Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 4(3): 576–592.
DOI:10.3934/math.2019.3.576
Received: 16 April 2019
Accepted: 15 May 2019
Published: 30 May 2019

Research article

Anisotropic variation formulas for imaging applications

Thomas Asaki[1,*] **and Heather A. Moon**[2]

[1] Department of Mathematics and Statistics, Washington State University, P.O. Box 3113, Pullman, WA 99164, USA

[2] Division of Natural Sciences and Mathematics, Lewis-Clark State College, 500 8th Avenue, Lewiston, ID 83501, USA

* **Correspondence:** Email: tasaki@wsu.edu; Tel: +15093353135; Fax: +15093351188.

Abstract: The discrete anisotropic variation, sometimes referred to as the anisotropic gradient, and its integral are important in a variety of image processing applications and set boundary measure computations. We provide a method for computing the weight factors for general anisotropic variation approximations of functions on $\mathbb{R}^2$. The method is developed in the framework of regular arrays, but applicability extends to arbitrary finite discrete sampling strategies. The mathematical model and computations use concepts from vector calculus and introductory linear algebra so the discussion is accessible for upper-division undergraduate students.

Keywords: anisotropic gradient; image analysis
Mathematics Subject Classification: 49Mxx, 68U05, 65D18, 65D25

1. Introduction

Given a function $f : \Omega \to \mathbb{R}$, $\Omega \subseteq \mathbb{R}^2$, bounded. The variation of f is given by

$$\mathsf{Var}_f(x, y) = \max_{\substack{(u,v)\in\mathbb{R}^2 \\ \|(u,v)\|=1}} \lim_{\alpha\to 0} \left| \frac{f(x + \alpha u, y + \alpha v) - f(x, y)}{\alpha} \right|. \tag{1.1}$$

In the case that f is differentiable, $\mathsf{Var}_f(x, y) = |\nabla f(x, y)|$. Measures of the total variation, and its generalization the total p-variation, in images are used to solve a variety of image and data processing tasks through the use of variational minimization methods [1, 6].

One key application is noise removal [8, 11]. The total variation is an approximate measure of the magnitude of the noise because noisy images have more pixel-to-pixel intensity variation. The left and right images in Figure 1 show an image and a noisy version of the same scene, respectively. The computed variation is about 18 times larger for the noisy image.

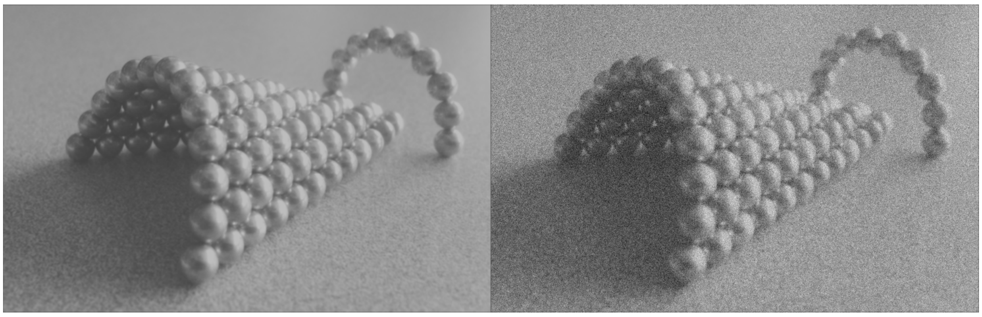


Figure 1. Left: A grayscale image of a set of magnets on a table. Right: A noisy version of the same scene. The total variation of the right image is approximately 18 times that of the left image.

A second application is that of edge-detection [12]. The image variation is relatively large at image locations with sudden changes in intensity, which is characteristic of a (visual) edge. The left and right images in Figure 2 shows an image of a pile of rocks and the corresponding variation image, respectively. The same method applied to characteristic functions (binary images) can be used to measure the length of a set boundary [7]. The left image of Figure 3 shows a binary image of the state of Idaho. The right image of Figure 3 shows the corresponding variation image for which the sum of the pixel intensities is proportional to the boundary length.

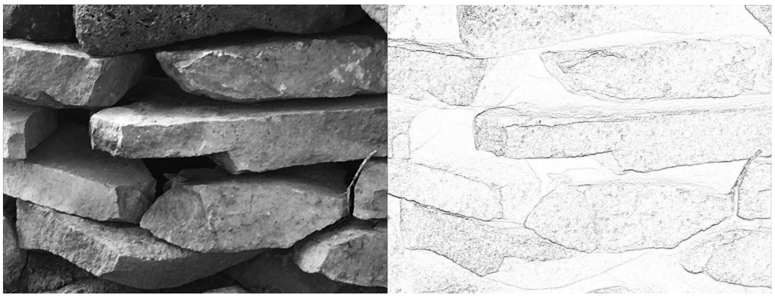


Figure 2. Left: A grayscale image of a rock wall. Right: The variation image which approximates the location of image intensity edges or intensity boundaries.

Other applications include the removal of blur artifacts due to imperfect focus of camera optics or long exposure of objects moving in a scene [4], inpainting [2, 5] which seeks to recover missing or corrupted data, image segmentation [3], compressed sensing [10], and determining characteristic length scales [14, 16].

