Angular Regularization of Vector-Valued Signals

Kevin M. Holt
Varian Medical Systems
Lincolnshire, IL, USA
Email: kevin.holt@varian.com

Abstract—A new class of pair-wise regularizers is proposed for vector-valued signals, including several new regularizers that penalize the angular difference between neighboring vectors. The pair-wise model is a generalization of the conventional difference model that includes discretized Tikhonov, L_1, and Total Variation (TV) as special cases. Whereas Tikhonov and TV regularization have become popular for encouraging signal values to be smooth or piecewise smooth, respectively, angular regularizers are proposed to encourage vector directions to be smooth or piecewise smooth. The regularizers are simple, effective, and general, making them easily applicable to a wide array of problems. Experimental results demonstrate the method’s effectiveness in denoising vector fields and color images.

I. INTRODUCTION

Regularization and variation models are important tools in many modern algorithms for signal processing. Some common forms include Tikhonov and L_2 regularization, which can encourage smooth solutions to ill-posed inverse problems, as well as L_1 and Total Variation (TV) regularization, which provide a very general method of encouraging piecewise smooth solutions without penalizing hard edges [1]. There have been numerous attempts to generalize such regularizers to vector-valued problems. The generalization of Tikhonov regularization to vector problems is straightforward, and TV has two obvious generalizations, Channel-by-Channel TV, which is the linear sum of the TV for each vector component, and Vector TV, which is a local quadrature sum of the TV for each component [2], [3]. One area of importance for vector field regularization is optical flow, where the classic Horn & Schunck method uses Tikhonov regularization on the gradient of the displacement field [4], and to allow discontinuities, subsequent works have traded Tikhonov for TV regularization [5]. Another important application is color image processing, where sometimes regularization is performed after first applying a transform to either decorrelate the RGB-vector components [6] or transform them onto a more perceptually relevant color space [2], [7]. However, these transform approaches do not generalize easily to other problems where no suitable transform is available.

Another approach, rather than regularizing vector values directly, is to split the regularization into a magnitude part and a vector part. Magnitude can be readily handled with scalar regularizers, and there are two main approaches to handling the direction component. One is to consider the vector as the product of magnitude and a normalized direction vector, and to regularize the direction vectors while restricting them to the surface of a unit sphere [8], [9]. However, we argue that for many vector-processing applications, the unit-norm constraint is artificial and not actually required by the application. Another approach is to treat the unnormalized vectors directly, and penalize a measure of the angle between neighboring vectors. This last approach is the one proposed in this paper. While a literature search turned up very little in this area, one exception was the squared norm of the cross-product (CP_2), which has been proposed as a penalty between neighboring pixels [10] and in a different form as a penalty between a reconstructed and prior image [11]. This angle-penalty approach avoids the complication of optimization along a manifold, can be made well-behaved for very small vectors, and provides more drastic alternatives to conventional distance measures.

In this work, we note that most existing regularizers are given in difference form. By respecifying them in pair-wise form, a whole new class of regularizers becomes available, including several new regularizers that penalize angle between neighbors. In Section II, pair-wise regularization is developed, and in Section III some new angle-based regularizers are presented (including CP_2 as a special case). Section IV then gives results using the proposed regularizers for denoising synthetic vector fields, as well for denoising color images. We find that the regularizers are quite simple yet effective, and can easily be extended to a wide range of problems. Section V then gives some conclusions and discussion.

