Image zooming with contour stencils

Pascal Getreuer

University of California Los Angeles, Mathematics Department, U.S.A.

ABSTRACT

We introduce "contour stencils" as a simple method for detecting the local orientation of image contours and apply this detection to image zooming. Our approach is motivated by the total variation along curves: small total variation along a candidate curve suggests that this curve is a good approximation to the contours. Furthermore, a relationship is shown between interpolation error and total variation. The contour stencil detection is used to develop two image zooming methods. The first one, "contour stencil interpolation," is simple and computationally efficient, yet competitive in a comparison against existing methods. The second method approaches zooming as an inverse problem, using a graph regularization where the graph is determined by contour stencil detection. Both methods extend naturally to vector-valued data and are demonstrated for grayscale and color images.

Keywords: Interpolation, contour stencils, total variation

1. INTRODUCTION

Image zooming (also called *image scaling* and *single-frame super-resolution*) is the problem of increasing the resolution of a given image to higher resolution. *Image interpolation* refers to the special case when the process is reversible by subsampling.

Three important objectives in image zooming are smoothness order, step response, and directional selectivity. Traditional interpolation methods focus on *smoothness order*, the highest polynomial order the method can reproduce. In spite of smoothness, the method's behavior near an edge, the *step response*, should have minimal blurring and minimal overshoots. The method's *directional selectivity* is the number of directions it can distinguish, which enables it to handle edges of different angles without producing jagged artifacts. And, contrary to all other objectives, the method should be simple and computationally efficient.

1.1 Recent Work

Recent research shows many ideas on how to approach the challenging problem of image zooming. We briefly describe three prominent categories: PDE-based, patch-based, and edge-directed methods.

1.1.1 PDE-based

PDE-based methods can be further categorized as derived from a minimization model or designed directly as a diffusion process. Total variation (TV) regularization, introduced by Rudin, Osher, and Fatemi,[1] has been successful in many image processing problems. The technique is to minimize the total variation $\|u\|_{\mathrm{TV}} := \int_\Omega |\nabla u| \, dx$ with a constraint according to the goal of the problem. Malgouyres and Guichard[2] posed image zooming as

$$\min_{u \in BV(\Omega)} \|u\|_{\mathrm{TV}} \quad \text{subject to } \operatorname{sample}(h * u) = v.$$

Similarly, Chan and Shen[3] applied TV regularization to image inpainting, where interpolation is a special case.

Applying gradient descent to a minimization model yields a corresponding diffusion process. Gradient descent on $\|u\|_{\mathrm{TV}}$ yields the "TV-flow" anisotropic diffusion $\partial_t u = \operatorname{div}(\nabla u / |\nabla u|)$. Rather than beginning with a minimization, some methods are designed directly as a diffusion process. Not every diffusion process corresponds to a minimization; this setting is more general. For image zooming, several proposed diffusions are

Bertalmío, Bertozzi, and Sapiro[4]	$\partial_t u = \nabla u^\perp \cdot \nabla \Delta u + \operatorname{div}\big(g(|\nabla u|)\nabla u\big)$
Belahmidi and Guichard[5]	$\partial_t u = \frac{\nabla u^\perp}{|\nabla u|} \cdot \nabla^2 u \cdot \frac{\nabla u^\perp}{|\nabla u|} + g(|\nabla u|) \frac{\nabla u}{|\nabla u|} \cdot \nabla^2 u \cdot \frac{\nabla u}{|\nabla u|}$
Roussos and Maragos[6]	$\partial_t u = \operatorname{div}\big(T\big(J_\rho(\nabla u_\sigma)\big)\nabla u\big)$

where in the first and second methods, g is an edge-stopping function, and in the third method, $T\big(J_\rho(\nabla u_\sigma)\big)$ is a tensor constructed from the eigenvalues and eigenvectors of the 2×2 structure tensor.

1.1.2 Patch-based

The nonlocal means method of Buades et al.[7] has inspired several methods.[8–10] These methods use the nonlocal means weights to identify similar patches in the image and combine them to estimate a high resolution patch.

Freeman et al.[11] and similarly other "example-based" methods[12,13] performed image zooming using a large database of image patches of low resolution and high resolution pairs. For each patch in a given low resolution image, they search the database for the corresponding high resolution patch.

Fractal image zooming is a patch-based technique derived from fractal image compression, where an image is compactly represented by identifying its self-similar structures.[14,15] The given image is first encoded as an iterated function system (IFS), a contraction that maps patches to similar patches such that the image is the fixed point of the IFS. The IFS is "resolution independent" in that the mapping can be applied at any resolution, so the original image can be zoomed by fixed point iteration at the desired resolution.

1.1.3 Edge-directed

While PDE- and patch-based methods produce high-quality results, they are often computationally expensive. In PDE-based methods, the result improves as it approaches steady state, but often the timestep must be restrictively small for stability and many steps are needed. In patch-based methods, expanding the search for similar patches usually improves the results. Naturally, they tend to accept extreme run times.

In contrast, edge-directed methods are often designed from the outset for computational efficiency. Edge-directed methods[16–20] try to interpolate along the direction of edges rather than across them, producing interpolations with sharp edges.

The difficulty in edge directed methods is in how they determine the direction of the edge, and this is primarily where they differ. Stepin[20] compares a pixel with each of its eight neighbors and decides by a threshold whether each neighbor is alike or different. For each of the 256 possible outcomes, the method assigns a tailored linear formula to interpolate the neighborhood.

1.2 Simplifying Objectives: Contour Consistency

Smoothness, step response, directional selectivity, efficiency—these objectives cannot be treated independently; they effect each other. In linear interpolation for example, the objectives of high smoothness order and sharp step response are in direct opposition. We simplify these objectives into a single objective called *contour consistency*: the local orientations of the interpolated contours should match and no new contour lines should be created. Contours in the interpolated image should be "consistent" with those in the given image.

Contour consistency implies that edges are interpolated without overshoots or jagged edges (step response and directional selectivity) and that constants are preserved, which is just enough constant order smoothness.

Image zooming motivated by contours has been successful in previous work. Jiang et al.[17] performed interpolation by solving two minimization problems. The first estimates the local contour orientations, and the second applies TV regularization to interpolate the image along the detected orientations. In the method of Luong et al.,[21] they begin by zooming the image with a linear method, then attempt to remove the artifacts. They pay special attention to the contours. TV flow is used to smooth out the contours, thereby removing jagged edge artifacts, and then a contour-preserving local contrast enhancement is used to sharpen blurry edges.