An image is the discretization of a function $f : \Omega \to \mathbb{R}$ restricted to a finite regular grid. For example, a digital camera image is a collection of values $\{f_{i,j}\}$ at (pixel) locations (i, j) on a rectangular grid. Figure 4 is an example of such an image. It has image intensities, shown as shades of gray, on a regular grid of square pixels. In displaying the image, each pixel value is assigned a grayscale color which covers its associated pixel. In this case, a completely black pixel corresponds to a pixel value of zero, while a completely white pixel corresponds to a value of 255. We say that this image was

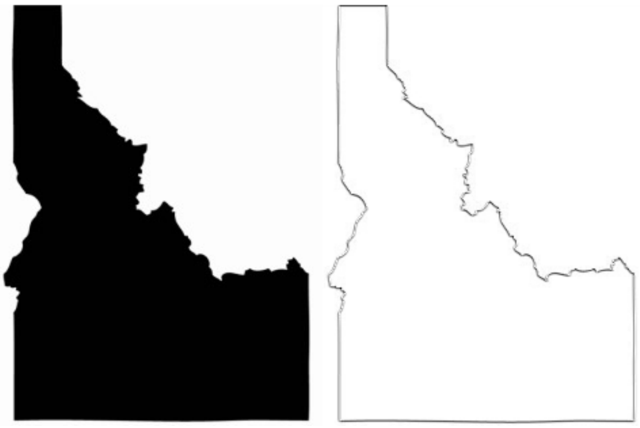
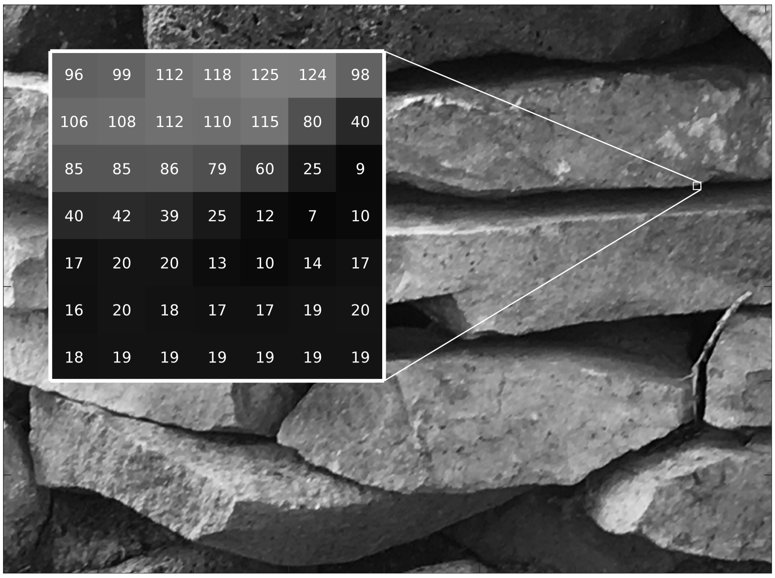


Figure 3. Left: A binary image of the state of Idaho. Right: The computed variation image. The sum of the values in the right image approximate boundary length.

obtained using an 8-bit camera because all pixel values are integers satisfying $0 \le f_{i,j} \le 255$.



96	99	112	118	125	124	98
106	108	112	110	115	80	40
85	85	86	79	60	25	9
40	42	39	25	12	7	10
17	20	20	13	10	14	17
16	20	18	17	17	19	20
18	19	19	19	19	19	19

Figure 4. A zoomed-in portion of the grayscale image of a rock wall (large white box) reveals pixel intensities (numerical values) arranged in a regular grid of subregions (pixels).

A regular grid may be composed of hexagonal pixels. One example is shown in Figure 5. This type of grid is not in widespread use because of the relative difficulty in simply specifying the geometry and the widespread use of rectangular coordinates in digital displays.

We propose a method for computing variation on images and image-like data as the sum of weighted differences of pixel intensities. Weights are chosen optimally for computing the variation on smooth arbitrary functions and their discrete representations. We begin in Section 2 by defining the anisotropic variation on discrete images and introduce displacement vector sets used to calculate the

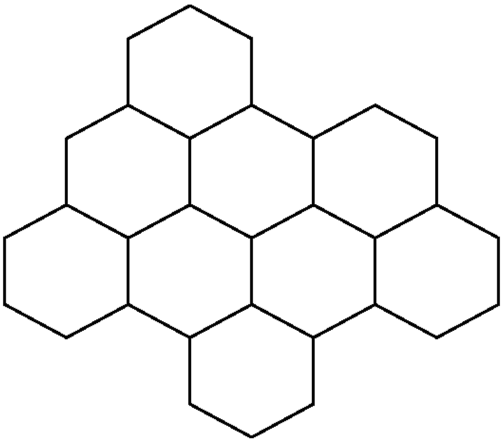


Figure 5. An example of a hexagonal array which could be used to display images.

variation. In Section 3 we develop the theoretical aspects of the variation on discrete data such as images and define the anisotropic gradient function. This function is defined in terms of a geometric sampling strategy and optimal weight parameters. In Section 4 we provide details on the computation of the anisotropic gradient weights for the general scenario and address existence and uniqueness. In Section 5, we provide specific computational examples for determining optimal weights including computations on rectangular grids, hexagonal grids and sampling uniformly in angle without reference to a grid. Conclusions are provided in Section 6.

2. Computing anisotropic image variation

A central task of image processing is to approximate key continuous functions, such as Var_f in Eq. (1.1), on the scene space using only the discrete image data. Recall that $\nabla f(x, y)$ is the vector whose components are the partial derivatives of f, each of which is defined by limits

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

and

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

In the case of discrete image data, we cannot take the limit $h \to 0$ because the function is defined at only discrete locations. Thus, for our image analysis tasks, f need not be differentiable. Instead, our plan is to approximate Var_f at each pixel location (x_ℓ, y_ℓ). That is, we are interested in a finite sampling strategy for approximating

$$g_\ell \equiv |\nabla f(x_\ell, y_\ell)|.$$

It is natural to consider a direct computation

$$g_\ell = \sqrt{f_x^2(x_\ell, y_\ell) + f_y^2(x_\ell, y_\ell)}, \text{ where } f_x = \frac{\partial f}{\partial x}, \text{ and } f_y = \frac{\partial f}{\partial y}.$$

However, f is known only at the finite set of locations (x_ℓ, y_ℓ) and we do not know f_x nor f_y. We can approximate the variation at each array location, $g_{i,j}$, with the forward difference strategy

$$g_{i,j} \approx \sqrt{\left(\frac{f_{i+1,j} - f_{i,j}}{h_x}\right)^2 + \left(\frac{f_{i,j+1} - f_{i,j}}{h_y}\right)^2}, \tag{2.1}$$

where h_x and h_y are the horizontal and vertical distances between pixels. Eq. (2.1) is an example of an isotropic gradient approximation, which is exact in the following sense for differentiable functions:

$$\begin{aligned} \mathsf{Var}_f(x, y) &= |\nabla f(x, y)| \\ &= \lim_{h_x, h_y \to 0} \sqrt{\left(\frac{f(x + h_x, y) - f(x, y)}{h_x}\right)^2 + \left(\frac{f(x, y + h_y) - f(x, y)}{h_y}\right)^2} \end{aligned}$$

Throughout the remaining discussion, we will take $h_x = h_y = 1$ for simplicity.

