \circ \gamma)^{-1}(\lambda)\big)\, d\lambda \\ &= \int_{\mathbb{R}} \mathcal{H}^0\big\{x \in C : u(x) = \lambda\big\}\, d\lambda, \end{aligned} \tag{2.6}$$

and $\mathcal{H}^0$ is the counting measure. Therefore, contours of u can be estimated by finding curves C where $\|u\|_{\mathrm{TV}(C)}$ is small.

Finding curves with small TV is closely related to the *path variation*,

$$V(u)(x, y) := \inf \|u\|_{\mathrm{TV}(C)}\,, \qquad x, y \in \mathbb{R}^2, \tag{2.7}$$

where the infimum is over all curves C with endpoints x and y. That is, contours are identified as the minimal paths. Path variation was used by Arbeláez and Cohen [3] for image segmentation and by Yatziv and Sapiro [61] for image colorization.

2.2. Direct discretization. A challenge with $\|u\|_{\mathrm{TV}(C)}$ is its discretization. In the discrete setting, the image values are available only at the grid points. As illustrated in Figure 1, an approach to approximating $\|u\|_{\mathrm{TV}(C)}$ is to distort C into a curve $\tilde{C}$ that follows the grid points and then discretize $\|u\|_{\mathrm{TV}(\tilde{C})}$ by a finite difference approximation. The next two results investigate the error in making such an approximation of $\|u\|_{\mathrm{TV}(C)}$.

Theorem 2.2. *Let C and $\tilde{C}$ be smooth curves parameterized by $\gamma : [0,T] \to C$ and $\tilde{\gamma} : [0,T] \to \tilde{C}$. Then if u is twice continuously differentiable,*

$$\begin{aligned} \big|\|u\|_{\mathrm{TV}(\tilde{C})} - \|u\|_{\mathrm{TV}(C)}\big| \le{}& \|\nabla u\|_\infty \left\|\tilde{\gamma}' - \gamma'\right\|_1 \\ &+ \|D^2 u\|_\infty \left(\tfrac{|\tilde{C}|+|C|}{2} + \tfrac{1}{4}\left\|\tilde{\gamma}' - \gamma'\right\|_1\right) \|\tilde{\gamma} - \gamma\|_\infty , \end{aligned} \tag{2.8}$$

where $\|D^2 u\|_\infty$ is the maximum matrix 2-norm of the Hessian.

Proof. Let $\Gamma(t,s) = s\tilde{\gamma}(t) + (1-s)\gamma(t)$. Then

$$\begin{aligned} \big|\|u\|_{\mathrm{TV}(\tilde{C})} - \|u\|_{\mathrm{TV}(C)}\big| &\le \int_0^T \Big|\big|\tfrac{\partial}{\partial t}u\big(\tilde{\gamma}(t)\big)\big| - \big|\tfrac{\partial}{\partial t}u\big(\gamma(t)\big)\big|\Big|\,dt \\ &\le \int_0^T \Big|\tfrac{\partial}{\partial t}u\big(\tilde{\gamma}(t)\big) - \tfrac{\partial}{\partial t}u\big(\gamma(t)\big)\Big|\,dt \\ &= \int_0^T \left|\int_0^1 \tfrac{\partial^2}{\partial t \partial s}u\big(\Gamma(t,s)\big)\,ds\right|\,dt \\ &= \int_0^T \left|\int_0^1 \nabla u \cdot (\tilde{\gamma}' - \gamma') + \Gamma_t \cdot (D^2 u) \cdot (\tilde{\gamma} - \gamma)\,ds\right|\,dt \\ &\le \|\nabla u\|_\infty \left\|\tilde{\gamma}' - \gamma'\right\|_1 + \|\Gamma_t\|_1 \|D^2 u\|_\infty \|\tilde{\gamma} - \gamma\|_\infty . \end{aligned}$$

The quantity $\|\Gamma_t\|_1$ is bounded by

$$\begin{aligned} \|\Gamma_t\|_1 &\le \int_0^T \int_0^1 \left|s - \tfrac{1}{2}\right| \left|\tilde{\gamma}' - \gamma'\right| + \tfrac{1}{2}\left(\left|\tilde{\gamma}'\right| + \left|\gamma'\right|\right)\,ds\,dt \\ &= \tfrac{1}{4}\|\tilde{\gamma}' - \gamma'\|_1 + \tfrac{1}{2}(|\tilde{C}| + |C|). \qquad ■ \end{aligned}$$

Theorem 2.3. *Suppose $f \in C^2[0,T]$ and $0 = t_0 < t_1 < \cdots < t_N = T$. Define $h_i = t_i - t_{i-1}$ for $i = 1,\ldots,N$, and let h_{max} and h_{avg} denote the maximum and average values of h_i. Then*

$$\|f\|_{\mathrm{TV}[0,T]} - \tfrac{1}{3}T\tfrac{h_{max}^2}{h_{avg}}\left\|f''\right\|_\infty \le \sum_{i=1}^N |f(t_i) - f(t_{i-1})| \le \|f\|_{\mathrm{TV}[0,T]} . \tag{2.9}$$

Proof. The upper bound holds by definition of TV (2.2). For the lower bound, we first consider the error over a single panel of length h. Using the Green's function for the associated Poisson problem, f may be expressed as

$$f(t) = \frac{f(h) - f(0)}{h}t + \int_0^h f''(s)G(t,s)\,ds, \qquad G(t,s) = \begin{cases} (\frac{s}{h} - 1)t & \text{if } t \le s, \\ (\frac{t}{h} - 1)s & \text{if } t \ge s. \end{cases}$$

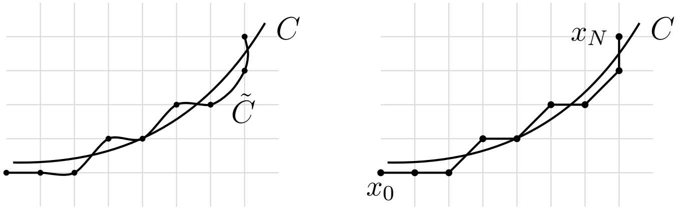


Figure 1. *Direct discretization of* $\|u\|_{\mathrm{TV}(C)}$. *Left: Approximation of* C *by* $\tilde{C}$. *Right: Finite difference approximation of* $\|u\|_{\mathrm{TV}(\tilde{C})}$.

With this representation, we obtain

$$\begin{aligned}\|f\|_{\mathrm{TV}[0,h]} &= \int_0^h \left| \tfrac{f(h)-f(0)}{h} + \int_0^h f''(s)\partial_t G(t,s)\,ds \right| dt \\ &\leq |f(h)-f(0)| + \|f''\|_\infty \int_0^h \int_0^h |\partial_t G(t,s)|\,ds\,dt \\ &= |f(h)-f(0)| + \|f''\|_\infty \frac{h^2}{3}.\end{aligned}$$

Therefore, the discrete TV over the panel $[0,h]$ is bounded from below as

$$\|f\|_{\mathrm{TV}[0,h]} - \|f''\|_\infty \frac{h^2}{3} \leq |f(h)-f(0)|\,.$$

Applying this bound over all panels yields

$$\|f\|_{\mathrm{TV}[0,T]} - \|f''\|_\infty \sum_{i=1}^N \frac{h_i^2}{3} \leq \sum_{i=1}^N |f(t_i)-f(t_{i-1})|\,.$$

Since $T = \sum_{i=1}^N h_i = N h_{avg}$, the sum can be bounded as

$$\sum_{i=1}^N \tfrac{h_i^2}{3} \leq \tfrac{1}{3} N h_{max}^2 = \tfrac{1}{3} T \tfrac{h_{max}^2}{h_{avg}}. \qquad ■$$

Together, the previous two results show that the TV along a curve $\|u\|_{\mathrm{TV}(C)}$ can be approximated as

$$\begin{aligned}&\left| \|u\|_{\mathrm{TV}(C)} - \sum_{i=1}^N |u(x_i)-u(x_{i-1})| \right| \leq \|\nabla u\|_\infty \left\|\tilde{\gamma}'-\gamma'\right\|_1 \\ &\quad + \|D^2 u\|_\infty \left(\tfrac{|\tilde{C}|+|C|}{2} + \tfrac{1}{4}\left\|\tilde{\gamma}'-\gamma'\right\|_1 \right) \|\tilde{\gamma}-\gamma\|_\infty + \|D^2 u\|_\infty \tfrac{1}{3} T \tfrac{h_{max}^2}{h_{avg}},\end{aligned} \tag{2.10}$$

where $x_0, \ldots, x_N \in \mathbb{Z}^2$ is a sequence of points on the grid sampled along a curve $\tilde{C}$ that is nearby and approximately parallel to C.

Unfortunately, for an arbitrary curve C it is unclear how to choose $\tilde{C}$. The three terms in the error bound (2.10) impose conflicting requirements: the curves should be close together, the curves should be approximately parallel, and the sample points should be close together. For example, if the maximum sampling distance h_{max} is limited to $\sqrt{2}$, then $\tilde{C}$ has difficulty following any direction except axial and diagonal (see Figure 1). So it is difficult to apply this approximation reliably.

Earlier versions of this work [22, 23, 24] restricted C to straight line segments. Following Theorem 2.2, the TV was estimated by averaging estimates (2.10) over several nearby parallel line segments. This leads to a discretization of the form

$$\|u\|_{\mathrm{TV}(C)} \approx \sum_{m,n\in\mathbb{Z}^2} \mathcal{S}(m,n)\big|u(m)-u(n)\big|, \tag{2.11}$$

where $\mathcal{S}:\mathbb{Z}^2\times\mathbb{Z}^2\to\mathbb{R}$ determines weighted edges between the pixels. The edges are selected so that they approximate a set of line segments parallel to C.

2.3. Collections of curves.

Instead of computing the TV over a single curve, we consider TV integrated over a collection of curves C_λ, $\lambda\in\mathbb{R}$:

$$\int_{\mathbb{R}} \|u\|_{\mathrm{TV}(C_\lambda)}\,d\lambda. \tag{2.12}$$

Compared to (2.10), discretization of (2.12) leads to more robust estimates of the image contours. To ensure that the curves do not cross, we choose the collection $C_\lambda=\psi^{-1}(\lambda)$ as the contours of a Lipschitz and differentiable function $\psi:\Omega\to\mathbb{R}$. An additional convenience with such a collection is that (2.12) is simplified by the coarea formula.