1.3 Outline

We develop two contour-based image zooming methods that approximate this objective. Section 2 develops an interpolation method, that is, approximately solving the degradation model

$$v = \mathrm{sample}(u) \tag{1}$$

for u, where v is the given image. Section 3 develops a second method adding a point spread function (PSF) h,

$$v = \mathrm{sample}(h * u). \tag{2}$$

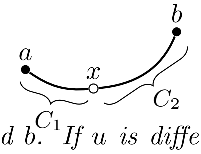
The interpolation method is more computationally efficient, requiring only a few dozen operations per pixel. The second method produces sharper results than the first, at the computational cost of solving a sparse symmetric positive definite linear system. Section 4 compares the proposed methods with several existing methods.

2. CONTOUR STENCIL INTERPOLATION

This section describes the contour stencil interpolation method. Given an image $u \in C^1(\mathbb{R}^2)$, denote by $\mathrm{TV}(C)$ its total variation along a smooth curve C,

$$\mathrm{TV}(C) := \int_0^T \left| \tfrac{\partial}{\partial t} u\bigl(\gamma(t)\bigr) \right| dt \quad \text{where } \gamma : [0,T] \to C. \tag{3}$$

Total variation has a relationship with interpolation error.



Theorem 2.1. *Consider the approximation*

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

and let $C = C_1 \cup C_2$ be a smooth curve passing through a, x, and b. If u is differentiable along C, then the approximation error is bounded by

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

Proof. We have

$$|u(a) - u(x)| = \left| \int_{C_1} \tfrac{\partial}{\partial t} u\bigl(\gamma(t)\bigr)\,dt \right| \le \mathrm{TV}(C_1)\,,$$

and similarly $|u(b) - u(x)| \le \mathrm{TV}(C_2)$. Combining these bounds yields the result. □

The proposed approach to interpolation comes directly from Thm. 2.1: minimize $\mathrm{TV}(C)$. If $\mathrm{TV}(C)$ is the smallest among a set of candidate curves, then the interpolation error has the smallest bound when interpolating along C. Next, we address how to estimate contours, which is where the "contour stencils" will come in.

2.1 Contour Stencils

Let $v : \Lambda \to \mathbb{R}$ be the given image where $v_{i,j}$ denotes the value of the (i,j)th pixel. Define the (i,j)th cell as the square whose corners correspond to $v_{i,j}$, $v_{i+1,j}$, $v_{i,j+1}$, $v_{i+1,j+1}$. For each cell, we want to find a curve C passing through the cell such that $\mathrm{TV}(C)$ is small.

A curve may be represented as a sequence of pixels $(\alpha_n) \in \Lambda$, and total variation may be discretized as

$$\sum_n |v(\alpha_{n+1}) - v(\alpha_n)| \approx \mathrm{TV}(C)\,. \tag{4}$$

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,*

$$\Bigl|\mathrm{TV}(\tilde{C}) - \mathrm{TV}(C)\Bigr| \le \int_0^T \bigl(M_1\,|\tilde{\gamma}'(t) - \gamma'(t)| + M_2\,|\tilde{\gamma}(t) - \gamma(t)|\bigr)\,dt$$

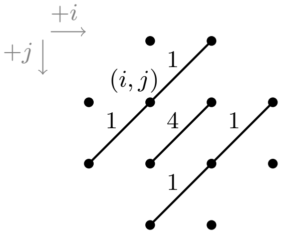
where $|\cdot|$ over a vector quantity denotes ℓ^2 norm.

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

$$\begin{aligned}
\Bigl|\mathrm{TV}(\tilde{C}) - \mathrm{TV}(C)\Bigr| &\le \int_0^T \Bigl| \bigl|\tfrac{\partial}{\partial t} u\bigl(\tilde{\gamma}(t)\bigr)\bigr| - \bigl|\tfrac{\partial}{\partial t} u\bigl(\gamma(t)\bigr)\bigr| \Bigr|\,dt \\
&\le \int_0^T \Bigl| \tfrac{\partial}{\partial t} u\bigl(\tilde{\gamma}(t)\bigr) - \tfrac{\partial}{\partial t} u\bigl(\gamma(t)\bigr) \Bigr|\,dt \\
&\le \int_0^T \left| \int_0^1 \tfrac{\partial^2}{\partial t \partial s} u\bigl(\Gamma(t,s)\bigr)\,ds \right| dt \\
&= \int_0^T \left| \int_0^1 \nabla u \cdot (\tilde{\gamma}' - \gamma') + \Gamma_t \cdot \nabla^2 u \cdot (\tilde{\gamma} - \gamma)\,ds \right| dt \\
&\le \int_0^T \bigl(M_1\,|\tilde{\gamma}'(t) - \gamma'(t)| + M_2\,|\tilde{\gamma}(t) - \gamma(t)|\bigr)\,dt
\end{aligned}$$

where $M_1 = \max_{t,s} \left|\nabla u\big(\Gamma(t,s)\big)\right|$, $M_2 = \max_{t,s} \left|\nabla^2 u \cdot \Gamma_t\big(\Gamma(t,s),s\big)\right|$. $\square$

The theorem implies that nearby parallel curves have approximately the same TV. Observing this property, the TV estimate (4) may be improved by averaging over several parallel curves.



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

Figure 1. An example contour stencil $\mathcal{S}$.

Let $\mathcal{S} : \Lambda \times \Lambda \to \mathbb{R}^+$ be a function describing weighted edges between pixels (that is, $\mathcal{S}(\alpha,\beta) > 0$ represents an edge weight connecting pixels $\alpha, \beta \in \Lambda$). We call $\mathcal{S}$ a *contour stencil*, and choose its nonzero edges such that they approximate several parallel curves localized over a small patch of the image (see Fig. 1). Define the total variation of $\mathcal{S}$ as

$$\mathrm{TV}(\mathcal{S}) := \tfrac{1}{|\mathcal{S}|} \sum_{\alpha,\beta\in\Lambda} \mathcal{S}(\alpha,\beta)\, |v_\alpha - v_\beta|\,, \tag{5}$$

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

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

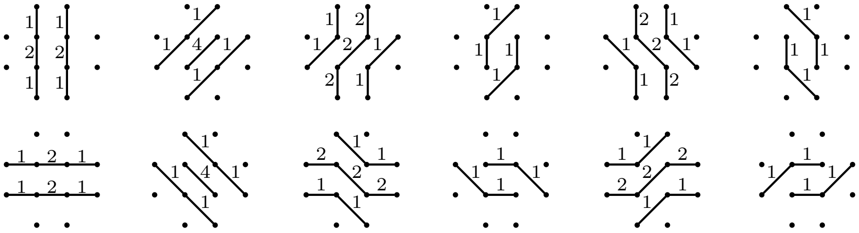


Figure 2. The 12 contour stencils used in the proposed method. Together they distinguish 8 directions.

