Polyakov Action for Efficient Color Image Processing

Guy Rosman Xue-Cheng Tai Lorina Dascal
Ron Kimmel
June 14, 2010

Abstract

The Laplace-Beltrami operator is an extension of the Laplacian from flat domains to curved manifolds. It was proven to be useful for color image processing as it models a meaningful coupling between the color channels. This coupling is naturally expressed in the Beltrami framework in which a color image is regarded as a two dimensional manifold embedded in a hybrid, five dimensional, spatial-chromatic \((x, y, R, G, B)\) space.

The Beltrami filter defined by this framework minimizes the Polyakov action, adopted from high-energy physics, which measures the area of the image manifold. Minimization is usually obtained through a geometric heat equation defined by the Laplace-Beltrami operator. Though efficient simplifications such as the bilateral filter have been proposed for the single channel case, so far, the coupling between the color channel posed a non-trivial obstacle when designing fast Beltrami filters.

Here, we propose to use an augmented Lagrangian approach to design an efficient and accurate regularization framework for color image processing by minimizing the Polyakov action. We extend the augmented Lagrangian framework for total variation (TV) image denoising to the more general Polyakov action case for color images, and apply the proposed framework to denoise and deblur color images.

1 Introduction

Nonlinear diffusion filters based on variational formulations have been extensively used in the last two decades for different tasks in image processing. Numerical schemes implementing them are designed with an emphasis on accuracy, stability and computational efficiency.

The Beltrami framework [36] describes a regularizing functional, well-suited for color image processing. The framework considers the image as a 2-D manifold

*This research was supported by the Israeli Science Foundation (ISF) grant no. 623/08, and by the Office of Naval Research (ONR) grant no. N00014-06-1-0978.
embedded in a hybrid spatial-feature space. Regularization of the image in this framework is expressed as minimization of area surface. The Beltrami filter is strongly related to the bilateral filter (see [45], [34], [2], [40], [35], [16], [4]), as well as to the nonlocal means filter, which was proposed in [1] and shown to be very effective in denoising gray-scale and color images. Minimization of the associated functional is usually obtained by considering its Euler-Lagrange equation as a gradient descent time evolution [36]. This evolution, expressed as an explicit scheme, is limited in its time-step, resulting in slow convergence and high computational complexity. Another possibility [3] is to obtain a fixed-point iteration for the Euler-Lagrange equation and solve the resulting linear system. Recently, several approaches were suggested for improving the speed of computation of minimizers for the Polyakov action [28]. These include an approximation using the short time kernel of the Beltrami operator [38], as well as employing vector extrapolation techniques [30], or operator splitting methods [14] for solving the flow equations. For the case of gray-scale images, the projection-based method of Chambolle [11] has been extended to the Polyakov function [6], but no suggestions were made for the vectorial case.

In [39], the augmented Lagrangian method [20, 29] is used to perform total variation regularization of images. In this paper we propose to use a similar constrained optimization approach for regularization of color images. Instead of discretizing the continuous optimality condition or the resulting minimizing flow, we discretize and minimize the Polyakov action itself. This approach is also similar, in a sense, to the one taken by Bruckstein et. al. [8] confronting problems in elastica. The resulting method is shown to be more efficient and accurate for image denoising and deblurring, compared to existing methods for Beltrami regularization in image processing. In Section 2 we review the Beltrami framework for color image regularization. In Section 3 we extend the coupled constrained optimization approach demonstrated in [39] to regularize color images by the Polyakov action. In Section 4 we display results of using our method for deblurring color images. Section 5 concludes the paper.

2 The Beltrami Framework

We now briefly review the Beltrami framework for non-linear diffusion in computer vision [23, 36, 37, 46]. The basic notions used in this introduction are taken from Riemannian geometry, and we refer the reader to [24, 15] for further reading.