Theorem 2.4. *Let Ω be an open subset of $\mathbb{R}^2$. Suppose that u and ψ are differentiable functions on Ω with $|\nabla u|\in L^1(\Omega)$ and ψ Lipschitz with $\nabla\psi\neq 0$; then*

$$\int_{\mathbb{R}} \|u\|_{\mathrm{TV}(\psi^{-1}(\lambda))}\,d\lambda = \int_\Omega |\nabla\psi(x)|\,\big|\tfrac{\nabla\psi^\perp(x)}{|\nabla\psi(x)|}\cdot\nabla u(x)\big|\,dx. \tag{2.13}$$

Proof. The left-hand side of (2.13) can be written as

$$\int_{\mathbb{R}} \|u\|_{\mathrm{TV}(\psi^{-1}(\lambda))}\,d\lambda = \int_{\mathbb{R}}\left(\int_{\psi^{-1}(\lambda)} g(x)\,d\mathcal{H}^1(x)\right)d\lambda$$

with $g(x)=\big|\frac{\nabla\psi(x)^\perp}{|\nabla\psi(x)|}\cdot\nabla u(x)\big|\in L^1(\Omega)$ since $|g(x)|\le|\nabla u(x)|$. Applying the coarea formula (Theorem 2.1) yields the result. ■

There is a resemblance between (2.13) and the usual two-dimensional TV, $\int_\Omega|\nabla u|\,dx$. TV over a collection of curves is equivalent to replacing $|\nabla u|$ with the quantity $|\nabla\psi|\,|\frac{\nabla\psi^\perp}{|\nabla\psi|}\cdot\nabla u|$. The factor $|\nabla\psi|$ accounts for the spatial density of the level lines of ψ. Energies of the form $\int|\nabla\psi^\perp\cdot\nabla u|^p\,dx$ have been used in variational image fusion [6, 40].

A result analogous to Theorem 2.2 on the perturbation of curves holds for this representation. If vector fields $\mathcal{S}$ and $\tilde{\mathcal{S}}$ are similar, then the TVs of u along those vector fields are similar.

Lemma 2.5. *Suppose that $|\nabla u|,|\nabla\tilde{u}|,|\mathcal{S}|,|\tilde{\mathcal{S}}|\in L^2(\Omega)$; then*

$$\left|\int_\Omega|\mathcal{S}\cdot\nabla u|\,dx-\int_\Omega|\tilde{\mathcal{S}}\cdot\nabla\tilde{u}|\,dx\right|\le\|\tilde{\mathcal{S}}\|_2\,\|\nabla u-\nabla\tilde{u}\|_2+\|\mathcal{S}-\tilde{\mathcal{S}}\|_2\,\|\nabla u\|_2\,. \tag{2.14}$$

Proof. By the reverse triangle inequality,

$$\begin{aligned}\left|\int_\Omega |\tilde{\mathcal{S}}\cdot\nabla u|\,dx - \int_\Omega |\tilde{\mathcal{S}}\cdot\nabla\tilde{u}|\,dx\right| &\le \int_\Omega \big||\tilde{\mathcal{S}}\cdot\nabla u| - |\tilde{\mathcal{S}}\cdot\nabla\tilde{u}|\big|\,dx\\ &\le \int_\Omega |\tilde{\mathcal{S}}\cdot\nabla u - \tilde{\mathcal{S}}\cdot\nabla\tilde{u}|\,dx\\ &\le \|\tilde{\mathcal{S}}\|_2\|\nabla u - \nabla\tilde{u}\|_2.\end{aligned}$$

Similarly, for the perturbation in $\mathcal{S}$,

$$\begin{aligned}\left|\int_\Omega |\mathcal{S}\cdot\nabla u|\,dx - \int_\Omega |\tilde{\mathcal{S}}\cdot\nabla u|\,dx\right| &\le \int_\Omega |\mathcal{S}\cdot\nabla u - \tilde{\mathcal{S}}\cdot\nabla u|\,dx\\ &\le \|\mathcal{S}-\tilde{\mathcal{S}}\|_2\,\|\nabla u\|_2\,. \qquad \blacksquare\end{aligned}$$

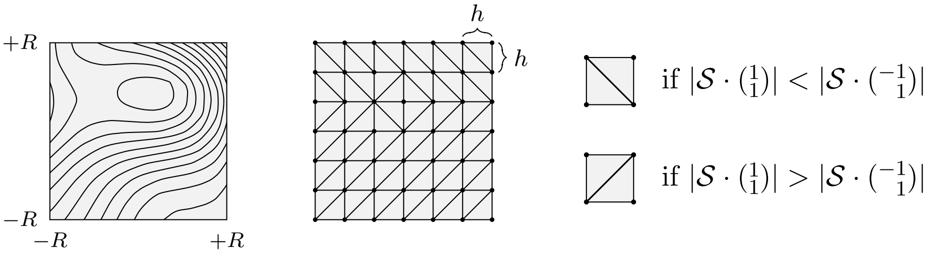
2.4. Contour stencils. Contour stencils are derived as a discretization of TV over a collection of curves. Let Ω be the square $\Omega = \{x \in \mathbb{R}^2 : |x_1| < R, |x_2| < R\}$, and given a resolution $h = R/N$, let $(\Omega_{i,j})$ partition Ω into square-shaped cells of size $h \times h$. We define $\mathcal{S}$ by computing the average of $\nabla\psi^\perp$ in each cell:

$$\mathcal{S}(x) = \frac{1}{h^2}\int_{\Omega_{i,j}} \nabla\psi(x)^\perp\,dx, \quad x \in \Omega_{i,j}. \tag{2.15}$$

Let T be a triangulation of Ω that splits each $\Omega_{i,j}$ into two triangles (see Figure 2). There are two ways to split a cell into two triangles, which we will address later. Let $\mathcal{I}u$ be a piecewise linear approximation of u such that $\mathcal{I}u(x) = u(x)$ at the grid points $x = (hi, hj)$ and $\mathcal{I}u$ is affine on each triangle of T. TV integrated over a collection of curves is discretized as

$$\int_\Omega |\mathcal{S}\cdot\nabla\mathcal{I}u|\;dx \approx \int_\Omega |\nabla\psi^\perp\cdot\nabla u|\,dx. \tag{2.16}$$

We call the discrete vector field $\mathcal{S}$ a *contour stencil* because it specifies a predetermined pattern of contours to test on the image u. The following theorem shows that (2.16) is a consistent discretization.


$+R$
$-R$
$-R$
$+R$
h
h
if $|\mathcal{S}\cdot\binom{1}{1}| < |\mathcal{S}\cdot\binom{-1}{1}|$
if $|\mathcal{S}\cdot\binom{1}{1}| > |\mathcal{S}\cdot\binom{-1}{1}|$

Curves C_λ in Ω Triangulation of Ω Splitting the cells $\Omega_{i,j}$

Figure 2. *Discretization of* Ω.

Theorem 2.6. *Suppose $u, \psi \in H^2(\Omega)$; then (2.16) is a first-order accurate discretization of (2.12),*

$$\left| \int_{\mathbb{R}} \|u\|_{\mathrm{TV}(\psi^{-1}(\lambda))} \, d\lambda - \int_{\Omega} |\mathcal{S} \cdot \nabla \mathcal{I} u| \, dx \right| \le h\big(C_1 \|\nabla\psi\|_{L^2} |u|_{H^2} + \tfrac{\sqrt{2}}{\pi} |\psi|_{H^2} \|\nabla u\|_{L^2}\big), \tag{2.17}$$

where $C_1 \lessapprox 0.493$ is a constant and $|u|_{H^2}$ is the seminorm

$$|u|_{H^2} := \left(\int_{\Omega} |\partial^2_{x_1} u|^2 + 2|\partial_{x_1}\partial_{x_2} u|^2 + |\partial^2_{x_2} u|^2 \, dx \right)^{1/2}.$$

Proof. By Lemma 2.5,

$$\left| \int_{\mathbb{R}} \|u\|_{\mathrm{TV}(\psi^{-1}(\lambda))} \, d\lambda - \int_{\Omega} |\mathcal{S} \cdot \nabla \mathcal{I} u| \, dx \right| \le \|\mathcal{S}\|_{L^2} \|\nabla u - \nabla \mathcal{I} u\|_{L^2} + \|\nabla\psi^{\perp} - \mathcal{S}\|_{L^2} \|\nabla u\|_{L^2}.$$

We obtain the result by bounding $\|\mathcal{S}\|_{L^2}$, $\|\nabla u - \nabla \mathcal{I} u\|_{L^2}$, and $\|\nabla\psi^{\perp} - \mathcal{S}\|_{L^2}$.

Over a single cell $\Omega_{i,j}$, the orthogonality $\int_{\Omega_{i,j}} \mathcal{S}(\nabla\psi^{\perp} - \mathcal{S}) \, dx = 0$ implies that $\|\mathcal{S}\|_{L^2(\Omega_{i,j})} \le \|\nabla\psi^{\perp}\|_{L^2(\Omega_{i,j})} = \|\nabla\psi\|_{L^2(\Omega_{i,j})}$. Hence summing the energy over all cells,

$$\|\mathcal{S}\|_{L^2(\Omega)} \le \|\nabla\psi\|_{L^2(\Omega)}.$$

The quantity $\|\nabla\psi^{\perp} - \mathcal{S}\|_{L^2}$ is bounded using the Poincaré inequality with the constant due to Payne and Weinberger [44]. Over a single cell,

$$\|\nabla\psi^{\perp} - \mathcal{S}\|_{L^2(\Omega_{i,j})} \le \frac{\mathrm{diam}(\Omega_{i,j})}{\pi} |\psi|_{H^2(\Omega_{i,j})},$$

where $\mathrm{diam}(\Omega_{i,j}) = \sqrt{2}\, h$ is the maximum diameter of $\Omega_{i,j}$. Hence over all cells,

$$\|\nabla\psi^{\perp} - \mathcal{S}\|_{L^2(\Omega)} \le \frac{\sqrt{2}\, h}{\pi} |\psi|_{H^2(\Omega)}.$$

Finally, we bound $\|\nabla u - \nabla \mathcal{I} u\|_{L^2}$. Observing that $\mathcal{I} u$ is a piecewise linear interpolant of u over isosceles right triangles with side length h, the result from Kikuchi and Liu [33] applies to obtain

$$\|\nabla u - \nabla \mathcal{I} u\|_{L^2(\Omega)} \le C_1 h \, |u|_{H^2(\Omega)}.$$

The constant C_1 is $C_1 = 1/\sqrt{\lambda}$, where λ is the minimum positive solution of $\sqrt{\lambda} + \tan\sqrt{\lambda} = 0$ and it has been proved that $0.49282 < C_1 < 0.49293$. Combining these estimates yields the result

$$\|\mathcal{S}\|_{L^2} \|\nabla u - \nabla \mathcal{I} u\|_{L^2} + \|\nabla\psi^{\perp} - \mathcal{S}\|_{L^2} \|\nabla u\|_{L^2} \le h\big(C_1 \|\nabla\psi\|_{L^2} |u|_{H^2} + \tfrac{\sqrt{2}}{\pi} |\psi|_{H^2} \|\nabla u\|_{L^2}\big). \qquad \blacksquare$$