Now we can find an estimate of the contours by comparing the total variations of different candidate stencils. The proposed stencil set Σ is shown in Fig. 2. The best-fitting stencil is

$$\mathcal{S}^* = \arg\min_{\mathcal{S}\in\Sigma} \mathrm{TV}(\mathcal{S}). \tag{6}$$

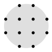
This stencil serves as a model of the underlying image contours. Since $\mathcal{S}^*$ is the stencil with the smallest estimated total variation, the estimated error bound is minimized when interpolating along its modeled contours:

$$|\hat{u}(x) - u(x)| \le \mathrm{TV}(C) \approx \mathrm{TV}(\mathcal{S}^*) = \min_{\mathcal{S}\in\Sigma} \mathrm{TV}(\mathcal{S})\,. \tag{7}$$

If the given image is noisy, spatially filtering $\mathrm{TV}(\mathcal{S})$ can improve the accuracy of the contour detection. Let $\mathrm{TV}^\rho(\mathcal{S}) = G_\rho * \mathrm{TV}(\mathcal{S})$, where G_ρ is a Gaussian with standard deviation ρ and the convolution is over Λ. Even for clean images, light filtering can improve the results.

The rationale for choosing the stencil set is, first, the stencil selection should be invariant to axis reflections and 90° rotations, and second, the support of the stencils should be as localized as possible. Let γ denote the center of the (i,j)th cell, then define the *radius* of Σ as

$$\rho(\Sigma) = \max\big\{|\gamma - \alpha| : \mathcal{S} \in \Sigma, \alpha \in \mathrm{supp}(\mathcal{S})\big\},$$

where $\operatorname{supp}(\mathcal{S}) \subset \Lambda$ denotes the support of stencil $\mathcal{S}$. The support  in the proposed stencil set (Fig. 2) is the smallest possible radius ($\rho \approx 1.581$) on which contour stencils can reliably distinguish more than two directions (on the support , the horizontal and vertical directions can be detected).

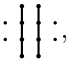
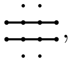
Remark 1. *The weights on the stencil edges have only a small effect on the detection. Results are similar if, for example, all edges have the same weight. The choice of weights affects the results most where there is a near tie in* (6) *for the minimum* $\mathrm{TV}(\mathcal{S})$, *that is, the weights can help where the detection gets confused.*

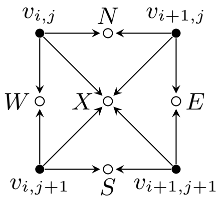
Remark 2. *The absolute value in the definitions* (3) *and* (4) *may be replaced by another metric. For color images for example,* $\left\|\vec{u}^1 - \vec{u}^2\right\| = \left|u^1_{\text{red}} - u^2_{\text{red}}\right| + \left|u^1_{\text{green}} - u^2_{\text{green}}\right| + \left|u^1_{\text{blue}} - u^2_{\text{blue}}\right|$.

2.2 Interpolation Formulas

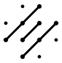
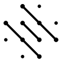
We now describe contour stencil interpolation, which takes inspiration from the one-dimensional Essentially Non-Oscillatory (ENO) interpolation technique.[22] In ENO, a piecewise polynomial interpolant is constructed in such a way as to avoid oscillations near singularities. For each interval, several stencils are considered and the one with the least divided difference is selected for interpolation. Similarly in our case, the $\mathrm{TV}(\mathcal{S})$, $\mathcal{S} \in \Sigma$, test how well the stencils fit the data, and we proceed using the best stencil as a model for the underlying function.

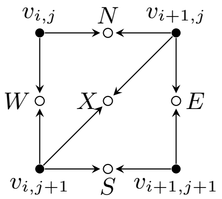
For the stencil set in Fig. 2, the following formulas give a simple way to do the interpolation. Filled circles • represent samples of v. The value of an interpolated node ∘ is the average of all incoming arrows.

For $\mathcal{S}^*_{i,j} =$ , , and the isotropic fallback, do standard bilinear interpolation:

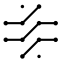
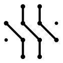
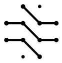
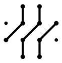


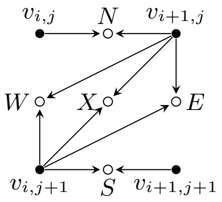
$$N = \tfrac{1}{2}(v_{i,j} + v_{i+1,j})$$
$$S = \tfrac{1}{2}(v_{i,j+1} + v_{i+1,j+1})$$
$$W = \tfrac{1}{2}(v_{i,j} + v_{i,j+1})$$
$$E = \tfrac{1}{2}(v_{i+1,j} + v_{i+1,j+1})$$
$$X = \tfrac{1}{4}(v_{i,j} + v_{i+1,j} + v_{i,j+1} + v_{i+1,j+1})$$

For $\mathcal{S}^*_{i,j} =$  and similarly for 

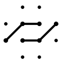
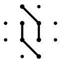
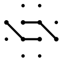
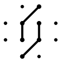


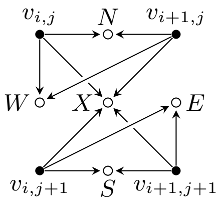
$$N = \tfrac{1}{2}(v_{i,j} + v_{i+1,j})$$
$$S = \tfrac{1}{2}(v_{i,j+1} + v_{i+1,j+1})$$
$$W = \tfrac{1}{2}(v_{i,j} + v_{i,j+1})$$
$$E = \tfrac{1}{2}(v_{i+1,j} + v_{i+1,j+1})$$
$$X = \tfrac{1}{2}(v_{i,j+1} + v_{i+1,j})$$

For $\mathcal{S}^*_{i,j} =$  and similarly for , , and 



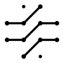
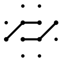
$$N = \tfrac{1}{2}(v_{i,j} + v_{i+1,j})$$
$$S = \tfrac{1}{2}(v_{i,j+1} + v_{i+1,j+1})$$
$$X = W = E = \tfrac{1}{2}(v_{i,j+1} + v_{i+1,j})$$

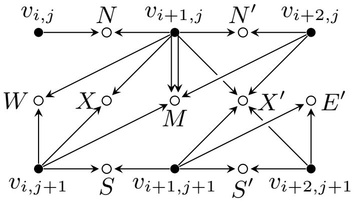
For $\mathcal{S}^*_{i,j} =$  and similarly for , , and 



$$N = W = \tfrac{1}{2}(v_{i,j} + v_{i+1,j})$$
$$S = E = \tfrac{1}{2}(v_{i,j+1} + v_{i+1,j+1})$$
$$X = \tfrac{1}{4}(v_{i,j} + v_{i+1,j} + v_{i,j+1} + v_{i+1,j+1})$$

In the typical case that an edge midpoint is shared between adjacent cells, its interpolated value is the average of the two assigned values.