We consider anisotropic approximations for the variation having the following form [8].

Definition 1. *Suppose $\Omega \subseteq \mathbb{R}^2$. Let $f : \Omega \to \mathbb{R}$, define the set of nonzero displacement vectors to be $V = \{d_k = (u_k, v_k)\}_{k=1}^m$, and let $\{w_k\}_{k=1}^m$ be a set of non-negative scalars. The function*

$$g(x, y) \equiv \sum_{k=1}^{m} w_k |f(x + u_k, y + v_k) - f(x, y)| \tag{2.2}$$

is an anisotropic variation of f.

If a location (x, y) corresponds to pixel array indices (i, j) in an image and if the components of the displacement vectors are integer, then $g(x, y)$ is determined by available pixel intensities. Several possible configurations of displacement vectors are shown in Figure 6. These cases correspond to the following displacement vector sets.

$$\begin{aligned} V_A &= \{(1, 0), (0, 1)\} \\ V_B &= \{(1, 0), (0, 1), (-1, 0), (0, -1)\} \\ V_C &= \{(1, 0), (0, 1), (-1, -1)\} \\ V_D &= \{(0, 1), (-1, -1), (1, -1)\} \\ V_E &= V_B \cup \{(1, 1), (-1, 1), (-1, -1), (1, -1)\} \\ V_F &= V_E \cup \{(2, 1), (1, 2), (-1, 2), (-2, 1), (-2, -1), (-1, -2), (1, -2), (2, -1)\} \end{aligned}$$

V_A is the two-direction set noted by Condat [6] with unit weights. V_B specifies computations on the Von Neumann neighborhood of four nearest neighbors and V_E the Moore neighborhood of eight nearest neighbors. V_F uses 16 nearest neighbors which has been successful in some recent computations [9]. V_A is a two-vector forward difference set yielding the approximation

$$\begin{aligned} g_{i,j} &= w_1 \left|f_{i+u_1, j+v_1} - f_{i,j}\right| + w_2 \left|f_{i+u_2, j+v_2} - f_{i,j}\right| \\ &= w_1 \left|f_{i+1,j} - f_{i,j}\right| + w_2 \left|f_{i,j+1} - f_{i,j}\right|. \end{aligned}$$

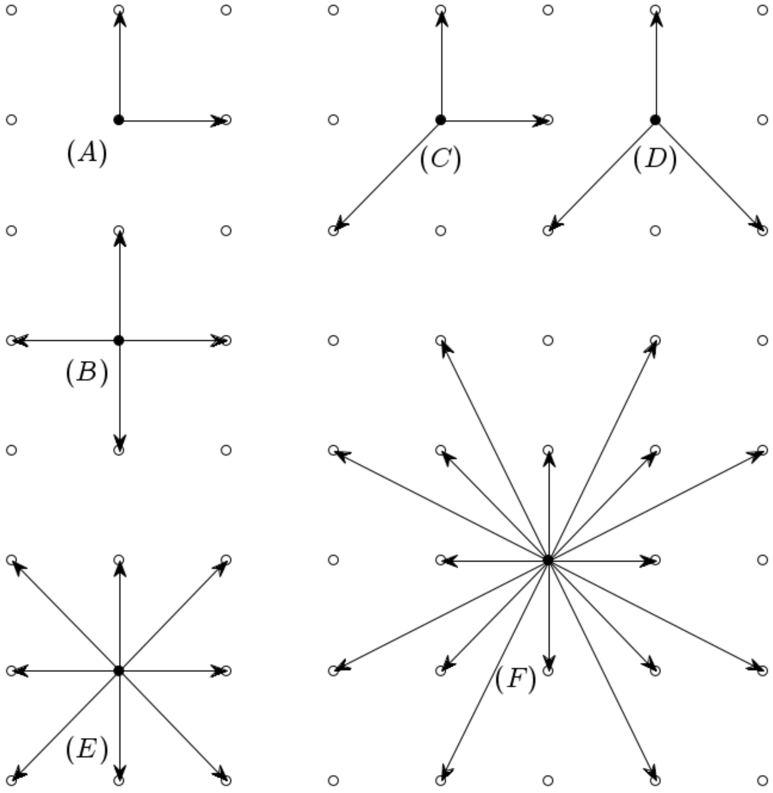


Figure 6. Six examples of displacement vector sets which can be used to compute discrete local anisotropic gradients on rectangular arrays such as digital images.

V_C shows displacement vectors that lead to the approximation

$$g_{i,j} = w_1 \left| f_{i+1,j} - f_{i,j} \right| + w_2 \left| f_{i,j+1} - f_{i,j} \right| + w_3 |f_{i-1,j-1} - f_{i,j}|.$$

These examples show that the anisotropic variation is not a unique function, but is defined by the chosen direction set V and scalar weights $\{w_k\}$. Notice also that, at a local extreme point for f, where $\nabla f = 0$, the anisotropic variation is not necessarily equal to zero. This suggests that the anisotropic variation does not accurately approximate the steepness of functions around local extrema.

3. Anisotropic weights

The weight computations presented here are *not* specific to the rectangular grids illustrated in Figure 6. We also consider direction sets on hexagonal tilings, some of which are shown in Figure 7. However, any other nonzero direction set in $\mathbb{R}^2$ can be used. No standard method for determining weights seems to be in general use, though some authors reference weights used in similar computational tasks [15] or weights based on geometric considerations [13]. In this section, we discuss a criterion for choosing optimal weights. We also discuss assumptions about the functions for which we use the anisotropic variation.