We choose which way to split a cell into two triangles according to the orientation of $\mathcal{S}$ (see Figure 2). Let $(\alpha,\beta)^T$ be the value of $\mathcal{S}$ in $\Omega_{i,j}$. If $|(\begin{smallmatrix}\alpha\\ \beta\end{smallmatrix})\cdot(\begin{smallmatrix}1\\ 1\end{smallmatrix})| \ge |(\begin{smallmatrix}\alpha\\ \beta\end{smallmatrix})\cdot(\begin{smallmatrix}1\\ -1\end{smallmatrix})|$, then the cell is split along the diagonal (i,j) to $(i+1,j+1)$; otherwise, the cell is split along the other diagonal $(i+1,j)$ to $(i,j+1)$. The integral over $\Omega_{i,j}$ is

$$\begin{aligned} &V_{i,j}(\mathcal{S},u) \\ &:= \int_{\Omega_{i,j}} |\mathcal{S}\cdot\nabla\mathcal{I}u|\,dx = \begin{cases} \frac{h}{2}\big(\big|\alpha u_{i+1,j+1} - (\alpha-\beta)u_{i,j+1} - \beta u_{i,j}\big| & \alpha\beta \ge 0, \\ \quad + \big|\beta u_{i+1,j+1} + (\alpha-\beta)u_{i+1,j} - \alpha u_{i,j}\big|\big), & \\ \frac{h}{2}\big(\big|\alpha u_{i,j+1} - (\alpha+\beta)u_{i+1,j+1} + \beta u_{i+1,j}\big| & \alpha\beta \le 0, \\ \quad + \big|\beta u_{i,j+1} - (\alpha+\beta)u_{i,j} + \alpha u_{i+1,j}\big|\big), & \end{cases} \end{aligned} \tag{2.18}$$

where $u_{i,j} := u(hi,hj)$. This choice of splitting is such that the $(\alpha \pm \beta)$ factors cancel at diagonal orientations, where $|\alpha| = |\beta|$. The integral over Ω is

$$V(\mathcal{S},u) := \int_\Omega |\mathcal{S}\cdot\nabla\mathcal{I}u|\,dx = \sum_{i=-N}^{N-1}\sum_{j=-N}^{N-1} V_{i,j}(\mathcal{S},u). \tag{2.19}$$

The computation simplifies to sums of absolute differences when $\mathcal{S}$ is axially or diagonally oriented:

$$\begin{cases} \frac{h|\alpha|}{2}\big(\big|u_{i+1,j+1} - u_{i,j+1}\big| + \big|u_{i+1,j} - u_{i,j}\big|\big), & \beta = 0, \\ \frac{h|\beta|}{2}\big(\big|u_{i,j+1} - u_{i,j}\big| + \big|u_{i+1,j+1} - u_{i+1,j}\big|\big), & \alpha = 0, \\ h\,|\alpha|\,\big|u_{i+1,j+1} - u_{i,j}\big|, & \alpha = -\beta, \\ h\,|\alpha|\,\big|u_{i,j+1} - u_{i+1,j}\big|, & \alpha = \beta. \end{cases} \tag{2.20}$$

A discrete analogy of (2.13) holds for these cases: they discretize $\int|\nabla\psi^\perp\cdot\nabla u|\,dx$, yet they can also be seen to be sums of (2.10) along adjacent parallel lines.

To extend to vector-valued images, $V(\mathcal{S},u)$ is defined by summing (2.19) applied to each channel. For color images, we use the channels defined by the $\mathrm{YP_BP_R}$ space:

$$\begin{pmatrix} Y \\ P_B \\ P_R \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.168736 & -0.331264 & 0.5 \\ 0.5 & -0.418688 & -0.081312 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}. \tag{2.21}$$

2.5. The best-fitting stencil. From here on, we fix $h = 1$ at the resolution of the input image $v : \mathbb{Z}^2 \to \mathbb{R}$. The contours of v are estimated by finding a stencil with small TV. Define the best-fitting stencil

$$\mathcal{S}^\star = \arg\min_{\mathcal{S}\in\Sigma} V(\mathcal{S},v), \tag{2.22}$$

where Σ is a set of candidate stencils. The best-fitting stencil $\mathcal{S}^\star$ is an estimate of the underlying contours.

The contours could be estimated globally by finding an optimal $\mathcal{S}^\star$ that describes the whole image. The minimization is a challenging high-dimensional problem in this case since Σ

must be large to be adequately descriptive, for example, the set of all unit magnitude vector fields satisfying a regularity requirement.

We instead take a local approach by estimating contours over small neighborhoods. Define the best-fitting stencil at pixel $k \in \mathbb{Z}^2$ by translation,

$$\mathcal{S}^\star(k) = \operatorname*{arg\,min}_{\mathcal{S}\in\Sigma} V\big(\mathcal{S}, v(\cdot - k)\big). \tag{2.23}$$

By limiting the estimation to a neighborhood, Σ is adequately descriptive with a few dozen stencils, so minimization by a simple exhaustive search is reasonable. This stencil selection is essentially a kind of template matching in which Σ is the set of templates and V is the dissimilarity function [5, 27].

Let $\mathcal{S}^{\star 2}$ denote a second best-fitting stencil (which may be nonunique),

$$\mathcal{S}^{\star 2} := \operatorname*{arg\,min}_{\mathcal{S}\in\Sigma\backslash\mathcal{S}^\star} V(\mathcal{S}, v). \tag{2.24}$$

The following theorem expresses the stability of the best-fitting stencil in terms of its separation from a second best-fitting stencil.

Theorem 2.7. *Suppose that for two images v and $\tilde{v}$*

$$\|v - \tilde{v}\|_{\ell^p} < \frac{V(\mathcal{S}^{\star 2}, v) - V(\mathcal{S}^\star, v)}{4\sqrt{2}\,\max\limits_{\mathcal{S}\in\Sigma} \|\mathcal{S}\|_{\ell^q}}, \tag{2.25}$$

where $\mathcal{S}^\star$ and $\mathcal{S}^{\star 2}$ are the best- and second best-fitting stencils on v and $1 \le p, q \le \infty$ are Hölder conjugates. Then $\mathcal{S}^\star$ is also the best-fitting stencil on $\tilde{v}$.

Proof. By the reverse triangle inequality,

$$\begin{aligned}
|V_{i,j}(\mathcal{S}, v) - V_{i,j}(\mathcal{S}, \tilde{v})| &\le \begin{cases}
\tfrac{1}{2}\big((|\alpha| + |\beta|)\,|v_{i+1,j+1} - \tilde{v}_{i+1,j+1}| & \alpha\beta \ge 0,\\
\quad +\ |\alpha - \beta|\,|v_{i,j+1} - \tilde{v}_{i,j+1}| & \\
\quad +\ |\alpha - \beta|\,|v_{i+1,j} - \tilde{v}_{i+1,j}| & \\
\quad +\ (|\alpha| + |\beta|)\,|v_{i,j} - \tilde{v}_{i,j}|\big), & \\
\tfrac{1}{2}\big((|\alpha| + |\beta|)\,|v_{i,j+1} - \tilde{v}_{i,j+1}| & \alpha\beta \le 0,\\
\quad +\ |\alpha + \beta|\,|v_{i+1,j+1} - \tilde{v}_{i+1,j+1}| & \\
\quad +\ |\alpha + \beta|\,|v_{i,j} - \tilde{v}_{i,j}| & \\
\quad +\ (|\alpha| + |\beta|)\,|v_{i+1,j} - \tilde{v}_{i+1,j}|\big), &
\end{cases}\\
&\le \frac{\sqrt{2}}{2} \sum_{k=0}^{1} \sum_{\ell=0}^{1} \sqrt{\alpha^2 + \beta^2}\,|v_{i+k,j+\ell} - \tilde{v}_{i+k,j+\ell}|\,,
\end{aligned}$$

where $(\alpha, \beta)^T = \mathcal{S}$ in cell $\Omega_{i,j}$. So the effect of perturbation in v is

$$\begin{aligned}
|V(\mathcal{S}, v) - V(\mathcal{S}, \tilde{v})| &\le \frac{\sqrt{2}}{2} \sum_{i,j} \sum_{k=0}^{1} \sum_{\ell=0}^{1} |\mathcal{S}_{i,j}|\,|v_{i+k,j+\ell} - \tilde{v}_{i+k,j+\ell}|\\
&\le M\,\|v - \tilde{v}\|_p\,, \quad M = 2\sqrt{2} \max_{\mathcal{S}\in\Sigma} \|\mathcal{S}\|_q\,.
\end{aligned}$$

For any $\mathcal{S} \in \Sigma \backslash \mathcal{S}^\star$,

$$\begin{aligned} V(\mathcal{S}, \tilde{v}) - V(\mathcal{S}^\star, \tilde{v}) &\geq V(\mathcal{S}, v) - V(\mathcal{S}^\star, v) - 2M \left\| v - \tilde{v} \right\|_p \\ &\geq V(\mathcal{S}^{\star 2}, v) - V(\mathcal{S}^\star, v) - 2M \left\| v - \tilde{v} \right\|_p > 0. \end{aligned}$$

Therefore, $\mathcal{S}^\star$ is also the best-fitting stencil for $\tilde{v}$. ■

It is possible that the minimum (2.23) is not unique, for example, when the image is locally constant. To ensure the stability of the minimum, we use the stencil $\mathcal{S}^\star$ only if the right-hand side of (2.25) is above a threshold, i.e.,

$$\frac{V(\mathcal{S}^{\star 2}, v) - V(\mathcal{S}^\star, v)}{4\sqrt{2} \max\limits_{\mathcal{S} \in \Sigma} \|\mathcal{S}\|_{\ell^1}} \geq T, \tag{2.26}$$

and otherwise fall back to an isotropic model. In the examples, the threshold is fixed at $T = 10^{-4}$ relative to pixel intensities in the range $[0, 1]$.

To design the stencils $\mathcal{S} \in \Sigma$, we must choose functions $\psi_\mathcal{S}$. We suppose that the observed image has been blurred by a point-spread function (PSF) h, so we use $\psi_\mathcal{S} = h * \varphi_\mathcal{S}$ to estimate the contours of the blurred image and then consider $\varphi_\mathcal{S}$ as a model of the contours of the underlying image. The $\varphi_\mathcal{S}$ are selected as signed distance functions of elementary curves $C_\mathcal{S}$; that is, $\varphi_\mathcal{S}$ is a solution of the eikonal equation

$$\begin{cases} |\nabla \varphi(x)| = 1 & \text{in } \Omega, \\ \quad \varphi(x) = 0 & \text{on } C_\mathcal{S}. \end{cases} \tag{2.27}$$

A distance function is a good choice for $\varphi_\mathcal{S}$ because it has evenly spaced contours and for piecewise smooth $C_\mathcal{S}$ it is a.e. differentiable. Distance functions for several curves are listed in Table 1. Other curves, such as T-junctions, are similarly possible.

An effective stencil set is formed by using several rotations of such distance functions. The stencils themselves are obtained by discretizing $\psi_\mathcal{S} = h * \varphi_\mathcal{S}$ with (2.15) and then dividing $\mathcal{S}$ by $\|\mathcal{S}\|_1$ so that all stencils have the same average magnitude. In the examples, the stencil set is defined by a circle; lines at orientations $\frac{\pi}{32}j$, $j = 0, \ldots, 31$; and corners and parabolas with $a = \frac{1}{\sqrt{2}}$ and $a = 1$ at orientations $\frac{\pi}{4}j$, $j = 0, \ldots, 7$, for a total of 57 stencils. Several of these stencils are shown in Figure 3. Instead of a square domain, we use twelve cells approximating a disk for better isotropy. The PSF h is a Gaussian with standard deviation 0.5 unless mentioned otherwise.