For example, if $\mathcal{S}^*_{i,j} =$  and $\mathcal{S}^*_{i+1,j} =$ , then



$$E = \tfrac{1}{2}(v_{i,j+1} + v_{i+1,j})$$
$$W' = \tfrac{1}{2}(v_{i+1,j} + v_{i+2,j})$$
$$M = \tfrac{1}{2}(E + W')$$

Again, these interpolation formulas are just a simple implementation of the guiding principle that values should be interpolated along the contours. The same principle may be applied to derive interpolation formulas for other stencils and interpolation factors other than 2.

The steps for contour stencil interpolation are the following for each cell of the input image: compute the TV estimates (5), determine the best-fitting stencil $\mathcal{S}^*_{i,j}$, compute the interpolated pixels, and average shared midpoint values. For efficient implementation, define the differences

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

then the total variations $\mathrm{TV}(\mathcal{S})$ can be computed by adding the appropriate differences, and differences may be reused between successive cells. If $\rho = 0$, the interpolation amounts to a few dozen operations per pixel.

Thanks to the contour stencils, the interpolation produces sharp edges and partially recovers oriented textures (Fig. 3). All images are displayed with nearest-neighbor rendering such that individual pixels are visible.

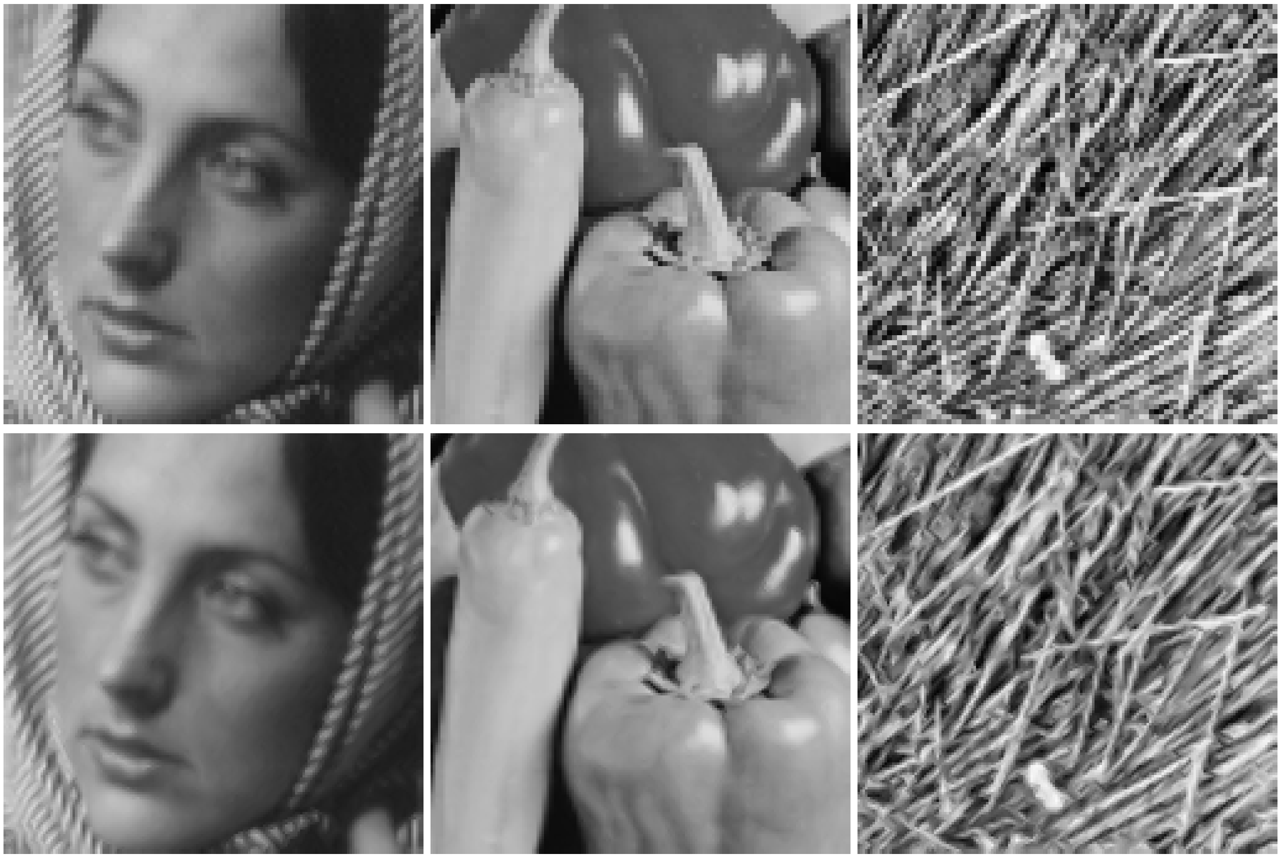


Figure 3. *Top row:* input images. *Bottom row:* contour stencil interpolation by factor 2.

3. CONTOUR STENCIL ZOOMING

The previous sections developed contour stencils and the contour stencil interpolation method. Here we use contour stencils again in a method that incorporates deconvolution as in degradation model (2). The advantage of deconvolution is sharper results, but the penalties are higher computational costs and overshoot artifacts if the PSF support is overestimated.

3.1 Graph Regularization

The method proposed here follows the graph regularization approach of Elmoataz et al.,[23] which they conceived for image denoising and segmentation and we adapt to image zooming. The high resolution image u is found as the solution of

$$\min_u E(u) = \sum_{\alpha\in\Lambda}\sum_{\beta\in\Lambda} w(\alpha,\beta)\,|u_\alpha - u_\beta|^2 \quad \text{subject to } {\downarrow}(h * u) = v \tag{8}$$

where $w : \Lambda\times\Lambda \to \mathbb{R}^+$ is a symmetric function defining a weighted graph on the pixels of u.

Suppose that h is identified as the lowpass filter of a wavelet (Appendix A addresses how to obtain such a wavelet). Let $Hv = {\downarrow}(h * u)$ and let $\tilde{H}, G, \tilde{G}$ be the corresponding operators such that $\tilde{H}H + \tilde{G}G = I$. Then the constraint can be eliminated: since ${\downarrow}(h * u) = v$ if and only if there exists d such that $u = \tilde{H}v + \tilde{G}d$, the minimization is equivalent to the unconstrained minimization $\min_d E(\tilde{H}v + \tilde{G}d)$.

We say the graph is *connected* if for any two pixels α, β there exists a finite sequence $\alpha = \alpha_0, \alpha_1, \ldots, \alpha_N = \beta$ in Λ such that $w(\alpha_n, \alpha_{n+1}) > 0$. If the graph is connected and finite, standard convex analysis shows that (8) has a unique solution.