The goal in choosing specific weights for the anisotropic variation is to minimize the discrepancy between $g(x, y)$ Eq. (2.1) and $\mathsf{Var}_f(x, y)$ Eq. (2.2) at pixel locations. Weights should be chosen so that

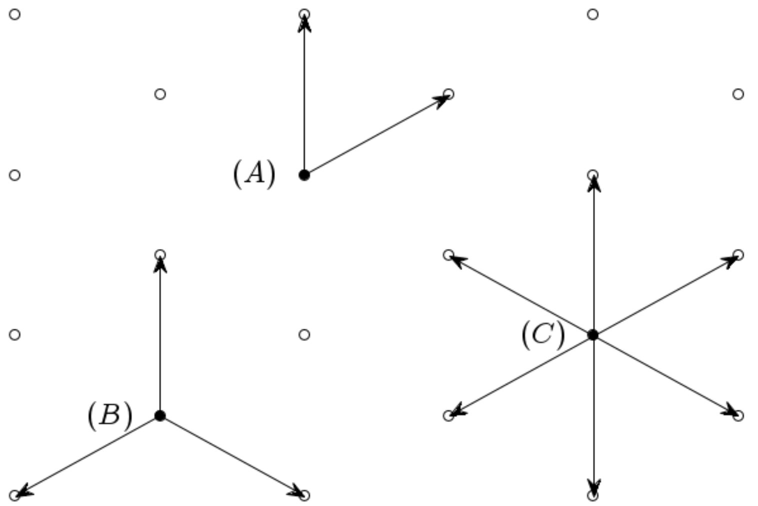


Figure 7. Three examples of displacement vector sets which can be used to compute discrete local anisotropic gradients on hexagonal arrays.

the approximation is good for arbitrary image functions $f(x, y)$ as no set of weights can be optimal for all functions. Thus, we must make assumptions on the types of functions we expect to encounter. Our first assumption is that the gradient of $f(x, y)$ is slowly varying over the sampling distances $||d_k||$. In this case, differencing strategies, such as Eq. (2.2), should produce good results because any variation calculation is only weakly affected by sampling distance. To understand this, consider a one-dimensional example in which we seek to compute $|df/dx|$ for the function $f : \mathbb{R} \to \mathbb{R}$. We have

$$\frac{df}{dx} = \lim_{h\to 0} \frac{f(x+h) - f(x)}{h}.$$

Notice that if f is approximately linear or affine on $[x, x+h]$ with slope s then

$$\frac{f(x+h) - f(x)}{h} \approx \frac{f(x) + sh - f(x)}{h} = s,$$

and $df/dx \approx s$ independent of sampling distance h.

Our second assumption is that we have no prior knowledge concerning the direction of the gradient. If we had some such information, we could preferentially weight the anisotropic gradient terms associated with directions which we believe are close to the gradient direction. So, whatever weights we choose must be independent of actual gradient vector orientation, but they could depend on relative angles between sampling directions. We expect the use of our anisotropic variation to be robust to computing gradients of any orientation.

With these two assumptions, we begin by considering a family of functions, $\mathcal{F}$, whose graphs are planes and whose gradient has unit length. We seek optimal weights relative to this family. Functions in this family satisfy $\nabla f_\theta = (\cos\theta, \sin\theta)$, where θ is a parameter identifying a family member. The functions are then given by

$$f_\theta(x, y) = x\cos\theta + y\sin\theta, \quad 0 \le \theta < 2\pi.$$

Our objective is to approximate the variation Var_{f_θ} with the anisotropic variation g_θ as closely as possible by determining optimal weights $\{w_k\}$ for a given direction set V. To add clarity of notation, let $\mathsf{w} = [\mathsf{w}_1, \mathsf{w}_2, \ldots, \mathsf{w}_m]^T$ and define g^{w} to be the anisotropic gradient with chosen weights w_k, $1 \leq k \leq m$. We measure optimality based on the squared error function. That is, for any given function $f_\theta \in \mathcal{F}$, we seek weights $w_1, w_2, \ldots, w_m$ which would minimize the function given by

$$\begin{aligned} E(\mathsf{w}, \theta) &= \left(\mathsf{Var}_{f_\theta} - g_\theta^{\mathsf{w}}\right)^2 \\ &= \left(1 - \sum_{k=1}^{m} \mathsf{w}_k \left|f_\theta(x + u_k, y + v_k) - f_\theta(x, y)\right|\right)^2 \\ &= \left(1 - \sum_{k=1}^{m} \mathsf{w}_k \left|u_k \cos\theta + v_k \sin\theta\right|\right)^2 . \end{aligned}$$

However, finding the weights that minimize the squared error above is optimal for only one such function $f_\theta \in \mathcal{F}$. We are interested in finding our best overall choice of weights that would minimize the total integrated squared error over all family members (for all possible θ):

$$\begin{aligned} H(\mathsf{w}) &= \int_0^{2\pi} E(\mathsf{w}, \theta)\, d\theta \\ &= \int_0^{2\pi} \left(1 - \sum_{k=1}^{m} \mathsf{w}_k \left|u_k \cos\theta + v_k \sin\theta\right|\right)^2 d\theta. \end{aligned} \tag{3.1}$$

That is, we want to find a point $w = [w_1, w_2, \ldots, w_m]^T$ whose coordinates are the weights so that $H(w) \leq H(\mathsf{w})$ for all $\mathsf{w} \in \mathbb{R}^m$. We seek minimizers w of H by finding stationary points (where $\nabla_{\mathsf{w}} H(w) = 0$). Here, we use the notation ∇_{w} to indicate that the derivatives are taken with respect to the variables $\mathsf{w}_1, \mathsf{w}_2, \ldots, \mathsf{w}_m$. The j^{th} component of $\nabla_{\mathsf{w}} H(w)$ $(1 \leq j \leq m)$ is

$$\begin{aligned} \frac{\partial H(w)}{\partial \mathsf{w}_j} &= -2 \int_0^{2\pi} \left|u_j \cos\theta + v_j \sin\theta\right| \left(1 - \sum_{k=1}^{m} w_k \left|u_k \cos\theta + v_k \sin\theta\right|\right) d\theta \\ &= -2 \int_0^{2\pi} \left|u_j \cos\theta + v_j \sin\theta\right| \, d\theta \\ &\quad + 2 \sum_{k=1}^{m} w_k \int_0^{2\pi} \left|u_j \cos\theta + v_j \sin\theta\right| \cdot \left|u_k \cos\theta + v_k \sin\theta\right| d\theta \end{aligned}$$