In the Beltrami framework, images are expressed as maps between Riemannian manifolds of a higher dimensional space. Denote such a map by $X : \Sigma \rightarrow M$, where $\Sigma$ is a two-dimensional manifold, parameterized by global coordinates $(\sigma^1, \sigma^2)$. We denote by $M$ the spatial-feature manifold, embedded in $\mathbb{R}^{d+2}$, where $d$ is the number of image channels. For example, a gray-level image can be represented as a 2D surface embedded in $\mathbb{R}^3$. The map $X$ in this case is $X(\sigma^1, \sigma^2) = (\sigma^1, \sigma^2, I(\sigma^1, \sigma^2))$, where $I$ is the image intensity. For color images, $X$ is given by $X(\sigma^1, \sigma^2) = (\sigma^1, \sigma^2, I^1(\sigma^1, \sigma^2), I^2(\sigma^1, \sigma^2), I^3(\sigma^1, \sigma^2))$. 
where $I^1, I^2, I^3$ are the three components of the color vector (for example, red, green, blue for the RGB color space). The canonical choice of coordinates $\sigma^1$ and $\sigma^2$ in image processing uses Cartesian coordinates aligned with the $x$ and $y$ directions.

Next, we choose a Riemannian metric on this surface. Its components are denoted by $g_{ij}$. We denote the elements of the inverse of the metric by superscripts $g^{ij}$, and the determinant by $g = \det(g_{ij})$.

Once images are defined as embedding of Riemannian manifolds, it is natural to look for a measure on this space of embedding maps. Denote by $(\Sigma, g)$ the image manifold and its metric, and by $(M, h)$ the space-feature manifold and its metric. The functional $S[X]$ characterizes the mapping $X : \Sigma \rightarrow M$, and is defined to be

$$S[X, g_{ij}, h_{ab}] = \int d^m \sigma \sqrt{g} ||dX||^2_{g,h},$$

where $m$ is the dimension of $\Sigma$, $g$ is the determinant of the image metric, and the range of indices is $i, j = 1, 2, \ldots, \dim(\Sigma)$ and $a, b = 1, 2, \ldots, \dim(M)$. The integrand $||dX||^2_{g,h}$ is given by $||dX||^2_{g,h} = (\partial_{\xi_1} I^a) g^{ij} (\partial_{\xi_2} I^b) h_{ab}$. We use here Einstein’s summation convention: identical indices that appear up and down are summed over. This functional, for $\dim(\Sigma) = 2$ and $h_{ab} = \delta_{ab}$, is known in string theory as the Polyakov action [28], and extends the action functional to the relativistic case.

The elements of the induced metric for color images (where $h_{ab} = \delta_{ab}$ and where the color coordinates are assumed to be Cartesian) are

$$G = (g_{ij}) = \begin{pmatrix}
1 + \beta^2 \sum_{a=1}^{3} (I^a_{x_1})^2 & \beta^2 \sum_{a=1}^{3} I^a_{x_1} I^a_{x_2} \\
\beta^2 \sum_{a=1}^{3} I^a_{x_1} I^a_{x_2} & 1 + \beta^2 \sum_{a=1}^{3} (I^a_{x_2})^2
\end{pmatrix},$$

where a subscript of $I$ denotes a partial derivative and the parameter $\beta > 0$ determines the ratio between the spatial and color coordinates.

The usual way of minimizing the functional $S$ involves time evolution of the image according to the Euler-Lagrange equations. These read, assuming a Euclidean embedding space,

$$0 = \frac{1}{\sqrt{g}} h^{ab} \frac{\delta S}{\delta I^a} = \frac{1}{\sqrt{g}} \text{div} (D \nabla I^a),$$

where the matrix $D = \sqrt{g} G^{-1}$. See [36] for explicit derivation. The operator that acts on $I^a$ is the natural generalization of the Laplacian from flat spaces to manifolds, it is called the Laplace-Beltrami operator, and is denoted by $\Delta_g$.

The time evolution forms the continuous equivalent of a gradient descent minimization process, according to the system of PDEs

$$I^a_t = -\frac{1}{\sqrt{g}} \frac{\delta S}{\delta I^a} = \Delta_g I^a,$$
with reflective boundary conditions and a smooth initial solution \( I^a|_{t=0} = I^a_0 \). Evolution according to these equations give us the Beltrami scale-space. For Euclidean embedding, the functional in Eq. 1 reduces to

\[
S(X) = \int \sqrt{g} \, da^1 \, da^2,
\]

where

\[
g = \det(G) = 1 + \beta^2 \sum_{a=1}^{3} \|\nabla I^a\|^2 + \frac{1}{2} \beta^4 \sum_{a,b=1}^{3} \|\nabla I^a \times \nabla I^b\|^2.
\]  