The circle stencil has a scale invariance. The contours of $\varphi = \sqrt{x_1^2 + x_2^2}$ are circles centered at the origin of any radius (up to the size of Ω), so the stencil equally detects any of these circles. Therefore, the circle stencil implies an isotropic model of the contours, as it prefers no direction and has arbitrary curvature.

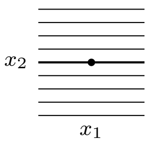
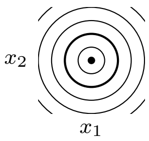
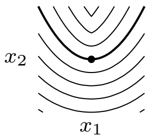
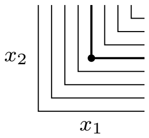
The proposed stencil set satisfies

$$\mathcal{S} = \mathop{\arg\min}_{\mathcal{S}' \in \Sigma} V(\mathcal{S}', \psi_\mathcal{S}) \quad \text{for every } \mathcal{S} \in \Sigma, \tag{2.28}$$

where $\psi_\mathcal{S} = h * \varphi_\mathcal{S}$ and $\varphi_\mathcal{S}$ is the distance function from which $\mathcal{S}$ was derived. That is, $\mathcal{S}$ is the best-fitting stencil on $(h * \varphi_\mathcal{S})$. The second best-fitting stencils $\mathcal{S}^{\star 2}$ on $\psi_\mathcal{S}$ indicate which stencils are closest to being confused with $\mathcal{S}$. Not surprisingly, these tend to be stencils which geometrically resemble $\mathcal{S}$; see Table 2.

Table 1

Several elementary curves and their distance functions.

Curve	Distance function	Contours
Line		
$C = \{x_2 = 0\}$	$\varphi(x) = x_2$	
Circle		
$C = \{x_1^2 + x_2^2 = 1\}$	$\varphi(x) = \sqrt{x_1^2 + x_2^2} - 1$	
Parabola		
$C = \{x_2 = \frac{a}{2}x_1^2\}$	$\varphi(x) = (x_2 - \frac{a}{2}s^2)\sqrt{1+(as)^2}$, where s is the minimum nonnegative root of $(\lvert x_1\rvert - s) + (x_2 - \frac{a}{2}s^2)as$	
Corner		
$C = \{x_1x_2 = 0,$ $x_1 \geq 0, x_2 \geq 0\}$	$\varphi(x) = \min\{x_1, x_2\}$	

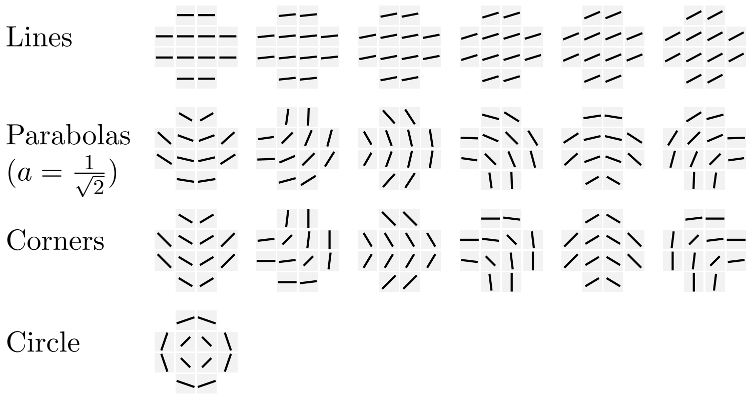


Figure 3. *Contour stencils at several orientations for lines, parabolas, corners, and a circle.*

2.6. Discussion. To estimate the image contours, the best-fitting stencil $\mathcal{S}^\star(k)$ is determined at each pixel k using (2.23). Figure 4 shows an example. For each pixel, a short line is drawn to indicate the curve $C_{\mathcal{S}^\star(k)}$ modeled by the best-fitting stencil. The lines are colored according to the type of geometric feature (line, curve, or corner). For comparison, two standard contour estimation methods are also shown.

Contour tracing [13] extracts the contours by tracking the boundaries of the image level

Table 2

The second best-fitting stencils $\mathcal{S}^{\star 2}$ in (2.28) indicate which stencils are closest to being confused.

Stencil $\mathcal{S}$	Stencils closest to being confused with $\mathcal{S}$
Line with $\theta = j\frac{\pi}{32}$	Line with $\theta = (j \pm 1)\frac{\pi}{32}$
Parabola with $a = \frac{1}{\sqrt{2}}, \theta = j\frac{\pi}{4}$	Parabola with $a = 1, \theta = j\frac{\pi}{4}$
Parabola with $a = 1, \theta = j\frac{\pi}{4}$	Parabola with $a = \frac{1}{\sqrt{2}}, \theta = j\frac{\pi}{4}$
Corner with $\theta = j\frac{\pi}{4}$	Parabola with $a = 1, \theta = (j-1)\frac{\pi}{4}$
Circle	Corners with $\theta = 0, \frac{\pi}{2}, 2\frac{\pi}{2}, 3\frac{\pi}{2}$

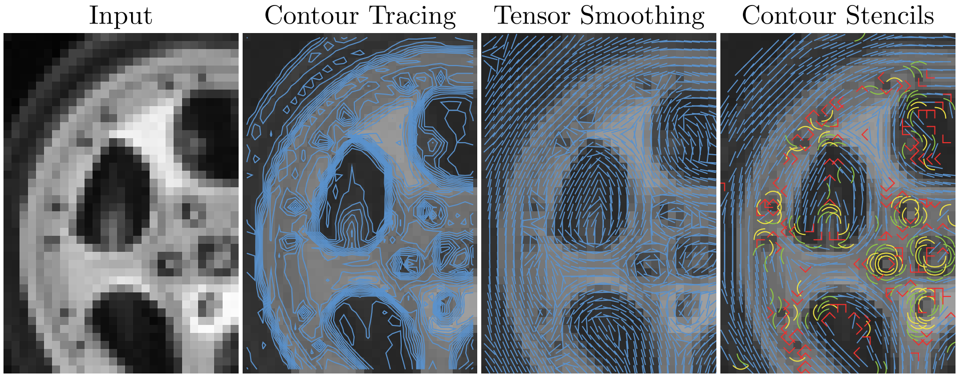


Figure 4. *Comparison of contour estimation methods on a grayscale image.*

sets $v_\lambda := \{x : v(x) \le \lambda\}$. The result shown in Figure 4 was produced by the MATLAB `contour` plot. This method recovers a global estimate of the contours that represents corners and other geometric structures. However, its edges have staircase artifacts, and there is a tendency to produce disconnected islands (for example, in the upper-left corner of the image). A significant limitation is that the method depends on extracting level sets, so it cannot be applied jointly to color images.

Another popular method is tensor smoothing [9, 19]. Given a vector field $c(x)$ approximating $\nabla v^\perp(x)$, define the tensor

$$J(x) = c(x) \otimes c(x) := \begin{pmatrix} c_1(x)^2 & c_1(x)c_2(x) \\ c_1(x)c_2(x) & c_2(x)^2 \end{pmatrix}, \tag{2.29}$$

where $J(x)$ is a 2×2 matrix at each point. The tensor is smoothed by Gaussian convolution applied per component, $\tilde{J} = G * J$. Finally, a denoised orientation field $\tilde{c}(x)$ is obtained as the eigenvector associated with the larger[2] eigenvalue of $\tilde{J}(x)$. In Figure 4, c was computed with centered finite differences, and G was a Gaussian with standard deviation 1. The smoothing eliminates staircasing, and the method is very effective over most of the image. However, some fine features like the bolts around the rim are lost. The method is limited locally to estimating straight lines. The presence of a corner can be inferred from the tensor eigenvalues [19, 59], but the method does not capture the orientation of the corner or other geometric structures.

[2] An equivalent formulation, as described by Weickert [59], is to estimate orientation as the eigenvector associated with the *smaller* eigenvalue of the smoothed structure tensor $G * (\nabla v \otimes \nabla v)$.

Compared to these two methods, estimation with contour stencils combines some of their advantages while avoiding the drawbacks. Contour stencils avoid staircasing, respond to fine features, and capture curves and corners. A disadvantage with contour stencils is that the angular resolution and distinction of geometric features are limited to what is represented in the stencil set. On the other hand, the choice of stencil set provides the flexibility to tune the method to detect particular types of features.

Similar to contour stencils, several existing edge-adaptive interpolation methods also use absolute differences for estimating edge orientations. Takeda, Farsiu, and Milanfar [53, subsection II-C] considered a general cost function

$$E(u) = \sum_{n\in\mathbb{Z}^2} \left| \sum_{k\in\mathbb{Z}^2} W(n,k)(u_{k+n} - u_k) \right|^q . \tag{2.30}$$

Different image regularizations are obtained according to the choice of the weights $W(n,k)$. The contour stencil TV sum (2.19) is obtained as a special case with $q = 1$ and W selected according to (2.18).

Stepin's hqx pixel art scaling methods [51] estimate the local structure about a pixel m by testing whether $|u_m - u_n|$ exceeds a fixed threshold for each of its eight adjacent neighbors n. These eight tests lead to 256 possible structural configurations. The challenge with this approach is tuning the threshold, which should be smaller than the least significant edge jump yet large enough to ignore smooth gradients and minor variations. Contour stencil selection (2.23) avoids this problem by comparing the stencil variations to each other rather than to a threshold.

In Muresan's AQua-2 method [41], a pixel is assigned both an axial label and a diagonal label by comparing the absolute values of second-order differences:

$$\begin{aligned} \textit{axial label} &= \operatorname{sign}\big(\,|u_{i-1,j} - 2u_{i,j} + u_{i+1,j}| \\ &\qquad - |u_{i,j-1} - 2u_{i,j} + u_{i,j+1}|\,\big), \qquad (2.31)\\ \textit{diagonal label} &= \operatorname{sign}\big(\,|u_{i-1,j-1} - 2u_{i,j} + u_{i+1,j+1}| \\ &\qquad - |u_{i-1,j+1} - 2u_{i,j} + u_{i+1,j-1}|\,\big). \qquad (2.32)\end{aligned}$$

Similarly, Sajjad, Khattuk, and Jafri [50] estimate edge orientation by thresholding absolute values of second-order differences. Instead of TV along curves, one could more generally consider the $W^{k,p}$ seminorm

$$\|u\|_{C,k,p} = \int_0^T \big|\tfrac{\partial^k}{\partial s^k} u\big(\gamma(s)\big)\big|^p \, ds, \tag{2.33}$$

where γ is an arc-length parameterization of C. Contour stencils, on the other hand, are based on first-order derivatives. Derivatives of order k have the advantage that they annihilate locally $(k-1)$ degree polynomials, but at least $(k+1)$ samples are needed to estimate them, which is a tradeoff between regularity and localization. Localization has arguably greater importance because natural images are usually finely detailed and irregular, so in this respect first-order derivatives are the best choice.