II. GENERALIZED PAIR-WISE REGULARIZATION

Let F be a signal, and F_n be its value at index n. The signal could be, for example, a 1D audio signal, a 2D image, or a 3D volume. Let D(F) be some cost that we wish to minimize — usually it is some type of distance from an imperfect measured signal. Thus the direct problem is to minimize D(F), and the regularized problem is then to minimize D(F) + λR(F), where λ is some Lagrange multiplier. The function R is designed to be small when F satisfies some a priori notion of a “good” signal, and large otherwise. Often, such as with TV, R might be inspired by a functional of a continuous function, but then implemented using a discrete approximation. Though usually presented differently, many of the discrete implementations fall under the following form

\[ R = \sum_n \left( \sum_{m \in \mathcal{N}(n)} h_{n,m} \rho^q (F_n - F_m) \right)^{1/q} \]  (1)
TABLE I: Common scalar regularizers

<table>
<thead>
<tr>
<th>Name</th>
<th>Difference penalty</th>
<th>Pair-wise penalty</th>
<th>( q )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tikhonov / L2</td>
<td>(</td>
<td>e</td>
<td>^2)</td>
</tr>
<tr>
<td>L1</td>
<td>(</td>
<td>e</td>
<td>)</td>
</tr>
<tr>
<td>Total Variation</td>
<td>(</td>
<td>e</td>
<td>)</td>
</tr>
</tbody>
</table>

TABLE II: Some pair-wise vector regularizers

<table>
<thead>
<tr>
<th>Name</th>
<th>Penalty ( r(a, b) )</th>
<th>( p )</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>L_p^q</td>
<td>(</td>
<td>a - b</td>
<td>_p^q)</td>
</tr>
<tr>
<td>M_p^q</td>
<td>(</td>
<td></td>
<td>a</td>
</tr>
<tr>
<td>Cosp</td>
<td>(1 - (\frac{a \cdot b}{</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>CFp</td>
<td>(</td>
<td></td>
<td>a</td>
</tr>
<tr>
<td>CBp</td>
<td>(</td>
<td></td>
<td>a</td>
</tr>
<tr>
<td>CPp</td>
<td>(</td>
<td>a \times b</td>
<td>^p)</td>
</tr>
</tbody>
</table>

where \( \mathcal{N}(n) \) is the set of indices that are considered neighbors to index \( n \), \( h \) is a weighting kernel, \( q \) is some exponent, and \( r(e) \) is what we will call a difference penalty. It is easy to see that any regularizer using difference penalties can be equivalently written as

\[
R = \sum_n \left( \sum_{m \in \mathcal{N}(n)} h_{n,m} r^q (F_n, F_m) \right)^{1/q}
\]

where \( r \) is now a pair-wise penalty. Table I shows several common penalties for scalar-valued signals, including both their difference and pair-wise forms. Note that for these penalties, \( r(a, b) = 0 \) when \( a = b \), and \( r > 0 \) otherwise. We can generalize this to say that \( r(a, b) \approx 0 \) when \( a \) and \( b \) are very similar, and \( r(a, b) \) is large when \( a \) and \( b \) are dissimilar.

Now consider when \( F_n \) is a vector of length \( K \), and \( F(k) \) is a scalar image that we call the \( k \)th channel of \( F \). Channel-by-channel regularization can be performed by applying (1) or (2) separately to each channel and summing the results. Another approach is to apply (1) or (2) directly to \( F \) by replacing each \( \| \cdot \| \) in Table I with a \( \| \cdot \|_1 \). This approach can sometimes be enhanced by replacing \( r(F_n, F_m) \) with \( r(TF_n, TF_m) \), where \( T \) is some transform. Another approach is to penalize only the difference in magnitudes by using \( r(a, b) = \tilde{r} (\|a\|, \|b\|) \), where \( \tilde{r} \) is a conventional scalar penalty (say, from Table I).

III. ANGLE-BASED REGULARIZERS

An important benefit of (2) is the ability to encourage neighboring vectors to be parallel. Let us first define some types of parallelism. For simplicity, assume for the moment that \( a \neq 0 \) and \( b \neq 0 \). Then we say that vectors \( a \) and \( b \) are

- Forward-parallel (FP) if \( a = ab \) for some \( a > 0 \).
- Anti-parallel (AP) if \( a = ab \) for some \( a < 0 \).
- Bi-parallel (BP) if \( a \) and \( b \) are FP or AP.