(4)

The role of the cross product term \( \sum_{a,b=1}^{3} \|\nabla I^a \times \nabla I^b\|^2 \) in the minimization was explored in [23], see also [22]. It penalizes deviations from the Lambertian model of image formation by looking at misalignments of the gradient directions in the various color channels. Taking large values of \( \beta \) therefore makes sense as we would expect both \( \sum_{a,b=1}^{3} \|\nabla I^a \times \nabla I^b\|^2 \) and \( \sum_{a=1}^{3} \|\nabla I^a\|^2 \) to be small.

We note that another generalization of the functional can be of the form

\[
\int \left( \beta_1 + \beta_2 \sum_{a=1}^{3} \|\nabla I^a\|^2 + \beta_3 \sum_{a,b=1}^{3} \|\nabla I^a \times \nabla I^b\|^2 \right),
\]

(5)

where the coefficients \( \beta_1, \beta_2, \beta_3 \) are set to arbitrary constants. While this approach cannot be explained by the physical interpretation of the Polyakov functional, it makes sense in terms of color image restoration. This form will be used in the results shown hereafter, in Figures 5–8.

The geometric functional \( S \) can be used as a regularization term for color image processing. In the variational framework, the reconstructed image is the minimizer of a cost-functional. Image functionals using the Beltrami regularization can be written in the general form

\[
\Psi = \frac{\alpha}{2} \sum_{a=1}^{3} \|KI^a - I^a_0\|^2 + S(X),
\]

where \( K \) is a bounded linear operator. In the denoising case, \( K \) is the identity operator \( Ku = u \), and in the deblurring case, \( Ku = k * u \), where \( k(x, y) \) is the blurring kernel (often assumed to be Gaussian). The parameter \( \alpha \) controls the smoothness of the solution. This functional have been used for image denoising [36, 3], blind deconvolution [21]. Its relation to active contours models was explored, along with an extension for multiscale segmentation in [7]. We introduce an approach for finding an optimum for the discretized form of the functional \( \Psi \) using the augmented Lagrangian method.
3 An augmented Lagrangian approach for Beltrami regularization

In recent years, several attempts have been made of optimizing total variation functionals using auxiliary variables (we refer the reader to [12, 10, 11, 26, 44, 47, 39, 17, 48, 32, 42, 33] and references therein). The functional is treated as an optimization problem (with certain properties being exploited), instead of using the Euler-Lagrange equations to form the traditional time-evolution [31].

These algorithms achieved great accuracy and efficiency, and are considered to be among state-of-the-art methods for TV restoration.

Specifically, in [39], total variation regularization is obtained by decoupling the optimization problem

$$\min_u \int |\nabla u| + \alpha \frac{1}{2} \|Ku - f\|^2$$

into a constrained optimization problem

$$\min_{u, q} \int |q| + \alpha \frac{1}{2} \|Ku - f\|^2 \quad \text{s.t.} \quad q = \nabla u,$$

where $q$ is an auxiliary field, parallel to the gradient of $u$. This constraint is then incorporated using an augmented Lagrangian penalty function of the form $\rho(u, q) = \mu^T (\nabla u - q) + \frac{r}{2} \|\nabla u - q\|^2$. The penalty is used to enforce the constraint $q = \nabla u$, without making the problem severely ill-conditioned, resulting in the saddle-point problem

$$\min_u \max_{q} \int \left\{ |q| + \alpha \frac{1}{2} \|Ku - f\|^2 + \mu^T (q - \nabla u) + \frac{r}{2} \|q - \nabla u\|^2 \right\}.$$  

We now describe a similar construction for the Polyakov action. Again, it is important to stress we are minimizing the functional itself, rather than discretizing the resulting minimizing PDE as in [36, 21, 38, 3, 30, 14].