3. Interpolation.

Given a discretely sampled image $v : \mathbb{Z}^2 \to \mathbb{R}$, interpolation estimates the underlying image $u : \mathbb{R}^2 \to \mathbb{R}$ from which v was discretized. We assume a discretization model as convolution followed by sampling

$$v = \operatorname{sample}(h * u), \tag{3.1}$$

where h, $\int h = 1$, is a PSF and $\operatorname{sample}(\cdot)$ denotes point-value sampling on the $\mathbb{Z}^2$ grid. For simplicity, this work focuses on the interpolation of noise-free images v or where v has been denoised as a preprocessing step.

3.1. Objectives.

Three sensible design objectives for interpolation are smoothness, step response, and directional selectivity:

Objective 1: *Smoothness.* In smooth regions, the method reproduces low-order polynomials.

Objective 2: *Step response.* Edges should interpolate without excessive blurring or overshoots.

Objective 3: *Directional selectivity.* Edges at any angle should interpolate without jagged artifacts.

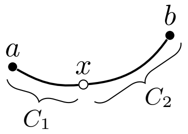
However, a design following multiple objectives is easily self-conflicting. For example, a linear method that reproduces quadratic polynomials necessarily produces edges with overshoots, which is a conflict between Objectives 1 and 2.

Instead, we combine the above three objectives into one: *contour consistency.* The local orientations of the contours should match and no new contour lines should be created. Similar contour-motivated objectives have been applied successfully in previous works [31, 38, 56]. Contour consistency is representative of the three previous objectives. If contours are interpolated consistently, then so too are the edges (Objectives 2 and 3). If no new contour lines are created, then constants are reproduced, which satisfies Objective 1 for constant-order smoothness.

A contour consistent interpolation method can be composed from two steps: first estimating the contours, then interpolating according to the detected contours. These steps are quite modular; any estimate of the local image structure for the first step can be used to guide any method that understands this information for the second step. As shown in [22, 23], a motivation for why contour stencils are a good choice for the first step is the following connection between TV and interpolation error.

Theorem 3.1. *Consider approximating $u(x)$ by*

$$\hat{u}(x) = (1-\lambda)u(a) + \lambda u(b) \tag{3.2}$$



and let $C = C_1 \cup C_2$ be a curve passing through a, x, and b. Then the approximation error is bounded by

$$|\hat{u}(x) - u(x)| \le \max\bigl\{|1-\lambda|\,,|\lambda|\bigr\}\|u\|_{\mathrm{TV}(C)}. \tag{3.3}$$

Proof. Let $\gamma : [0,T] \to C_1$ parameterize curve C_1; then

$$\|u\|_{\mathrm{TV}(C_1)} := \sup_{\substack{(t_n)\,:\,t_n \in [0,T],\\ t_n < t_{n+1}}} \sum_n \bigl|u\bigl(\gamma(t_{n+1})\bigr) - u\bigl(\gamma(t_n)\bigr)\bigr| \ge |u(a) - u(x)|\,, \tag{3.4}$$

and similarly $\|u\|_{\mathrm{TV}(C_2)} \geq |u(b) - u(x)|$. Together these bounds yield the result. ■

The theorem justifies the *interpolate along edges* idea of edge directed interpolation: the closer C is to an image contour, the smaller the error bound $\|u\|_{\mathrm{TV}(C)}$. Furthermore, if the interpolation curve C is chosen according to the best-fitting stencil (2.23), then the estimated bound on the interpolation error is minimized:[3]

$$|\hat{v}(x) - v(x)| \leq \|u\|_{\mathrm{TV}(C)} \approx V(\mathcal{S}^\star, v) = \min_{\mathcal{S}\in\Sigma} V(\mathcal{S}, v). \tag{3.5}$$

Thus contour stencils are well suited for edge directed interpolation.

3.2. Interpolation with contour stencils.

In this section, we extend the interpolation method described in [24] to contour stencils of curves of any shape.

The interpolant is constructed as a function in a manner similar to classic B-spline interpolation [29] or kernel regression [46, 52, 53]. This enables interpolation by arbitrary scale factors and other resampling operations like parallax correction. The method incorporates the PSF from the discretization model (3.1), but for efficiency, the discretization model is satisfied only approximately: sample$(h * u) \approx v$. This is key since satisfying the discretization model exactly is a global problem, while it can be well approximated locally.

Windowed reconstruction. We begin by forming local reconstructions

$$u_k(x) = v_k + \sum_{n\in\mathcal{N}} c_n\, \rho^n_{\mathcal{S}^\star(k)}(x - n), \tag{3.6}$$

where $\mathcal{N} \subset \mathbb{Z}^2$ is a neighborhood of the origin and $\rho^n_{\mathcal{S}}$ is a function oriented in the direction

$$\theta^n_{\mathcal{S}} = \angle \int_{[-\frac{1}{2},+\frac{1}{2}]^2} \nabla\varphi^\perp_{\mathcal{S}}(x - n)\, dx \tag{3.7}$$

and with anisotropy

$$\mu_{\mathcal{S}} = \min_{n\in\mathcal{N}} \left| \int_{[-\frac{1}{2},+\frac{1}{2}]^2} \nabla\varphi_{\mathcal{S}}(x - n)\, dx \right| \tag{3.8}$$

so that $\rho^n_{\mathcal{S}^\star(k)}$ is oriented with the contour modeled by the best-fitting stencil $\mathcal{S}^\star(k)$ at point $k + n$ (see Figure 5). The larger the anisotropy parameter $\mu_{\mathcal{S}}$, the more strongly $\rho^n_{\mathcal{S}}$ points in the direction $\theta^n_{\mathcal{S}}$. For the circle stencil, $\mu_{\mathcal{S}}$ is set to zero to favor no direction.

The c_n are chosen such that u_k satisfies the discretization model locally, $(h{*}u_k)(m) = v_{k+m}$ for $m \in \mathcal{N}$, which implies

$$\sum_{n\in\mathcal{N}} (A_{\mathcal{S}^\star(k)})_{m,n}\, c_n = v_{k+m} - v_k, \quad m \in \mathcal{N}, \tag{3.9}$$

where $A_{\mathcal{S}}$ is an $|\mathcal{N}| \times |\mathcal{N}|$ matrix with elements $(A_{\mathcal{S}})_{m,n} = (h * \rho^n_{\mathcal{S}})(m - n)$. Similar to the local approximation approach in kernel regression, the idea here is that $u_k(x)$ is an accurate interpolation for x near zero.

[3] It is possible that the actual error is larger, however, since $V(\mathcal{S}^\star, v)$ is an *estimated* error bound.

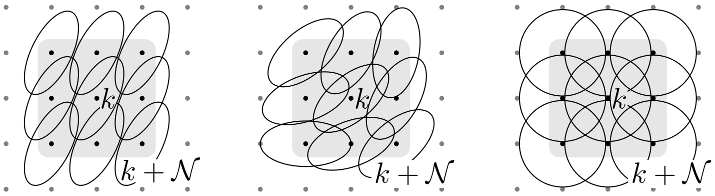


Figure 5. *Local reconstruction u_k with oriented functions $\rho^n_{\mathcal{S}}$. Left: Reconstruction with a line-shaped stencil. Center: With a parabola-shaped stencil. Right: With the circle stencil.*

The local reconstructions are combined with overlapping windows as

$$u(x) = \sum_{k \in \mathbb{Z}^2} w(x-k) u_k(x-k), \tag{3.10}$$

where w is a function with compact support such that $\sum_k w(x-k) = 1$ for $x \in \mathbb{R}^2$ and $w(k) = 0$ for $k \in \mathbb{Z}^2 \backslash \mathcal{N}$. For computation, it is useful to define

$$\tilde{\rho}^n_{\mathcal{S}}(x) = w(x) \sum_{m \in \mathcal{N}} (A^{-1}_{\mathcal{S}})_{m,n} \, \rho^m_{\mathcal{S}}(x-m). \tag{3.11}$$

Then the interpolation is expressed directly in terms of v as

$$u(x) = \sum_{k \in \mathbb{Z}^2} \left[w(x-k) v_k + \sum_{n \in \mathcal{N} \backslash \{0\}} (v_{k+n} - v_k) \, \tilde{\rho}^n_{\mathcal{S}^\star(k)}(x-k) \right]. \tag{3.12}$$

The functions $\tilde{\rho}^0_{\mathcal{S}}$ are unused since $(v_{k+n} - v_k) = 0$ when $n = 0$. The proposition shows that this interpolation satisfies some basic properties.

Proposition 3.2. *The reconstruction* (3.12) *satisfies the following properties:*

(i) *If v is constant, then $u = v$.*

(ii) *If $w, \rho^n_{\mathcal{S}} \in C^k$, then $u \in C^k$.*

Proof. The first result follows from (3.12) and $\sum_k w(x-k) = 1$. The second result follows from

$$u(x) = \sum_{k \in \mathbb{Z}^2} w(x-k) \left[v_k + \sum_{n \in \mathcal{N}} c_n \, \rho^n_{\mathcal{S}^\star(k)}(x-k-n) \right]$$

and the fact that w has compact support. ■

For general h, reconstruction (3.12) satisfies the discretization model approximately, i.e., sample$(h * u) \approx v$. The source of this discrepancy is essentially the noncommutativity of multiplication and convolution,

$$h * (w \cdot u_k) \neq w \cdot (h * u_k). \tag{3.13}$$

However, if h is well localized compared to w, the two sides are approximately equal. The following theorem shows how $\|v - \text{sample}(h * u)\|_\infty$ relates to h and w.

Theorem 3.3. *The residual in satisfying the discretization model is bounded as*

$$\|v - \operatorname{sample}(h * u)\|_\infty \le K \sup_n \|u_n\|_\infty \,, \tag{3.14}$$

$$K = \sum_{k \in \mathbb{Z}^2} \int_{\mathbb{R}^2} |h(x)| \, |w(k) - w(k-x)| \, dx.$$

In particular, if $h = \delta$, then $K = 0$.

Proof. Observe that since $w(k) = 0$ for $k \notin \mathcal{N}$ and $(h * u_{m-k})(k) = v_m$ for $k \in \mathcal{N}$,

$$v_m = \sum_{k \in \mathbb{Z}^2} w(k) \, (h * u_{m-k})(k). \tag{3.15}$$

Therefore, the difference can be bounded as

$$\begin{aligned} \big|v_m - (h * u)(m)\big| &= \left| \sum_{k \in \mathbb{Z}^2} \int_{\mathbb{R}^2} h(x) \big[w(k) - w(k-x)\big] u_{m-k}(k-x) \, dx \right| \\ &\le K \sup_n \|u_n\|_\infty \,. \qquad \blacksquare \end{aligned}$$

Iterative refinement. Given an approximate right inverse B, $AB \approx I$, iterative refinement finds a solution of the linear system $Ax = y$ by the iteration

$$y^0 = y, \quad y^{i+1} = y^i + (y - ABy^i). \tag{3.16}$$

Provided $(I - AB)$ is a contraction, $x = \lim_{i \to \infty} By^i$ satisfies $Ax = y$.