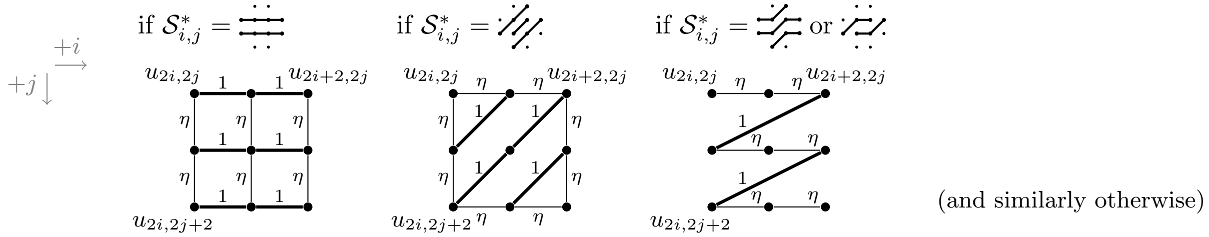

+i
+j
if $\mathcal{S}^*_{i,j} =$
if $\mathcal{S}^*_{i,j} =$
if $\mathcal{S}^*_{i,j} =$ or
$u_{2i,2j}$
$u_{2i+2,2j}$
$u_{2i,2j+2}$
1
η
(and similarly otherwise)

Figure 4. Graph construction for contour stencil zooming. The values denote the edge weights $w(\alpha, \beta)$.

For each cell in the given image, edges are added to the graph depending on the best-fitting stencil $S^*_{i,j}$ as shown in Fig. 4. Most edges follow the flow of the detected contours. For regularity and to ensure the graph is connected, a few edges with weight η link between adjacent contours. In the experiments we use $\eta = 1/4$.

3.2 Implementation

Let L denote the graph Laplacian $(Lu)_\alpha := \sum_\beta w(\alpha,\beta)(u_\alpha - u_\beta)$, then the minimizer of (8) satisfies

$$(\tilde{G}^* L \tilde{G})d = -\tilde{G}^* L \tilde{H} v. \tag{9}$$

The matrix $(\tilde{G}^* L \tilde{G})$ is symmetric positive definite. If h has finite impulse response, then $\tilde{G}$ can be selected to be sparse and hence $(\tilde{G}^* L \tilde{G})$ is also sparse. So the linear system is solved efficiently by conjugate gradients.

Figure 5 compares the proposed zooming method and Malgouyres' TV minimization.[2] Despite being based on a quadratic minimization, edges with the proposed method are nevertheless sharp. Figure 6 shows a close-up of the bear's fur, where we can observe that the proposed method recovers the fur texture more accurately.

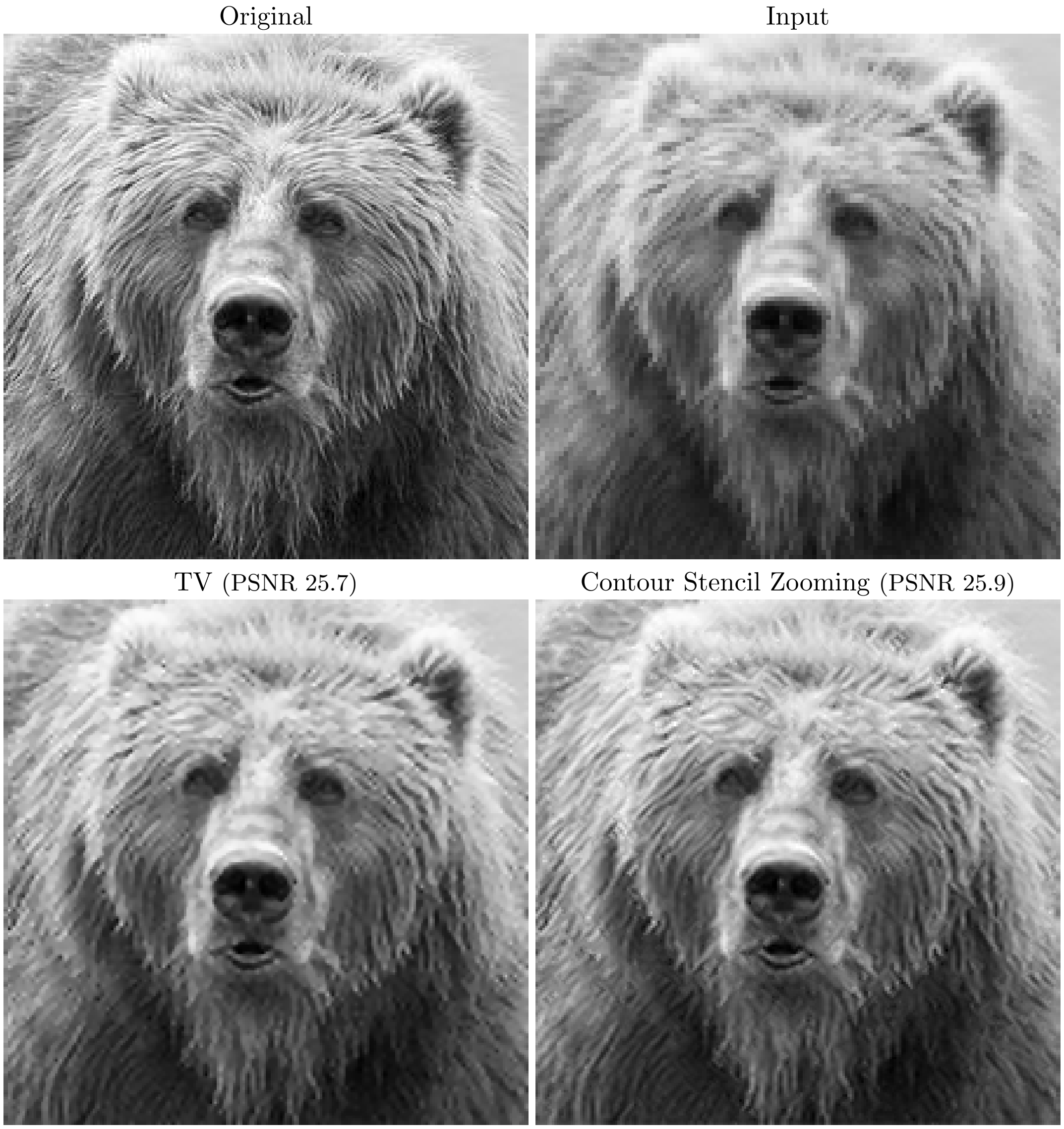


Figure 5. The TV-based method of Malgouyres[2] vs. contour stencil zooming. *Photograph by Steve Hillebrand, U.S. Fish and Wildlife Service.*