We deal with the case of color images, for which the regularization offered by the Beltrami framework is more meaningful. Specifically, we replace the gradient norm penalty used in total variation regularization by the action functional of Equations 1. This is done by replacing the first term in Equation 7 by the term

$$\int \sqrt{1 + \beta^2 \sum_{i \in \{R, G, B\}} \|q_i\|^2 + \beta^4 \sum_{i \in \{R, G, B\}} \sum_{j \neq i} \|q_i \times q_j\|^2},$$

where $\beta$ is the spatial-intensity aspect ratio, and $\{q_i\}_{i \in \{R, G, B\}}$ denote components of the auxiliary field $q$, which parallel the gradient of each of the image channels. We then trivially extend the rest of the functional to the vectorial (per-pixel) case, obtaining the following functional
\[
\mathcal{L}_{\text{BEL}}(u, q, \mu) = \int \left\{ \sqrt{1 + \beta^2 \sum_{i \in \{R,G,B\}} \|q_i\|^2 + \beta^2 \sum_{i \in \{R,G,B\}} \sum_{j \neq i} \|q_i \times q_j\|^2 + \sum_{i \in \{R,G,B\}} \mu_i^T (q_i - \nabla u_i) + \frac{\mu}{2} \|Ku - f\|^2 + \frac{\mu}{2} \sum_{i \in \{R,G,B\}} \|q_i - \nabla u_i\|^2} \right\},
\]

which corresponds to Beltrami regularization. The expressions optimizing \(u\) and \(\mu\) are replaced by their per-channel equivalents, \(\{u_i\}\) and \(\{\mu_i\}\), for \(i \in \{R,G,B\}\).

The augmented Lagrangian algorithm for regularizing an image using the Polyakov action is given as Algorithm 1.

**Algorithm 1** Augmented Lagrangian optimization of the Beltrami framework

1. \(\mu^0 \leftarrow 0\)
2. for \(k=0,1,\ldots\) do
3. \(\{u_i^k\}, \{q_i^k\} \leftarrow \arg\min_{\{u_i\}, \{q_i\}} \mathcal{L}_{\text{BEL}}(\{u_i\}, \{q_i\}, \{\mu_i^k\})\) (10) according to Equation 11 and Subsection 3.1.
4. Update the Lagrange multipliers \(\{\mu_i\}\) according to Equation 12.
5. end for

The variables \(u_i\) and \(\mu_i\) are optimized as suggested in [39]. At each inner iteration \(k\), \(\{u_i^k\}, \{q_i^k\}\) is updated in the Fourier domain using the same analytic expression as in [39],

\[
u_i^k = \mathcal{F}^{-1} \left\{ \frac{\alpha \mathcal{F}(K^+) \mathcal{F}(D_x^+) - \mathcal{F}(D_y^+) (\mu_i^k)' + r (p_i^k)' - \mathcal{F}(D_x^+) (\mu_i^k)' + r (q_i^k)'}{\alpha \mathcal{F}(K^+) \mathcal{F}(K) - r \mathcal{F}(\Delta)} \right\},
\]

(11)

where \(D_x^+, D_y^+, \Delta\) denote the backward derivative along the \(x\) and \(y\) directions, and the Laplacian operator, respectively, and \(\mathcal{F}\{\cdot\}, \mathcal{F}^{-1}\{\cdot\}\) denote the Fourier transform and its inverse, respectively. We explicitly write \(q_i = (p_i, q_i), i \in \{R,G,B\}\), for the components of \(q\) of each color channel, approximating its \(x\) and \(y\) derivatives, computed using backward differences.

We note that the optimization of \(u\) using the Fourier domain resembles, in a sense, the approach taken by [27]. Since, however, it is done with respect to the dual field, iteratively, its effect is suited to the nonlinear nature of the Beltrami flow. We expand upon this relation in Subsection 4.3. An update rule for the dual field \(q\) of each channel is described in Subsection 3.1.

The Lagrange multipliers \(\mu_i\) are updated so as to approximate the optimal Lagrange multipliers (see [5], Section 2, for an in-depth discussion),

\[
(\mu_i)^k = (\mu_i)^{k-1} + r (q_i - (\nabla u_i)^k).
\]

(12)
Finally, the coefficient $r$ is updated between each outer iteration by multiplying $r$ with a scalar $\gamma > 1$. Unlike the quadratic penalty method, $r$ needs not become very large, thus avoiding ill-conditioning of the functional $L_{BEL}(u, q, \mu)$.