Iterative refinement may be applied to improve the accuracy in satisfying the discretization model $v = \operatorname{sample}(h * u^i)$. Define the linear operator $\mathcal{R}$ by (3.12) such that $u = \mathcal{R}v$ (considering $\mathcal{S}^\star$ as fixed parameters), and iterate

$$u^0 = 0, \quad u^{i+1} = u^i + \mathcal{R}\big(v - \operatorname{sample}(h * u^i)\big). \tag{3.17}$$

In implementation, each interpolation $\mathcal{R}\big(v - \operatorname{sample}(h * u^i)\big)$ is evaluated on a fine grid and the convolution $h * u^i$ is approximated by discrete convolution.

By the linearity of $\mathcal{R}$, an equivalent but computationally useful iteration is

$$v^0 = v, \quad \begin{cases} u^{i+1} = \mathcal{R}v^i, \\ v^{i+1} = v^i + \big(v - \operatorname{sample}(h * \mathcal{R}v^i)\big). \end{cases} \tag{3.18}$$

The interpolation and convolution $h * \mathcal{R}v^i$ is evaluated discretely on a grid. The iteration prefilters the input v such that $u = \mathcal{R}v^i$ accurately approximates $v = \operatorname{sample}(h * u)$.

Compared to (3.17), the advantage of (3.18) is that the final interpolation $u = \mathcal{R}v^i$ may be evaluated directly at any point rather than as an accumulation of intermediate interpolations. Thus the final interpolation may be evaluated at points other than the grid used to compute $h * \mathcal{R}v^i$ during prefiltering. This flexibility is crucial if the final interpolation is evaluated at arbitrary locations. Additionally, for image scaling by large scale factors, prefiltering on a coarser grid than the grid used in the final interpolation is computationally efficient with only a small loss in quality. In the examples, prefiltering is performed at twice the resolution of the input image.

Remark 3.1. If there exists $C < 1$ such that for any $v \in \ell^\infty$ and k

$$|v_k - (h * \mathcal{R}v)(k)| \le C\,|v_k|\,, \tag{3.19}$$

then v^i converges in ℓ^∞ and u^i converges uniformly to u that satisfies $\operatorname{sample}(h * u) = v$ exactly. Unfortunately, we have no proof of a practically useful condition that ensures $C < 1$. In numerical experiments, it appears the iteration reliably converges when using the suggested parameters and h with the standard deviation less than 0.8.

The proposed interpolation has the same computational complexity as the method in [24]. Let W bound the number of nonzero terms,

$$\#\{k \in \mathbb{Z}^2 : w(x-k) \neq 0\} \le W \quad \text{for all } x \in \mathbb{R}^2. \tag{3.20}$$

Then supposing that W is $\mathcal{O}(|\mathcal{N}|)$, the computational cost of factor-d scaling with $\mathcal{R}$ by formulas (3.6) and (3.10) is $\mathcal{O}(|\mathcal{N}|^2\, d^2)$ operations per input pixel.

For scaling by rational d, samples of w and $\tilde\rho^n_{\mathcal{S}}$ can also be precomputed, and scaling by formula (3.12) costs $6\,|\mathcal{N}|\,Wd^2$ operations per input pixel. For the settings used in the examples, this is $864d^2$ operations per input pixel.

Parameters. In the experiments, w is the cubic B-spline

$$w(x) = B(x_1)B(x_2), \quad B(t) = \big(1 - |t| + \tfrac{1}{6}\,|t|^3 - \tfrac{1}{3}\big|1-|t|\big|^3\big)^+, \tag{3.21}$$

and the neighborhood is $\mathcal{N} = \{-1,0,1\} \times \{-1,0,1\}$, for which $W = 16$. For $\rho^n_{\mathcal{S}}$, we use an oriented Gaussian

$$\rho^n_{\mathcal{S}}(x) = \exp\big(-\tau^2/(2\sigma_\tau^2) - \nu^2/(2\sigma_\nu^2)\big), \quad \begin{pmatrix}\tau\\ \nu\end{pmatrix} = \begin{pmatrix}\cos\theta & \sin\theta\\ -\sin\theta & \cos\theta\end{pmatrix} x, \tag{3.22}$$

where $\sigma_\tau = 1.2$ and $\sigma_\nu = \sigma_\tau(1 - \mu^4/2)$, with θ and μ determined by (3.7) and (3.8).

In numerical experiments, the iteration (3.18) converges quickly: typically three or four iterations are sufficient for accurate deconvolution. Figure 6 shows an example initial interpolation u^1 and the interpolation after two refinement passes u^3. Tables 3 and 4 show the residual norms $\|r^i\|_\infty$ where the image intensity range is $[0,1]$. The interpolation achieves sharp results comparable with exact deconvolution while maintaining computational efficiency.

Figure 7 demonstrates contour stencil windowed interpolation on textured images. Such images are difficult because they have very fine scale structure. Contour stencils enable the method to approximate oriented textures like hair and fabric where the contours flow smoothly together in mostly the same direction. The method is less successful on rough textures like asphalt (rightmost image).

3.3. Comparison. This section compares the proposed interpolation method with several existing methods. We compare it with Muresan's AQua-2 edge directed interpolation [41], TV minimization [36], and the tensor-driven diffusion method of Roussos and Maragos [49]. All

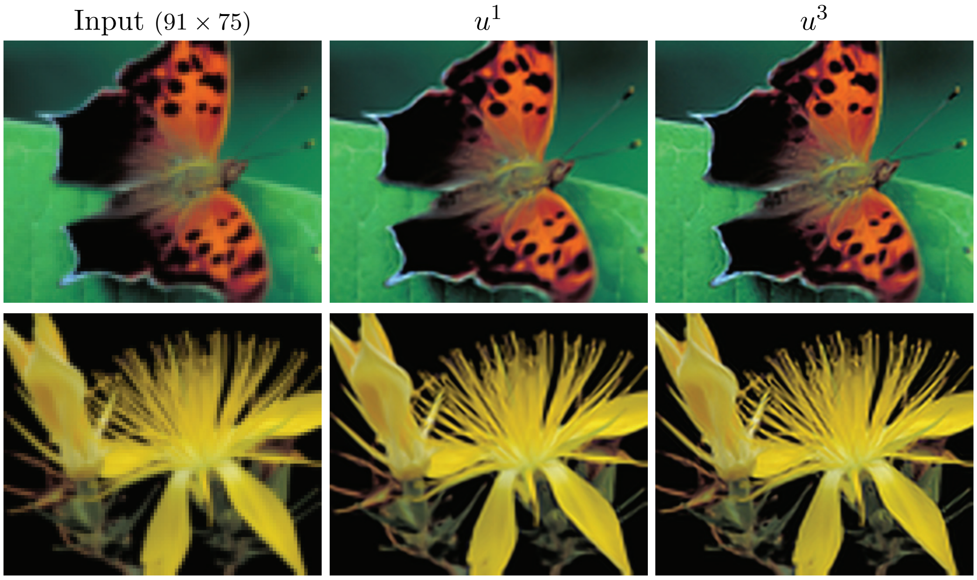


Figure 6. *Factor-4 windowed interpolation with iterative refinement (3.18). Original photos by Thomas G. Barnes, U.S. Fish and Wildlife Service.*

Table 3
Residual reduction in Figure 6, top row.

Iteration i	Residual norm $\|r^i\|_\infty$
1	0.05853
2	0.01779
3	0.00908
4	0.00515
5	0.00307

Table 4
Residual reduction in Figure 6, bottom row.

Iteration i	Residual norm $\|r^i\|_\infty$
1	0.06601
2	0.02314
3	0.01241
4	0.00789
5	0.00499

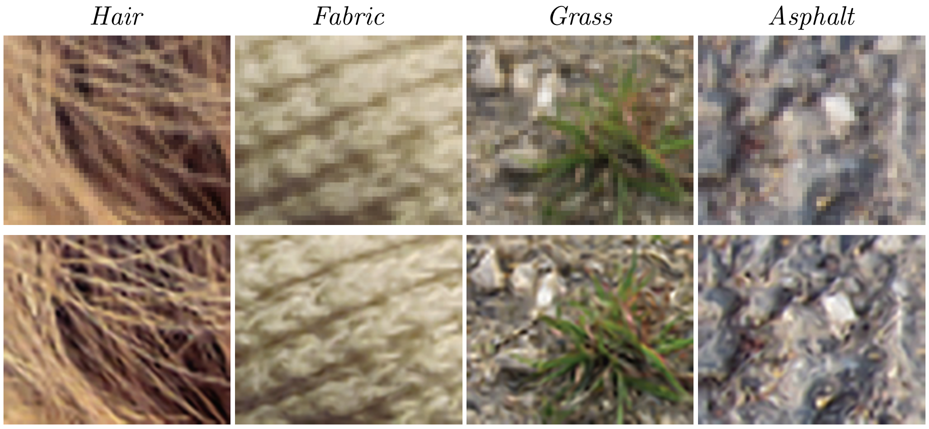
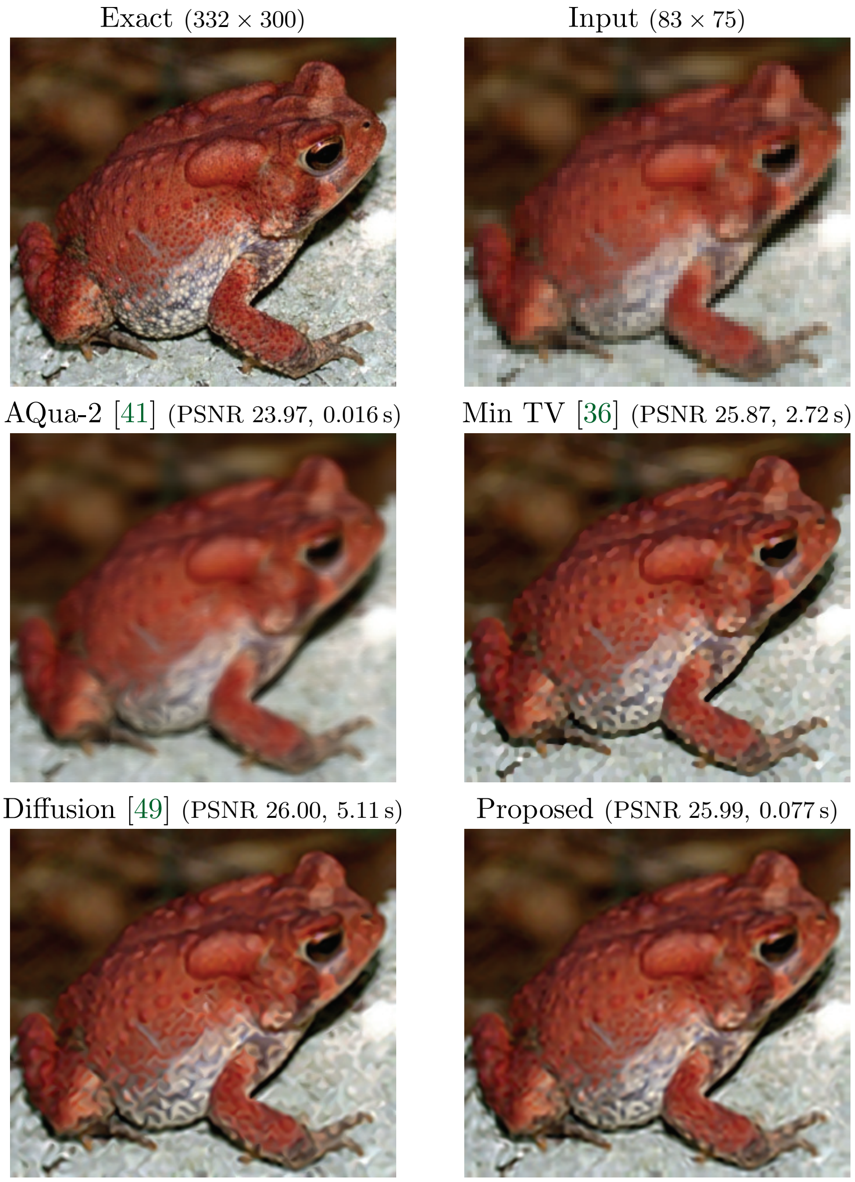