In order to simplify the representation and facilitate further computations, we define the integrals in each of these terms as

$$J(k) = \int_0^{2\pi} \left|u_k \cos\theta + v_k \sin\theta\right| d\theta$$

and

$$K(k, j) = \int_0^{2\pi} \left|u_k \cos\theta + v_k \sin\theta\right| \cdot \left|u_j \cos\theta + v_j \sin\theta\right| d\theta.$$

Thus

$$\frac{\partial H(w)}{\partial \mathsf{w}_j} = -2J(j) + 2 \sum_{k=1}^{m} w_k K(k, j).$$

Notice that this second term, involving K, can be written using a matrix product:

$$2\sum_{k=1}^{m} w_k K(k,j) = 2\begin{bmatrix} K(j,1) & K(j,2) & \cdots & K(j,m) \end{bmatrix}\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix}.$$

Now, stationary points w of H satisfy $\sum_{k=1}^{m} w_k K(k,j) = J(j)$, for $1 \le j \le m$. That is, the matrix equation below holds.

$$\begin{bmatrix} K(1,1) & K(1,2) & \cdots & K(1,m) \\ K(2,1) & K(2,2) & \cdots & K(2,m) \\ \vdots & \vdots & & \vdots \\ K(m,1) & K(m,2) & \cdots & K(m,m) \end{bmatrix}\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix} = \begin{bmatrix} J(1) \\ J(2) \\ \vdots \\ J(m) \end{bmatrix}, \tag{3.2}$$

The integrals evaluate to

$$J(k) = 4\|d_k\|$$

and

$$K(k,j) = \|d_k\|\|d_j\|\left[\left(\pi - 2\Delta_{kj}\right)\cos(\Delta_{kj}) + 2\sin(\Delta_{kj})\right],$$

where Δ_{kj} is the minimum positive angle between sampling directions d_j and d_k.

4. Determining optimal weights

Collecting the integral computations, we find that the optimal weights w satisfy the following system of equations.

$$\begin{gathered} Kw = J, \\ K(k,j) = \|d_k\|\ \|d_j\|\ \left(\left(\pi - 2\Delta_{kj}\right)\cos(\Delta_{kj}) + 2\sin(\Delta_{kj})\right), \\ K(k,k) = \pi\|d_k\|^2, \\ J(k) = 4\|d_k\|. \end{gathered}$$

We see that the entries of each row of the matrix K and the corresponding entry in J contain a common factor, $\|d_k\|$ for row k. These common factors do not affect the solution and can be removed. Furthermore, the entries of each column of the matrix K contain a common factor, $\|d_j\|$ for column j. Thus, each entry in the solution vector w can be rescaled by the corresponding vector norm. This leads to the somewhat simplified system of equations:

$$\begin{gathered} Az = B, \\ A(k,j) = \left(\pi - 2\Delta_{kj}\right)\cos(\Delta_{kj}) + 2\sin(\Delta_{kj}), \\ A(k,k) = \pi, \\ B(k) = 4, \\ w_j = z_j/\|d_j\|.. \end{gathered} \tag{4.1}$$

Now we consider existence and uniqueness of solutions to $Az = B$. We first show that at least one solution exists. Next, we show that if some pair of direction vectors are colinear, then there exist infinitely many solutions. Finally, we choose, among possibly infinitely many solutions, the solution z that has minimum 2-norm.

Claim 1. *The system of equations $Az = B$ given in (4.1) has at least one solution.*

Proof. Recall that solutions to (4.1) are stationary points of the function H given in Eq. (3.1). We will show that H has at least one stationary point. First note that $H(\mathsf{w}) \geq 0$ for all choices of weights $\mathsf{w}_1, \mathsf{w}_2, \ldots, \mathsf{w}_m$.

Now, let us use the integral computations above to rewrite $H(\mathsf{w})$.

$$\begin{aligned} H(\mathsf{w}) &= \int_0^{2\pi} \left(1 - \sum_{k=1}^{m} \mathsf{w}_k \left|u_k \cos\theta + v_k \sin\theta\right|\right)^2 \, d\theta \\ &= \int_0^{2\pi} \Bigg[1 - 2\sum_{k=1}^{m} \mathsf{w}_k \left|u_k \cos\theta + v_k \sin\theta\right| \\ &\qquad + \sum_{k=1}^{m}\sum_{j=1}^{m} \mathsf{w}_k \mathsf{w}_j \left|u_k \cos\theta + v_k \sin\theta\right| \left|u_j \cos\theta + v_j \sin\theta\right| \Bigg] d\theta \\ &= 2\pi - 2\mathsf{w}^T J + \mathsf{w}^T K \mathsf{w}, \end{aligned}$$

where J and K are defined above. Notice that H is a quadratic function of $\mathsf{w}_1, \mathsf{w}_2, \ldots, \mathsf{w}_m$ (the weights).

Notice also that if K is negative definite or indefinite (possessing at least one negative eigenvalue) then the quadratic, H, would obtain negative values for some w. So K must be positive definite (or positive semi-definite). Thus, H is a convex quadratic function, though not necessarily strictly convex, and bounded below by zero. Therefore H has at least one stationary point and the system of equations (4.1) has at least one solution. □

Another way to understand the result of the above proof is to note that a quadratic function which is bounded below (non-negative in this case) must be either convex quadratic or constant along any direction. If the function were to behave (nonconstant) linearly along some direction then it could not be bounded below.