### 3.1 Updating the dual field $q$

For optimizing $q$, a short inner-loop of a fixed-point solver with iterative reweighted least squares (IRLS) allows us to efficiently obtain a solution. In numerical experiments, optimization over $q$ takes less than half the computational time of the optimization. Furthermore, since this problem is solved per pixel, it can easily be solved in parallel on commodity GPU hardware leading to further speed-up.

The update of $q_i = (p_i, q_i), i \in \{R, G, B\}$, the components of $q$ at each pixel, is done by optimizing the function

$$\sqrt{1 + \beta^2 \sum_i (p_i^2 + q_i^2) + \beta^4 \sum_{j \neq i} \frac{1}{2} (p_i q_j - q_i p_j)^2} + \frac{r}{2} \sum_i \|q_i - (\nabla u_i)\|^2 + \sum_{i \in \{R, G, B\}} (\mu_i^k)^T (q_i - \nabla u_i),$$

where $(\nabla u)_i = ((u_i)_x, (u_i)_y)^T$ denote the components of the various image channel gradients. Each fixed-point sub-iteration updates the elements of $q_i$ according to the IRLS approach, by replacing the square root with a weighted version of the quadratic expression inside. Thus, the equation used to update $p_i$ is of the form

$$\frac{2 \left( \beta^2 + \frac{\beta^4}{2} \sum_{j \neq i} q_j^4 \right)}{1 + \beta^2 \sum_i (p_i^2 + q_i^2) + \beta^4 \sum_{j \neq i} (p_i q_j - q_i p_j)^2} p_i + 2 \left( -\frac{\beta^4}{2} \sum_{j \neq i} q_j p_j q_i \right) + r (p_i - (u_i)_x) + (\mu_i^k)_x = 0,$$

and similarly for $q_i$. Next, we freeze the denominator and solve for each component of $q_i$, thereby obtaining a fixed-point iteration of the form

$$p_i^l = \left( 2w_i^{l-1} \left( \beta^2 + \frac{\beta^4}{2} \sum_{j \neq i} q_j^4 \right) + r \right) p_i^l = \left( 2w_i^{l-1} \left( \beta^4 \sum_{j \neq i} q_j^4 q_i^4 \right) + r (u_i)_x - (\mu_i^k)_x \right),$$

where $w_i^{l-1}$ are the IRLS weights,

$$w_i^{l-1} = \left( 1 + \beta^2 \sum_i \left( (p_i^{l-1})^2 + (q_i^{l-1})^2 \right) + \beta^4 \sum_{j \neq i} \left( p_i^{l-1} q_j^{l-1} - q_i^{l-1} p_j^{l-1} \right)^2 \right)^{-1/2},$$

and $l$ denotes the IRLS iteration number. These iterations are repeated for all 6 elements of $q$ for several inner iterations, at each pixel.
4 Results

We now demonstrate the minimization of the Polyakov functional using the augmented Lagrangian method.

4.1 A Denoising Example

In Figure 1, results are shown for smoothing an image using various values of $\alpha$, with $K$ equal to the identity operator, and $\beta = \sqrt{4000} \approx 63.25$. This parallels samples along the Beltrami scale-space, which is defined by the Beltrami flow evolution equations. We used the same initialization of the penalty parameter, $r = 0.5$, for which in practice the constraints are satisfied after very few iterations. Fixed-point iterations over $q$ were limited to 2 inner and 2 outer (IRLS) iterations for each cycle. The number of outer iterations, updating $\mu$, in Figure 1 was 150, although fewer iterations suffice. The residual plot is shown in Figure 2, plotting the Euclidean norm of the difference between iterant vectors of $u$.

A comparison of the results of the augmented Lagrangian method and the locally one-dimensional (LOD) [43] and additive operator splitting (AOS) [25] methods shows that the augmented Lagrangian method results in faster convergence, as can be seen in Figure 3. In this experiment, $\alpha$ was set for optimal results for both the augmented Lagrangian and the splitting methods. The PSNR plot also demonstrates the more accurate discretization of the proposed method compared to the backward-forward discretization of the diffusion process. This can be easily seen in the preservation of edges and better removal of color artifacts seen in Figure 3. Experiments comparing our method to explicit scheme time evolution with backward-forward discretization led to similar conclusions.