Figure 7. *Interpolation of textures. Top: Input images. Bottom: Factor-4 interpolation with (3.18).*



Exact (332 × 300)

Input (83 × 75)

AQua-2 [41] (PSNR 23.97, 0.016 s)

Min TV [36] (PSNR 25.87, 2.72 s)

Diffusion [49] (PSNR 26.00, 5.11 s)

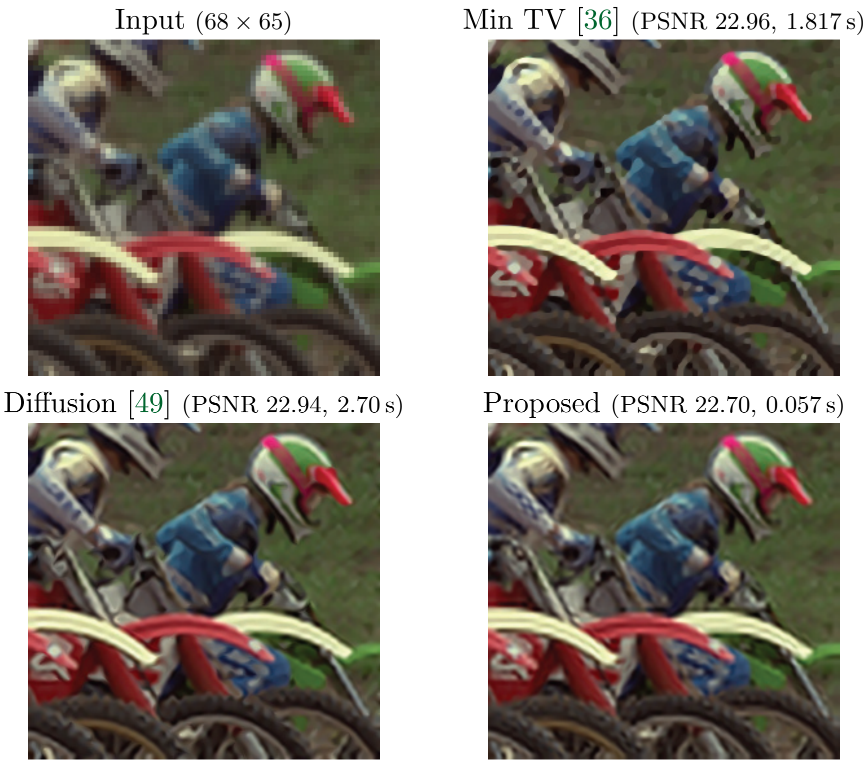
Proposed (PSNR 25.99, 0.077 s)

Figure 8. *Factor-4 color interpolation. Original photo by John D. Willson, Amphibian Research and Monitoring Initiative, U.S. Geological Survey.*

methods were implemented in C so that computation times could be meaningfully compared.[4] The TV minimization was efficiently implemented using the split-Bregman algorithm [26]. For the diffusion method, $\operatorname{div}(T \operatorname{grad} u)$ was discretized using the fast 5×5 explicit scheme proposed in [60].

In Figures 8 and 9, we perform the following experiment. The image is first coarsened

[4] Computations were timed on a 2.40GHz Intel Core 2 Duo T7700 with 2GB RAM. Timing was not possible for [43] since its software is closed-source.



Input (68 × 65)

Min TV [36] (PSNR 22.96, 1.817 s)

Diffusion [49] (PSNR 22.94, 2.70 s)

Proposed (PSNR 22.70, 0.057 s)

Figure 9. *Comparison on a crop from image #5 of the Kodak Image Suite.*

by convolution with h and factor-4 downsampling, $v = \downarrow(h * u_o)$. The coarsened image is interpolated by factor 4 to find u such that $v = \downarrow(h * u)$. The interpolation u is compared to the original u_o with the peak signal-to-noise ratio (PSNR) metric, and the time needed to compute the interpolation is also shown. The PSF h is a Gaussian with standard deviation of 0.5 in units of pixels at the coarse scale (2.0 at the fine scale).

Bilinear and AQua-2 are defined only for point-value interpolation, which finds u such that $v = \downarrow(u)$. So for these methods, interpolation is done this way, and the experiment is otherwise the same.

Kodak Image Suite. Table 5 shows the average performance on the Kodak Lossless True Color Image Suite, a set of 24 natural color images.[5] The average image quality is also shown in terms of the mean structural similarity (MSSIM) metric introduced by Wang et al. [57]. Figure 9 shows a crop from image #5.

Bilinear and AQua-2 are computationally simple and are the only methods in the comparison that are point-value interpolating. This explains their good computational speed but relatively low quality.

At the other extreme, the tensor-driven diffusion method seems to have the best visual quality and also performs best in the quantitative comparison. Computationally, however, the method is dominated by six FFTs per iteration, which are used for the orthogonal projection of the diffusion.

[5] Available online at http://www.cipr.rpi.edu/resource/stills/kodak.html.

Table 5

Average PSNR, MSSIM, and CPU times on the Kodak Image Suite.

Method	PSNR	MSSIM	CPU time (s)
Bilinear	24.71	0.6443	0.001
AQua-2 [41]	24.77	0.6445	0.056
Fourier	25.87	0.6853	0.308
Previous method [24]	25.97	0.7069	0.398
Proposed method	**26.07**	**0.7112**	**0.293**
Min TV [36]	26.32	0.7189	12.778
Diffusion [49]	26.37	0.7238	27.214

The TV minimization method finds the interpolation u as the solution of

$$\operatorname*{arg\,min}_{u \in BV} \|u\|_{\mathrm{TV}} \quad \text{subject to} \quad v = \operatorname{sample}(h * u). \tag{3.23}$$

The method performs well quantitatively in PSNR and MSSIM and is about twice as fast as the tensor-driven diffusion method. Due to the staircasing artifact with TV regularization [11], the method has the downside that results have a piecewise constant cartoon-like look, which is not a desirable effect for natural images.

The proposed interpolation method using contour stencils has a balance between computational cost and image quality. The visual and quantitative quality is comparable with the diffusion method, yet the computation is nearly 100 times faster. The table shows that the computation time of the proposed method is within an order of magnitude of AQua-2.

3.4. Pixel art scaling.

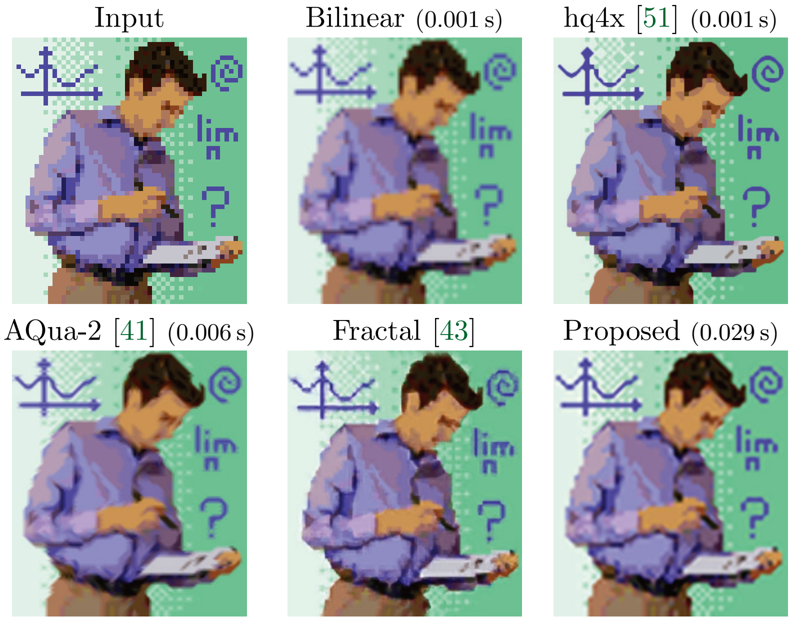
While the focus of this work is the interpolation of natural images, edge-adaptive methods are also needed for scaling pixel art images. Pixel art images are edited at the pixel level and are characterized by crisp edges and limited number of colors. Pixel art appears in older video games and mobile devices.

Figure 10 shows an example pixel art image scaled by factor 4 using standard bilinear interpolation, Stepin's hq4x method [51], AQua-2 [41], and fractal zooming [43]. In pixel art, the pixel values are more accurately modeled as point-value samples than as weighted averages as in (3.1), so the proposed method is applied with PSF $h = \delta$. The TV minimization and tensor-driven diffusion methods are not shown since they are not intended for point-value interpolation.

Pixel art scaling with linear methods is usually overly blurry and with strong staircase artifacts, as shown in the result with bilinear interpolation. hq4x is a method specifically designed for pixel art scaling. It performs very well on the cartoon-like features in the background, but it misses edges (like the subject's right elbow) if there is significant color variation along the edge. AQua-2, fractal zooming, and the proposed method perform reasonably well because they have edge-adaptive strategies, though they are blurrier than hq4x. Notice that none of the methods appropriately handles the dithered gradient in the background.

4. Conclusions.

Contour stencils estimate the image contours by approximating total variation (TV) along curves. They are derived as a consistent discretization of an integral (section 2.3) and can be designed to distinguish different types geometric features, including lines, curves, corners, and circles (section 2.5).



Input — Bilinear (0.001 s) — hq4x [51] (0.001 s)

AQua-2 [41] (0.006 s) — Fractal [43] — Proposed (0.029 s)

Figure 10. *Pixel art scaling comparison by factor* 4.

Interpolation error can be bounded in terms of TV along curves (Theorem 3.1), which implies that edge directed interpolation with contour stencils minimizes an estimated error bound. The proposed interpolation method has image quality comparable to existing high-quality methods yet with significantly lower computational cost (section 3.3). The method is also reasonable though somewhat blurry for pixel art scaling (section 3.4).

Contour stencils may be useful in other kinds of imaging problems that involve estimating the image structure, for example, color demosaicing and joint interpolation-denoising. Future work will explore these directions.

Acknowledgments. I thank Prof. L. Vese and Prof. J. M. Morel for guidance and useful suggestions on this work.

REFERENCES

[1] V. R. Algazi, G. E. Ford, and R. A. Potharlanka, *Directional interpolating of images based on visual properties and rank order filtering*, in Proceedings of the 1991 IEEE International Conference on Acoustics, Speech, and Signal Processing, 1991, pp. 3005–3008.