Claim 2. *Suppose there exist two distinct vectors $d_k, d_j \in V$ which are colinear. Then, the system of equations $Az = B$ given in (4.1) has infinitely many solutions.*

Proof. The entries in row k of matrix A are determined solely by the minimum angle between each vector in V and the line $\{v = \alpha d_k \mid \alpha \in \mathbb{R}\}$. If d_j is colinear with d_k, then they define the same line and rows j and k of A must be equal. Thus, rank$(A) < m$ and, by the previous claim, $Az = B$ has infinitely many solutions. □

If the solution is unique, then we have $z^* = A^{-1}B$. If the solution is not unique, we proceed by diagonalizing A and constructing a pseudo-inverse operator P such that $z^* = PB$.

Since A is a real symmetric matrix, it is diagonalizable by an orthogonal matrix Q as $A = QDQ^T$ where D is a diagonal matrix consisting of the eigenvalues of A. We can order the eigenvalues λ_i, and the corresponding columns of Q so that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$. Note also that $\lambda_m \geq 0$ because of the convexity of $H(\mathsf{w})$. Suppose there are κ positive (not zero) eigenvalues. Let $\tilde{Q}$ be the matrix consisting of the first κ columns of Q. Let $\tilde{D}$ be the $\kappa \times \kappa$ diagonal matrix with entries $\lambda_1, \lambda_2, \cdots, \lambda_\kappa$.

Claim 3. *The pseudo-inverse solution $\tilde{z}$ of minimum 2-norm for solving the system of equations (4.1) is $\tilde{z} = PB = \tilde{Q}\tilde{D}^{-1}\tilde{Q}^T B$.*

Proof. Consider the unique representation $z = \tilde{z} + z_0$, where z_0 is in $\mathcal{N}(A)$, the null space of A, and $\tilde{z}$ is in $\mathcal{N}(A)^\perp$, the orthogonal complement of $\mathcal{N}(A)$. We show that $PB = \tilde{z}$. That is, P is the invertible operator that maps from the range of A to $\mathcal{N}(A)^\perp$. We write $Q = \begin{bmatrix} \tilde{Q} & Q_0 \end{bmatrix}$ and note that the columns of Q_0 form a basis for $\mathcal{N}(A)$ and the columns of $\tilde{Q}$ form a basis for $\mathcal{N}(A)^\perp$. Observe,

$$\begin{aligned}
PB &= PAz \\
&= \left(\tilde{Q}\tilde{D}^{-1}\tilde{Q}^T\right)\left(\begin{bmatrix} \tilde{Q} & Q_0 \end{bmatrix}\begin{bmatrix} \tilde{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\begin{bmatrix} \tilde{Q}^T \\ Q_0^T \end{bmatrix}\right)z \\
&= \left(\tilde{Q}\tilde{D}^{-1}\tilde{Q}^T\right)\left(\begin{bmatrix} \tilde{Q} & Q_0 \end{bmatrix}\begin{bmatrix} \tilde{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\begin{bmatrix} \tilde{Q}^T z \\ Q_0^T z \end{bmatrix}\right) \\
&= \tilde{Q}\tilde{D}^{-1}\tilde{Q}^T\left(\tilde{Q}\tilde{D}\tilde{Q}^T z\right) \\
&= \tilde{Q}\tilde{D}^{-1}\tilde{D}\tilde{Q}^T z \\
&= \tilde{Q}\tilde{Q}^T z \\
&= \tilde{z}
\end{aligned}$$

Thus, P returns solution $\tilde{z}$: $A\tilde{z} = A\tilde{z} + 0 = A\tilde{z} + Az_0 = Az = B$.

Next, we show that $\tilde{z}$ represents the solution of minimum 2-norm. If $\tilde{z}$ is the unique solution, then it is also the solution of minimum norm. Otherwise, let $\tilde{y}$ be any other solution. As $A\tilde{y} = B$, $\tilde{y} - \tilde{z} \in \mathcal{N}(A)$. Consider the standard inner product on $\mathbb{R}^m$ and noting that any vector in $\mathcal{N}(A)$ is orthogonal to $\tilde{z}$, we have $\langle \tilde{y}, \tilde{y} \rangle = \langle \tilde{z} + \tilde{y} - \tilde{z}, \tilde{z} + \tilde{y} - \tilde{z} \rangle = \langle \tilde{z}, \tilde{z} \rangle + \langle \tilde{y} - \tilde{z}, \tilde{y} - \tilde{z} \rangle$. Or, equivalently, $||\tilde{y}||^2 = ||\tilde{z}||^2 + ||\tilde{y} - \tilde{z}||^2$. Thus, $||\tilde{y}|| \geq ||\tilde{z}||$. □

5. Example weight computations

We next present example computations illustrating the determination of weights for several scenarios from Figures 6 and 7. Table 1 provides weight computation results for each scenario. We end this section with the computation for uniformly distributed (in angle) unit vector directions.

5.1. *Two nearest neighbor direction set, V_A*

We have direction set $V_A = \{(1, 0), (0, 1)\}$. We first compute the direction vector norms and the matrix elements $A(k, j)$:

$$||d_1|| = ||d_2|| = 1.$$

$$\Delta_{1,2} = \pi/2.$$

$$A(1, 1) = A(2, 2) = \pi.$$

$$A(1, 2) = A(2, 1) = 2.$$

For this set, Eq. (4.1) takes the form

$$\begin{bmatrix} \pi & 2 \\ 2 & \pi \end{bmatrix}\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 4 \end{bmatrix}.$$

The matrix A is invertible. The solution is

$$z_1 = z_2 = \frac{4}{\pi + 2} \approx 0.77797\ ,$$

and the weights are

$$w_1 = w_2 = \frac{4}{\pi + 2} \approx 0.77797\ .$$

5.2. Non-colinear three-direction set, V_C

We have direction set $V_C = \{(1,0),(0,1),(-1,-1)\}$ We first compute the direction vector norms and the matrix elements $A(k, j)$:

$$||d_1|| = 1,\ \ ||d_2|| = 1,\ \ ||d_3|| = \sqrt{2},$$

$$\Delta_{1,2} = \pi/2,\ \ \Delta_{1,3} = \Delta_{2,3} = \pi/4.$$

$$A(1,1) = A(2,2) = A(3,3) = \pi$$

$$A(1,2) = 2,\ \ A(1,3) = A(2,3) = \frac{\pi + 4}{2\sqrt{2}}.$$

For this set, Eq. (4.1) takes the form

$$\begin{bmatrix} \pi & 2 & \frac{\pi+4}{2\sqrt{2}} \\ 2 & \pi & \frac{\pi+4}{2\sqrt{2}} \\ \frac{\pi+4}{2\sqrt{2}} & \frac{\pi+4}{2\sqrt{2}} & \pi \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} 4 \\ 4 \\ 4 \end{bmatrix}.$$