Similarly, we define two types of angle-based penalties:

- If \( r(a, b) \geq 0 \), with equality if and only if \( a \) and \( b \) are BP, then we call \( r \) a BP penalty.
- If \( r(a, b) \geq 0 \), with equality if and only if \( a \) and \( b \) are FP, then we call \( r \) an FP penalty.

Thus an FP penalty encourages vectors to point in the same (i.e. FP) direction, and a BP penalty encourages vectors to point in the same (i.e. FP) or opposite (i.e. AP) direction.

Next consider the Cauchy-Schwarz inequality,

\[
- \|a\| \cdot \|b\| \leq \|a\| \cdot \|b\| \quad (3)
\]

where the left side is equality when \( a \) and \( b \) are AP, the right side is equality when \( a \) and \( b \) are BP, exactly one of the two sides are equality when \( a \) and \( b \) are BP and non-zero, and both sides are equality when \( a = 0 \) or \( b = 0 \). Therefore, one can use (3) to judge the type of an arbitrary penalty \( r \). If a penalty has \( r = 0 \) only when the right side of (3) is equality, then the penalty is FP. If a penalty has \( r = 0 \) only when either side of (3) is equality, then the penalty is BP.

A number of regularizers can also be found by manipulating (3). Some examples are shown in Table II. The \( CF \) and \( CB \) forms come from subtracting the middle of (3) from the sides. The \( Cos \) form is essentially computer science’s vector-cosine similarity measure, and can be derived just like \( CF \), only first normalizing (3) by its right side. The \( CP \) form is based on the vector cross product, since \( a \times b = 0 \) only when \( a \) and \( b \) are BP. From the well-known identities,

\[
\|a\|^2 \cdot \|b\|^2 - (a \cdot b)^2 = \|a \times b\|^2 = \sum_{1 \leq k \leq \ell \leq K} (a(k)b(\ell) - a(\ell)b(k))^2 \quad (4)
\]

we see that \( CB_p^2 \) (the left side), the \( CP_p^2 \) penalty of [10] (the middle), and the spectral regularizer of [11] (the right side) are all identical and are BP regularizers.

IV. EXPERIMENTS

A. Synthetic Vector Field

Figure 1a shows a vector field that was artificially created for demonstration purposes, with \( x \) and \( y \) component values ranging from approximately -1 to 1. Figure 1b then shows this same field with the addition of white Gaussian noise with \( \sigma = 0.2 \). Using the MSE fidelity term \( D(F) = \sum_{n,k} (F_n(k) - F_n^{\text{noise}}(k))^2 \) (where \( N \) is the number of pixels), we then denoise Figure 1b by minimizing \( R(F) \) subject to \( D(F) \leq (0.15)^2 \). When \( r \) contained a norm operator, the operator was regularized to \( \|x\|^2 = \sqrt{x^T x + \delta} \) with \( \delta = 10^{-8} \), to permit derivative-based optimization. The optimization was performed from an all-zero initialization with a conjugate gradient search [12] using \texttt{minFunc} [13] to minimize the Lagrangian cost, using a gradient derived analytically from the appropriate formula in Table II. The search was repeated for different \( \lambda \) values, keeping the solution from the largest \( \lambda \) value that satisfied the constraint on \( D \).