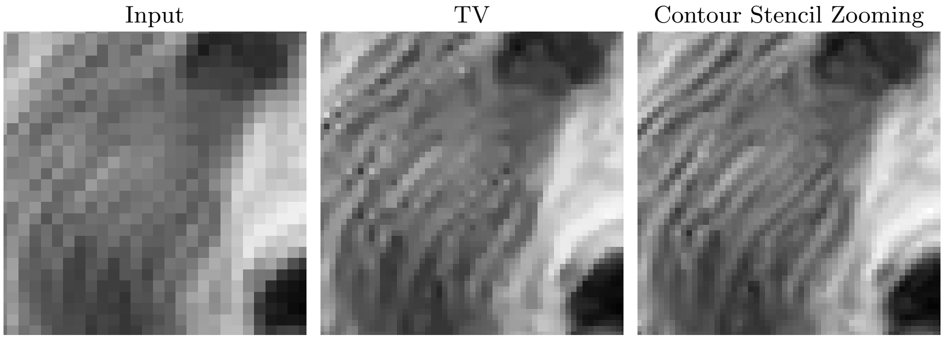
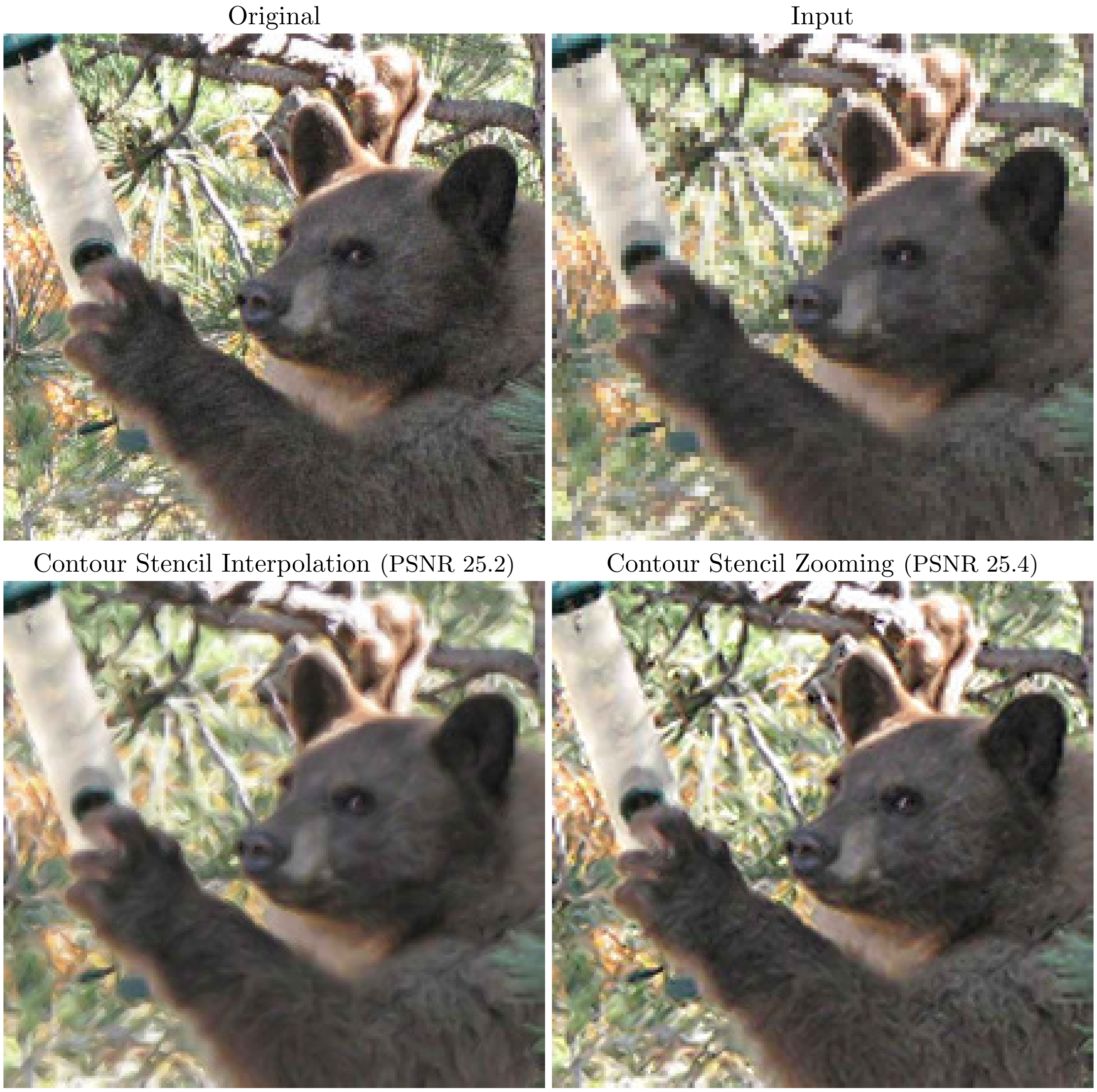


Figure 6. Close-up view of Fig. 5.

3.3 Color Zooming

For color images (or any vector-valued data), stencil selection may be done jointly over all channels as described in Remark 2. We find that the minimization (8) may be performed independently for each channel without significant color artifacts.