The matrix A is invertible. The solution is

$$z_1 \approx 0.72501\ ,$$

$$z_2 \approx 0.72501\ ,$$

$$z_3 \approx 0.10784\ .$$

Thus,

$$w_1 = z_1/||d_1|| \approx 0.72501\ ,$$

$$w_2 = z_2/||d_2|| \approx 0.72501\ ,$$

$$w_3 = z_3/||d_3|| \approx 0.07635\ .$$

5.3. *Four nearest neighbor set, V_B*

Consider Von Neumann set $V_B = \{(1,0),(0,1),(-1,0),(0,-1)\}$. Proceeding as before, we find

$$\|d_1\| = \|d_2\| = \|d_3\| = \|d_4\| = 1.$$

$$\Delta_{1,2} = \Delta_{2,3} = \Delta_{3,4} = \Delta_{1,4} = \pi/2.$$

$$\Delta_{1,3} = \Delta_{2,4} = 0.$$

$$A(1,1) = A(2,2) = A(3,3) = A(4,4) = \pi.$$

$$A(1,2) = A(2,3) = A(3,4) = A(1,4) = 2.$$

$$A(1,3) = A(2,4) = \pi.$$

For this set, Eq. (4.1) takes the form

$$\begin{bmatrix} \pi & 2 & \pi & 2 \\ 2 & \pi & 2 & \pi \\ \pi & 2 & \pi & 2 \\ 2 & \pi & 2 & \pi \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} 4 \\ 4 \\ 4 \\ 4 \end{bmatrix}.$$

The matrix A has rank 2 and is not invertible. A is diagonalizable with

$$\tilde{D} = \begin{bmatrix} 2\pi + 4 & 0 \\ 0 & 2\pi - 4 \end{bmatrix}, \quad \tilde{Q} = \begin{bmatrix} 1/2 & -1/2 \\ 1/2 & 1/2 \\ 1/2 & -1/2 \\ 1/2 & 1/2 \end{bmatrix}.$$

The pseudo-inverse operator is

$$P = \tilde{Q}\tilde{D}^{-1}\tilde{Q}^T = \begin{bmatrix} p_1 & p_2 & p_1 & p_2 \\ p_2 & p_1 & p_2 & p_1 \\ p_1 & p_2 & p_1 & p_2 \\ p_2 & p_1 & p_2 & p_1 \end{bmatrix},$$

where

$$p_1 = \frac{\pi}{4\pi^2 - 16}, \quad p_2 = \frac{-2}{4\pi^2 - 16}.$$

The solution is $z = PB$:

$$z_1 = z_2 = z_3 = z_4 = \frac{2}{\pi + 2} \approx 0.38898\ .$$

Since all direction vectors have norm 1, we have

$$w_1 = w_2 = w_3 = w_4 = \frac{2}{\pi + 2} \approx 0.38898\ .$$

Table 1. Weights for each displacement vector set example in Figures 6 and 7.

Scenario	V	w
rectangular		
(A)	$d_1 = (1,0), d_2 = (0,1)$	$w_1 = w_2 \approx 0.77797$
(B)	$d_1 = (0,1), d_3 = (-1,0),$ $d_2 = (1,0), d_4 = (0,-1)$	$w_1 = w_3 \approx 0.38898$ $w_2 = w_4 \approx 0.38898$
(C)	$d_1 = (1,0), d_2 = (0,1),$ $d_3 = (-1,-1)$	$w_1 = w_2 \approx 0.72501$ $w_3 \approx 0.07635$
(D)	$d_1 = (0,1),$ $d_2 = (-1,-1), d_3 = (1,-1)$	$w_1 \approx 0.10784$ $w_2 = w_3 \approx 0.51266$
(E)	$d_1 = (0,1), d_5 = (-1,0),$ $d_2 = (1,1), d_6 = (-1,-1),$ $d_3 = (0,1), d_7 = (0,-1),$ $d_4 = (-1,1), d_8 = (1,-1)$	$w_1 = w_5 \approx 0.19624$ $w_2 = w_6 \approx 0.13876$ $w_3 = w_7 \approx 0.19624$ $w_4 = w_8 \approx 0.13876$
(F)	$d_1 = (1,0), d_9 = (-1,0)$ $d_2 = (2,1), d_{10} = (-2,-1)$ $d_3 = (1,1), d_{11} = (-1,-1)$ $d_4 = (1,2), d_{12} = (-1,-2)$ $d_5 = (0,1), d_{13} = (0,-1)$ $d_6 = (-1,2), d_{14} = (1,-2)$ $d_7 = (-1,1), d_{15} = (1,-1)$ $d_8 = (-2,1), d_{16} = (2,-1)$	$w_1 = w_9 \approx 0.12214$ $w_2 = w_{10} \approx 0.04542$ $w_3 = w_{11} \approx 0.04769$ $w_4 = w_{12} \approx 0.04542$ $w_5 = w_{13} \approx 0.12214$ $w_6 = w_{14} \approx 0.04542$ $w_7 = w_{15} \approx 0.04769$ $w_8 = w_{16} \approx 0.04542$
hexagonal		
(A)	$d_1 = \left(\frac{\sqrt{3}}{2}, \frac{1}{2}\right), d_2 = (0,1)$	$w_1 = w_2 \approx 0.74112$
(B)	$d_1 = (0,1),$ $d_2 = \left(-\frac{\sqrt{3}}{2}, -\frac{1}{2}\right), d_3 = \left(\frac{\sqrt{3}}{2}, -\frac{1}{2}\right)$	$w_1 \approx 0.52268$ $w_2 = w_3 \approx 0.52268$
(C)	$d_1 = (0,1), d_4\,(0,-1)$ $d_2 = \left(-\frac{\sqrt{3}}{2}, \frac{1}{2}\right), d_5 = \left(\frac{\sqrt{3}}{2}, -\frac{1}{2}\right)$ $d_3 = \left(-\frac{\sqrt{3}}{2}, -\frac{1}{2}\right), d_6 = \left(\frac{\sqrt{3}}{2}, \frac{1}{2}\right)$	$w_1 = w_4 \approx 0.26134$ $w_2 = w_5 \approx 0.26134$ $w_3 = w_6 \approx 0.26134$