The results are shown in Figures 1c-d for the \( L_1^1 \) and \( CF_1^1 \) regularizers (both using \( q = 1 \)). One can see that the
magnitudes are much smoother for the $L_1^1$ penalty than $CF_{1/2}^1$, which is not surprising since $CF_{1/2}^1$ does not encourage smooth magnitudes. To see the direction component, Figures 1a-d were normalized to unit norm, and are shown again in Figures 1e-h. This was also done for four more regularizers, whose normalized results are shown in Figures 1i-j. We see that $CP_2$, $Cos_2$, $Cos_1$, and $CF_1$ all exhibit some type of swirling patterns, where sharp changes in direction are blurred out in a way analogous to how Tikhonov regularization blurs scalar edges. However, $Cos_1$ and $CF_1$ also allow some sharp transitions. At first glance, it appears that $Cos_2$ and $CP_2$ also allow some sharp transitions, but on closer inspection, these transitions tend to be close to $180^\circ$, an artifact (often unwanted) of their BP nature. Note that $L_1^1$ and the FP regularizers do not exhibit this characteristic. We also see that $L_1^1$, $CP_2$, and $CF_1$ all contain some local wiggles for vectors with small magnitude. Evidently, for these pixels, the fidelity criterion has overwhelmed the regularization penalty. In some applications, this may actually be a desired effect, as the directions of very small vectors may be unimportant. $CF_{1/2}^1$ contains very few outliers, and did an excellent job of preserving sharp transitions. Thus we conclude that when smooth fields are desired, $Cos_1$ or $CF_1$ may be the desired approach (depending on how noise is to be handled), and to preserve hard transitions, $CF_{1/2}^1$ appears to be the best choice.

**B. Color Image Denoising**

Figure 2 shows an example of regularization for denoising of color images. The original image is a small portion of lena, which is then corrupted by white Gaussian noise whose standard deviation is 10% of full scale. The images were denoised by minimizing the same cost functional as in Section IV-A. The cost was again minimized using $minFunc$ from an all-zero initialization.

Figure 2c shows the result from transforming the RGB data to $LC_1C_2$ colorspace then using $L_1^1$ regularization on the result (this mimics [6]). Figures 2d-f show the proposed angular regularizers. The $CF_{1/2}^1$ regularizer encourages smooth vector directions without regard to magnitudes, so Figure 2d has noisy intensities, but its hues are smooth except for some sharp edges. Conversely, the $M_1^1$ regularizer encourages smooth magnitudes without regard for direction, so Figure 2e has smooth intensities but noisy hues. The two regularizers are then combined as $CF_{1/2}^1 + M_1^1$ by adding their $R$ functions, and the result is piecewise smooth in both intensity and hue.

For fair comparison, the $\lambda$ weights were chosen for Figures 2c and 2f so that both images have the same RMS distance from the original image. Subjectively, the hues appear slightly more consistent in 2f. To obviate this, the hues (as defined by [14]) are shown in Figures 2g-i, with black mapped to hue -0.25 (purple) and white mapped to hue 0.12 (gold). We see that for both methods, some small details in hue are obliterated. However, we see that $CF_{1/2}^1 + M_1^1$ indeed provides a more constant hue than the conventional $L_1^1$ approach. This demonstrates that explicit smoothing of vector directions can be superior to smoothing vector values, even when transforming to a carefully chosen colorspace.

**V. Conclusions**

The angular regularizers presented here have simple forms but powerful effects. They are fairly straightforward to optimize, and are applicable to a wide range of problems without needing special knowledge such as color space. While we demonstrate their effectiveness for simple vector fields (such as for optical flow or fingerprint recognition) and for color image denoising and restoration, the methods presented could also be easily applied to more diverse problems such as hyperspectral fusion, Mumford-Shah type segmentation of color images, or medical perfusion or diffusion tensor imaging. In experiments, we find that the $Cos_1$ or $CF_1$ regularizers are best for obtaining smooth direction fields, and $CF_{1/2}^1$ is best for obtaining piecewise smooth directions. We also note the importance of using a FP regularizer when FP-type parallelism is important (such as for omnidirectional vectors), whereas the FP/BP distinction is unimportant for vectors that all fall within one quadrant or orthant, such as color images. Future work may include additional regularizers, a better theoretical understanding of the performance of angular regularization, more efficient optimization algorithms, and additional applications.

**REFERENCES**


Fig. 1: Effect of different regularizers in the denoising of a vector field.

Fig. 2: Denoising of Lena image (note a-f are intended to be viewed in color).