[2] J. P. Allebach and P. W. Wong, *Edge-directed interpolation*, in Proceedings of the 1996 IEEE International Conference on Image Processing, Vol. 3, 1996, pp. 707–710.

[3] P. A. Arbeláez and L. D. Cohen, *Energy partitions and image segmentation*, J. Math. Imaging Vision, 20 (2004), pp. 43–57.

[4] S. P. Awate and R. T. Whitaker, *Unsupervised, information-theoretic, adaptive image filtering for image restoration*, IEEE Trans. Pattern Anal. Mach. Intell., 28 (2006), pp. 364–376.

[5] D. H. Ballard and C. M. Brown, *Computer Vision*, Prentice–Hall, Englewood Cliffs, NJ, 1982.

[6] C. Ballester, V. Caselles, L. Igual, J. Verdera, and B. Rougé, *A variational model for $P+XS$ image fusion*, Int. J. Comput. Vision, 69 (2006), pp. 43–58.

[7] A. Belahmidi and F. Guichard, *A partial differential equation approach to image zoom*, in Proceedings of the 2004 IEEE International Conference on Image Processing, 2004, pp. 649–652.

[8] M. Bertalmío, G. Sapiro, V. Caselles, and C. Ballester, *Image inpainting*, in Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '00, ACM Press, Addison-Wesley, New York, 2000, pp. 417–424.

[9] J. Bigün and G. H. Granlund, *Optimal orientation detection of linear symmetry*, in Proceedings of the First IEEE International Conference on Computer Vision, London, 1987, pp. 433–438.

[10] A. Buades, B. Coll, and J. M. Morel, *A review of image denoising algorithms, with a new one*, Multiscale Model. Simul., 4 (2005), pp. 490–530.

[11] T. F. Chan, S. Esedoglu, F. Park, and A. M. Yip, *Total variation image restoration: Overview and recent developments*, in Handbook of Mathematical Models in Computer Vision, Springer, New York, 2006, pp. 17–31.

[12] T. F. Chan and J. Shen, *Mathematical models for local nontexture inpaintings*, SIAM J. Appl. Math., 62 (2002), pp. 1019–1043.

[13] G. Cottafava and G. L. Moli, *Automatic contour map*, Comm. ACM, 12 (1969), pp. 386–391.

[14] N. Dyn, D. Levin, and S. Rippa, *Data dependent triangulations for piecewise linear interpolation*, IMA J. Numer. Anal., 10 (1990), pp. 137–154.

[15] A. Efros and T. Leung, *Texture synthesis by non-parametric sampling*, in Proceedings of the IEEE International Conference on Computer Vision, 1999, pp. 1033–1038.

[16] G. Facciolo, P. Arias, V. Caselles, and G. Sapiro, *Exemplar-based interpolation of sparsely sampled images*, in Energy Minimization Methods in Computer Vision and Pattern Recognition, D. Cremers, Y. Boykov, A. Blake, and F. R. Schmidt, eds., Lecture Notes in Comput. Sci. 5681, Springer, Berlin, Heidelberg, 2009, pp. 331–344.

[17] H. Federer, *Curvature measures*, Trans. Amer. Math. Soc., 93 (1959), pp. 418–491.

[18] Y. Fisher, *Fractal Image Compression: Theory and Application to Digital Images*, Springer-Verlag, New York, 1995.

[19] W. Förstner and E. Gülch, *A fast operator for detection and precise location of distinct points, corners, and centers of circular features*, in Proceedings of the Intercommission Conference on Fast Processing of Photogrammetric Data, Interlaken, Switzerland, 1987, pp. 281–305.

[20] W. T. Freeman, T. R. Jones, and E. C. Pasztor, *Example-based super-resolution*, IEEE Comput. Graph. Appl., 22 (2002), pp. 56–65.

[21] P. Getreuer, *Image interpolation with geometric contour stencils*, Image Processing On Line, 2011, DOI:10.5201/ipol.2011.g_igcs.

[22] P. Getreuer, *Contour stencils for edge-adaptive image interpolation*, in Proceedings of the SPIE, Vol. 7257, 2009, 725718.

[23] P. Getreuer, *Image zooming with contour stencils*, in Proceedings of the SPIE, Vol. 7246, 2009, 72460P.

[24] P. Getreuer, *Image interpolation with contour stencils*, Image Processing On Line, 2011, DOI:10.5201/ipol.2011.g_iics.

[25] D. Glasner, S. Bagon, and M. Irani, *Super-resolution from a single image*, in Proceedings of the 12th IEEE International Conference on Computer Vision, 2009.

[26] T. Goldstein and S. Osher, *The split Bregman method for L1-regularized problems*, SIAM J. Imaging Sci., 2 (2009), pp. 323–343.

[27] A. Goshtasby, S. H. Gage, and J. F. Bartholic, *A two-stage cross-correlation approach to template matching*, IEEE Trans. Pattern Anal. Mach. Intell., 6 (1984), pp. 374–378.

[28] Y.-S. Ho and A. Gersho, *Variable-rate contour-based interpolative vector quantization for image coding*, in Proceedings of the 1988 IEEE Global Telecommunications Conference, 1988, pp. 750–754.

[29] H. Hou and H. Andrews, *Cubic splines for image interpolation and digital filtering*, IEEE Trans. Acoust. Speech Signal Process., 26 (1978), pp. 508–517.

[30] K. Jensen and D. Anastassiou, *Subpixel edge localization and the interpolation of still images*, IEEE Trans. Image Process., 4 (1995), pp. 285–295.

[31] H. Jiang and C. Moloney, *A new direction adaptive scheme for image interpolation*, in Proceedings of the IEEE International Conference on Image Processing, Vol. 3, 2002, pp. 369–372.

[32] C. Jordan, *Sur la séie de fourier*, C. R. Acad. Sci. Paris, 92 (1881), pp. 228–230.

[33] F. Kikuchi and X. Liu, *Estimation of interpolation error constants for the P_0 and P_1 triangular finite elements*, Comput. Methods Appl. Mech. Engrg., 196 (2007), pp. 3750–3758.

[34] X. Li and M. T. Orchard, *New edge-directed interpolation*, IEEE Trans. Image Process., 10 (2001), pp. 1521–1527.

[35] H. Q. Luong, A. Ledda, and W. Philips, *Non-local interpolation*, in Proceedings of the IEEE International Conference on Image Processing, 2006, pp. 693–696.

[36] F. Malgouyres and F. Guichard, *Edge direction preserving image zooming: A mathematical and numerical analysis*, SIAM J. Numer. Anal., 39 (2001), pp. 1–37.

[37] S. Mallat and G. Yu, *Super-resolution with sparse mixing estimators*, IEEE Trans. Image Process., 19 (2010), pp. 2889–2900.

[38] S. Masnou and J.-M. Morel, *Level lines based disocclusion*, in Proceedings of the IEEE International Conference on Image Processing, Vol. 3, 1998, pp. 259–263.

[39] E. Meijering, *A chronology of interpolation: From ancient astronomy to modern signal and image processing*, Proc. IEEE, 90 (2002), pp. 319–342.

[40] M. Moeller, T. Wittman, and A. L. Bertozzi, *A variational approach to hyperspectral image fusion*, in Proceedings of the SPIE, Vol. 7334, 2009.

[41] D. D. Muresan, *Fast edge directed polynomial interpolation*, in Proceedings of the IEEE International Conference on Image Processing, Vol. 2, 2005, pp. 990–993.

[42] I. P. Natanson, *Theory of Functions of a Real Variable*, Frederick Ungar, New York, 1955.

[43] onOne Software, *Genuine Fractals*, http://www.ononesoftware.com (2008).

[44] L. E. Payne and H. F. Weinberger, *An optimal Poincaré inequality for convex domains*, Arch. Rational Mech. Anal., 5 (1960), pp. 286–292.

[45] G. Peyré, S. Bougleux, and L. D. Cohen, *Non-local regularization of inverse problems*, Inverse Probl. Imaging, 5 (2011), pp. 511–530.

[46] T. Q. Pham, L. J. van Vliet, and K. Schutte, *Robust fusion of irregularly sampled data using adaptive normalized convolution*, EURASIP J. Appl. Signal Process., 2006 (2006), 83268.

[47] M. Protter, M. Elad, H. Takeda, and P. Milanfar, *Generalizing the nonlocal-means to super-resolution reconstruction*, IEEE Trans. Image Process., 18 (2009), pp. 36–51.

[48] T. Riemersma, *Quick Image Scaling by* 2, http://www.compuphase.com/graphic/scale2.htm (2006).

[49] A. Roussos and P. Maragos, *Reversible interpolation of vectorial images by an anisotropic diffusion-projection PDE*, Int. J. Comput. Vision, 84 (2009), pp. 130–145.

[50] M. Sajjad, N. Khattak, and N. Jafri, *Image magnification using adaptive interpolation by pixel level data-dependent geometrical shapes*, in Proceedings of World Academy of Science, Engineering and Technology, Vol. 25, 2007.

[51] M. Stepin, *hq4x Magnification Filter*, http://www.hiend3d.com/hq4x.html (2003).

[52] H. Takeda, S. Farsiu, and P. Milanfar, *Kernel regression for image processing and reconstruction*, IEEE Trans. Image Process., 16 (2007), pp. 349–366.

[53] H. Takeda, S. Farsiu, and P. Milanfar, *Deblurring using regularized locally adaptive kernel regression*, IEEE Trans. Image Process., 17 (2008), pp. 550–563.

[54] P. Thévenaz, T. Blu, and M. Unser, *Interpolation revisited*, IEEE Trans. Med. Imaging, 19 (2000), pp. 739–758.

[55] D. Tschumperlé and R. Deriche, *Vector-valued image regularization with PDEs: A common framework for different applications*, IEEE Trans. Pattern Anal. Mach. Intell., 27 (2005), pp. 506–517.

[56] Q. Wang and R. K. Ward, *A new orientation-adaptive interpolation method*, IEEE Trans. Image Process., 16 (2007), pp. 889–900.

[57] Z. Wang, A. C. Bovik, H. R. Sheikh, S. Member, and E. P. Simoncelli, *Image quality assessment: From error measurement to structural similarity*, IEEE Trans. Image Process., 13 (2004), pp. 600–612.

[58] J. Weickert, *Theoretical foundations of anisotropic diffusion in image processing*, Comput. Suppl., 11 (1996), pp. 221–236.

[59] J. Weickert, *Anisotropic Diffusion in Image Processing*, ECMI Ser., Teubner-Verlag, Stuttgart, Germany, 1998.

[60] J. Weickert and H. Scharr, *A scheme for coherence-enhancing diffusion filtering with optimized rotation invariance*, J. Vis. Commun. Image Rep., 13 (2002), pp. 103–118.

[61] L. Yatziv and G. Sapiro, *Fast image and video colorization using chrominance blending*, IEEE Trans. Image Process., 15 (2006), pp. 1120–1129.

[62] X. Yu, B. S. Morse, and T. W. Sederberg, *Image reconstruction using data-dependent triangulation*, IEEE Comput. Graph. Appl., 21 (2001), pp. 62–69.