Original

Input

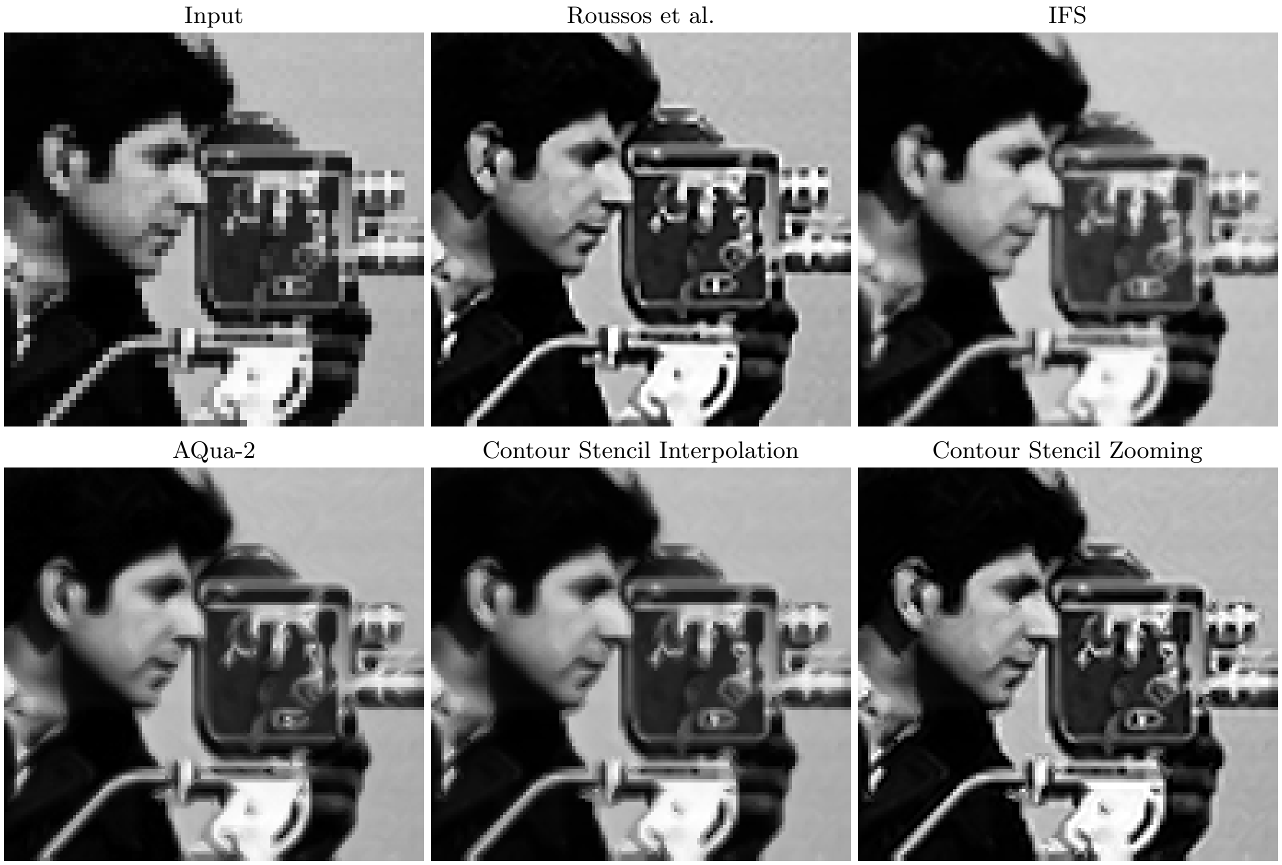
Contour Stencil Interpolation (PSNR 25.2)

Contour Stencil Zooming (PSNR 25.4)

Figure 7. Color example of factor-2 zooming with the proposed methods. The stencil selection is done using the ℓ^1 norm in $\mathrm{YC_bC_r}$ space (see Remark 2).

4. COMPARISON

Figure 8 compares contour stencil interpolation and contour stencil zooming alongside the PDE diffusion method of Roussos et al.,[6] an implementation of IFS image scaling,[24] and the edge directed method AQua-2 of Muresan.[16] The stencil denoising parameter is $\rho = 0.5$ and the PSF h used is the tensor product of the filter $1 + \frac{1}{4}(z + z^{-1})$.


Input

Roussos et al.

IFS

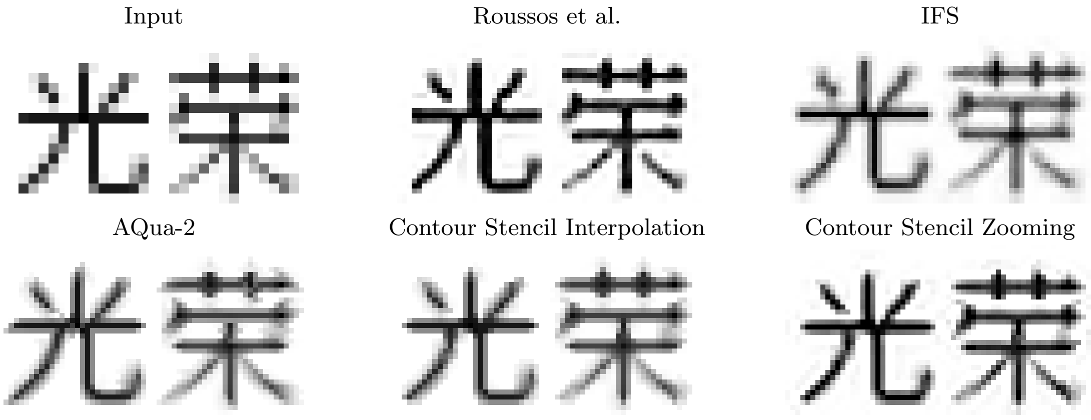
AQua-2

Contour Stencil Interpolation

Contour Stencil Zooming

Figure 8. Comparison with existing methods on *cameraman*.

Figure 9 compares the methods on a text image. This image, with its strongly oriented features in the line strokes, is especially well-suited for the proposed contour-based methods.


Input

Roussos et al.

IFS

AQua-2

Contour Stencil Interpolation

Contour Stencil Zooming

Figure 9. Comparison with existing methods on a text image.

5. CONCLUSIONS

To conclude, we consider strengths and weaknesses of the proposed approach.

Strengths

- Competing objectives are resolved by following the single objective of contour consistency (Sec. 1.2).
- Contour stencil selection minimizes an approximate bound of the interpolation error (Thm. 2.1).
- Contour stencil interpolation requires only a few dozen operations per pixel and can be parallelized. For contour stencil zooming, the bottleneck computation is solving a sparse symmetric positive definite linear system, which is relatively mild compared to most PDE- and patch-based methods.
- Visual quality is comparable to existing methods (Sec. 4).

Weaknesses, directions for future work

- The choice of stencil edge weights is ad hoc (Remark 1).
- Directional selectivity is limited to the directions represented in the stencil set.
- It is not clear how to extend to arbitrary scale factors or irregularly-sampled data.

APPENDIX A. CONSTRUCTING THE WAVELET

The constraint in minimization (8) may be eliminated if the PSF h is identified as the analysis lowpass filter of a wavelet. For arbitrary h, this requires constructing a wavelet. (Or in the other direction, we could choose the PSF h as the analysis lowpass filter of some existing wavelet.) Wavelets for this purpose can be designed using the wavelet lifting scheme.[25] As an example, we design wavelets for 5-tap symmetric PSFs.

Suppose that the PSF is the tensor product of a symmetric 5-tap filter

$$h(z) \propto 1 + a(z + z^{-1}) + b(z^2 + z^{-2}). \tag{10}$$

To produce a wavelet that has (10) as its lowpass analysis filter, we begin with a generic lifting scheme

$$\begin{bmatrix} h_e(z) & g_e(z) \\ h_o(z) & g_o(z) \end{bmatrix} = \begin{bmatrix} 1 & \alpha(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \beta(z+1) & 1 \end{bmatrix} \begin{bmatrix} 1 & \delta(1+z^{-1}) \\ 0 & 1 \end{bmatrix}$$

yielding

$$\begin{aligned} h(z) &\equiv h_e(z^2) + z^{-1} h_o(z^2) \\ &= (1 + 2\alpha\beta) + \beta(z + z^{-1}) + \alpha\beta(z^2 + z^{-2}). \end{aligned}$$

Filter (10) is obtained with $\alpha = b/a$ and $\beta = a/(1-2b)$. We choose $\delta = -(1+2\alpha)/(2+4\beta+8\alpha\beta)$ to ensure that the wavelet has at least one vanishing moment.