5.4. Uniformly distributed directions

Consider the case of m unit-length displacement vectors uniformly distributed in angle:

$$(u_k, v_k) = \left(\sin\left(2\pi\,\tfrac{k-1}{m}\right), \cos\left(2\pi\,\tfrac{k-1}{m}\right)\right),\ k = 1, 2, \cdots, m.$$

Symmetry dictates that $w_1 = w_2 = \cdots = w_m$ and Eq. (3.2) reduces to

$$w = \frac{4}{\sum_{k=1}^{m} A(1,k)},$$

where the unique weight, w, is expressed as a function of the number of displacement vectors. If we consider $m \to \infty$, noting that $0 < \Delta < \frac{\pi}{2}$, we have

$$\lim_{m\to\infty} \left(\frac{\pi}{2m} \sum_{k=1}^{m} A(1,k) \right) = \int_0^{\frac{\pi}{2}} \left[(\pi - 2\theta)\cos\theta + 2\sin\theta \right] \, d\theta = 4.$$

Thus,

$$\lim_{m\to\infty} mw = \frac{\pi}{2}.$$

Several weight values are given in Table 2. Notice that $m = 3, 4, 6$ are scenarios which also appear in Table 1.

Table 2. Optimal anisotropic gradient weight w for m uniformly distributed unit displacement vectors. For $m \geq 3$, w is very closely approximated by $\frac{\pi}{2m}$.

m	w	$\frac{\pi}{2m}$	m	w	$\frac{\pi}{2m}$
1	1.27324	1.57080	7	0.22439	0.22440
2	0.63662	0.78540	8	0.19624	0.19635
3	0.52268	0.52360	9	0.17453	0.17453
4	0.38898	0.39270	10	0.15704	0.15708
5	0.31409	0.31416	11	0.14280	0.14280
6	0.26134	0.26180	12	0.13089	0.13090

6. Conclusion

The discrete anisotropic gradient and its integral are important in a variety of image processing applications and set boundary measure computations. We provided a method for computing weight factors for general anisotropic variation approximations of functions on $\mathbb{R}^2$. Optimal weights are found by minimizing the total integrated square error over all unit gradient functions. The method was developed in the framework of regular arrays, but applies more broadly to arbitrary finite discrete sampling strategies.

Acknowledgments

We would like to thank the anonymous referees and AIMS editor for their assistance in improving the clarity and focus of this paper.

Conflict of interest

The authors declare no conflict of interest.

References

1. V. Caselles, A. Chambolle, D. Cremers, et al. *An introduction to total variation for image analysis*, In: Fornasier M. Editor, *Theoretical Foundations and Numerical Methods for Sparse Recovery*, De Gruyters, 2010.
2. R. H. Chan, S. Setzer and G. Steidl, *Inpainting by flexible Haar-wavelet shrinkage*, SIAM J. Imaging Sci., **1** (2008), 273–293.
3. T. F. Chan and S. Esedoḡlu, *A multiscale algorithm for Mumford-Shah image segmentation*, UCLA CAM Report 03-57, 2003.
4. H. Chen, C. Wang, Y. Song, et al. *Split Bregmanized anisotropic total variation model for image deblurring*, J. Vis. Commun. Image R., **31** (2015), 282–293.
5. R. Choksi, Y. V. Gennip and A. Oberman, *Anisotropic total variation regularized L1-approximation and denoising/deblurring of 2D bar code*, Inverse Probl. Imag., **5** (2010), 591–617.
6. L. Condat, *Discrete total variation: New definition and minimization*, SIAM J. Imaging Sci., **10** (2017), 1258–1290.
7. V. Duval, J. F. Aujol and Y. Gousseau, *The TVL1 model: A geometric point of view*, J. Multiscale Model. Simulat., **8** (2009), 154–189.
8. S Esedoḡlu and S. J. Osher, *Decomposition of images by the anisotropic Rudin-Osher-Fatemi model*, Commun. Pure Appl. Math., **57** (2004), 1609–1626.
9. D. Goldfarb and W. Yin, *Parametric maximum flow algorithms for fast total variation minimization*, SIAM J. Sci. Comput., **31** (2009), 3712–3743.
10. T. Goldstein and S. Osher, *The split Bregman method for L1-regularized problems*, SIAM J. Imaging Sci., **2** (2009), 323–343.
11. Y. Lou, T. Zeng, S. Osher, et al. *A weighted difference of anisotropic and isotropic total variation model for image processing*, SIAM J. Imaging Sci., **8** (2015), 1798–1823.
12. S. P. Morgan and K. R. Vixie, *L^1TV computes the flat norm for boundaries*, Abstr. Appl. Anal., **2007** (2007), Article ID 45153.
13. L. B. Montefusco, D. Lazzaro and S. Papi, *Fast sparse image reconstruction using adaptive nonlinear filtering*, IEEE Trans. Image Process., **20** (2011), 534–544.
14. H. A. Moon and T. Asaki, *A finite hyperplane traversal Algorithm for 1-dimensional L^1pTV minimization, for* $0 < p \leq 1$, Comput. Optim. Appl., **61** (2015), 783–818.
15. H. T. Nguyen, M. Worring and R. V. D. Boomgaard, *Watersnakes: Energy-driven watershed segmentation*, IEEE T. Pattern Anal., **25** (2003), 330–342.

16. D. Strong and T. F. Chan, *Edge-preserving and scale-dependent properties of total variation regularization*, Inverse Probl., **19** (2003), S165–S187.