In one dimension, the wavelet transform's lifting scheme implementation is

Forward transform: $s = Hx$, $d = Gx$

$$\begin{cases} x_{2n+1} \leftarrow x_{2n+1} + \alpha(x_{2n+2} + x_{2n}) \\ x_{2n} \leftarrow x_{2n} + \beta(x_{2n+1} + x_{2n-1}) \\ x_{2n+1} \leftarrow x_{2n+1} + \delta(x_{2n+2} + x_{2n}) \\ (Hx)_n = x_{2n} \\ (Gx)_n = x_{2n+1} \end{cases}$$

Inverse transform: $x = \tilde{H}s + \tilde{G}d$

$$\begin{cases} x_{2n} \leftarrow s_n \\ x_{2n+1} \leftarrow d_n \\ x_{2n+1} \leftarrow x_{2n+1} - \delta(x_{2n+2} + x_{2n}) \\ x_{2n} \leftarrow x_{2n} - \beta(x_{2n+1} + x_{2n-1}) \\ x_{2n+1} \leftarrow x_{2n+1} - \alpha(x_{2n+2} + x_{2n}) \end{cases}$$

Similarly, for the adjoint transform,

Forward transform adjoint: $s = \tilde{H}^* x$, $d = \tilde{G}^* x$

$$\begin{cases} x_{2n} \leftarrow x_{2n} - \alpha(x_{2n+1} + x_{2n-1}) \\ x_{2n+1} \leftarrow x_{2n+1} - \beta(x_{2n+2} + x_{2n}) \\ x_{2n} \leftarrow x_{2n} - \delta(x_{2n+1} + x_{2n-1}) \\ (\tilde{H}^* x)_n = x_{2n} \\ (\tilde{G}^* x)_n = x_{2n+1} \end{cases}$$

Inverse transform adjoint: $x = H^* s + G^* d$

$$\begin{cases} x_{2n} \leftarrow s_n \\ x_{2n+1} \leftarrow d_n \\ x_{2n} \leftarrow x_{2n} + \delta(x_{2n+1} + x_{2n-1}) \\ x_{2n+1} \leftarrow x_{2n+1} + \beta(x_{2n+2} + x_{2n}) \\ x_{2n} \leftarrow x_{2n} + \alpha(x_{2n+1} + x_{2n-1}) \end{cases}$$

The wavelet is extended to two dimensions by the tensor product of the one dimensional transform.

REFERENCES

[1] Rudin, L. I., Osher, S. J., and Fatemi, E., "Nonlinear total variation based noise removal algorithms," *Physica D* **60**, 259–268 (1992).

[2] Malgouyres, F. and Guichard, F., "Edge direction preserving image zooming: A mathematical and numerical analysis," *SIAM Journal on Numerical Analysis* **39**(1), 1–37 (2002).

[3] Chan, T. F. and Shen, J., "Mathematical models for local nontexture inpaintings," *SIAM Journal on Applied Mathematics* **62**(3), 1019–1043 (2002).

[4] Bertalmío, M., Bertozzi, A. L., and Sapiro, G., "Navier-Stokes, fluid-dynamics and image and video inpainting," *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* , I355–I362 (2001).

[5] Belahmidi, A. and Guichard, F., "A partial differential equation approach to image zoom," in [*ICIP*], 649–652 (2004).

[6] Roussos, A. and Maragos, P., "Reversible interpolation of vectorial images by an anisotropic diffusion-projection PDE," *Int. J. Comput. Vision, Special Issue for the SSVM '07 conference* . to appear.

[7] Buades, A., Coll, B., and Morel, J. M., "A review of image denoising algorithms, with a new one," *Multiscale Modeling & Simulation* **4**(2), 490–530 (2005).

[8] Luong, H. Q., Ledda, A., and Philips, W., "Non-local interpolation," in [*ICIP*], 693–696 (2006).

[9] Protter, M., Elad, M., Takeda, H., and Milanfar, P., "Generalizing the non-local-means to super-resolution reconstruction," *IEEE Trans. on Image Process.* (2008).

[10] Peyré, G., Bougleux, S., and Cohen, L., "Non-local regularization of inverse problems," in [*ECCV08*], III: 57–68 (2008).

[11] Freeman, W. T., Jones, T. R., and Pasztor, E. C., "Example-based super-resolution," *IEEE Comput. Graphics Appl.* **22**(2), 56–65 (2002).

[12] Ebrahimi, M. and Vrscay, E., "Solving the inverse problem of image zooming using 'self-examples'," in [*ICIAR07*], 117–130 (2007).

[13] Li, D., Simske, S. J., and Mersereau, R. M., "Single image super-resolution based on support vector regression," in [*IJCNN*], 2898–2901 (2007).

[14] Fisher, Y., "Fractal image compression. SIGGRAPH'92 course notes," (1992).

[15] Chen, Y., Luo, Y., and Hu, D., "Image superresolution using fractal coding," *Optical Engineering* **47**(1), 017007 (2008).

[16] Muresan, D. D., "Fast edge directed polynomial interpolation," in [*ICIP (2)*], 990–993 (2005).

[17] Jiang, H. and Moloney, C., "A new direction adaptive scheme for image interpolation," in [*ICIP (3)*], 369–372 (2002).

[18] Kraus, M., Eissele, M., and Strengert, M., "GPU-based edge-directed image interpolation," in [*SCIA07*], 532–541 (2007).

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

[20] Stepin, M., "HQ2x magnification filter," www.hiend3d.com/hq2x.html (2003).

[21] Luon, H. Q. and Philips, W., "Sharp image interpolation by mapping level curves," *Proceedings for Visual Communications and Image Processing* , 2012–2022 (2005).

[22] Harten, A., Engquist, B., Osher, S., and Chakravarthy, S. R., "Uniformly high order accurate essentially non-oscillatory schemes, III," *J. Computational Physics* **71**(2), 231–303 (1987).

[23] Elmoataz, A., Lezoray, O., and Bougleux, S., "Nonlocal discrete regularization on weighted graphs: A framework for image and manifold processing," *IEEE Trans. on Image Process.* **17**(7), 1047–1060 (2008).

[24] onOne Software, "Genuine Fractals," www.ononesoftware.com (2008).

[25] Daubechies, I. and Sweldens, W., "Factoring wavelet transforms into lifting steps," tech. rep. (1996).