Table 1 demonstrates the CPU-time required for several images (shown in Figure 4) for our algorithm, compared to Beltrami filtering with operator splitting techniques (in this case we used AOS splitting). The time-step $\Delta t$ was taken to be the largest possible so as to avoid instability of the splitting method or inaccurate operator approximation.

Since the solution obtained by discretizing the functional and by discretizing the resulting Euler-Lagrange equation is not expected to be the same, a different halting condition was used. After measuring the PSNR of each algorithm with respect to the original image, we measured the CPU time each algorithm took to gain 99% of the maximal rise in signal to noise ratio (SNR). While this measure does not provide a halting criteria in real applications, it does give us an objective measure of the time it takes for each algorithm to converge in practice where cost function value, iterant residual norm, or the norm of the EL equation all fail as objective halting criteria.

The speedups obtained are at least of a factor $\times 2$ compared to AOS splitting [25], which is one of the fastest methods for Beltrami regularization [14]. We note that the time-step allowed in the splitting algorithm was large enough to cause visible artifacts in the denoised image, and yet the augmented Lagrangian
method resulted in still more accurate results and a shorter computation time.

4.2 Deblurring Examples

The result of deblurring a color image using the Beltrami framework are shown in Figures 5–8. In Figures 5–7, The color image is blurred using a disc kernel of radius 5 pixels. In Figure 8, the point spread function (PSF) of the filter is given by

\[
k(x, y) = \frac{1}{1 + x^2 + y^2}, \quad x, y = -7, \ldots, 7.
\]  

(13)

This filter is often used for comparing deblurring operators, see for example [18].

The results of Beltrami-regularized deblurring are compared to standard deblurring algorithms available in Matlab, as well as to a deblurring based on the Block Matching and 3D Denoising (BM3D) algorithm [13], applied at each color channel, and to the FTVd package for total variation regularization [41, 44]. Where the algorithms require as input the noise level, it is supplied. Where the algorithms accept a different quality parameter, it is empirically set to minimize the mean squared error of the restoration. For Figures 5–8, we have chosen to use the more general form of the function, given in Equation 5. We set $\beta_1$ set to a very small positive constant. This choice reflects a preference of a flow without a linear diffusion part. Furthermore, we set $\beta_2 = \beta_3 = \frac{\beta_4}{20}$. This dampens the relative strength of the gradient coupling term, which may be less important as a prior in constant regions with noise.

The results clearly demonstrate the visually convincing and accurate deblurring obtained using the regularization offered by the Beltrami framework for natural color images. Beltrami regularization allowed us to obtain a slightly better result in terms of PSNR, compared to TV regularization. Beyond the PSNR reading, careful examination of the images show the tendency of Beltrami regularization to avoid artifacts which do not fit the appearance of natural images. This behavior includes the tendency of Beltrami regularization to discourage uneven coloring artifacts resulting from deblurred noise in color image, as can be seen in Figure 5, with magnified details shown in Figure 6. We note the same discrepancy between PSNR reading and visual results in color image processing has already been noted by Goldluecke et. al [19]. We simply iterate this word of caution here, and refer the reader to the images themselves.

4.3 On the Connection to the Bilateral Filter

As mentioned before, our usage of the Fourier domain in order to update the image can be related to fast bilateral filtering [27]. The bilateral filter, however, makes the implicit assumption of clusters of pixels [45], which is well suited to piecewise constant images. This assumption, however, is less appropriate where the distance between the clusters varies wildly, or where the image involves areas with a color gradient. While the latter constraint can be eased by changing our view from that of a weighted average to that of a regression problem [9], this
Figure 1: Smoothing, under various $\alpha$ values, of the Fruits image, with added Gaussian noise with $\sigma = 20$ intensity levels per channel.
Figure 2: The residual norm, $\|u^{k+1} - u^k\|_2^2$, as a function of iterations, for the augmented Lagrangian method for Beltrami regularization in Figure 1 ($\alpha = 0.15$).

Table 1: Comparison of maximal SNR values and CPU time required to complete 99% of the rise in SNR. The maximal SNR obtained was determined in advance by allowing the algorithms to converge.
Figure 3: A comparison of the results for LOD and AOS, and the augmented Lagrangian method, as well as the PSNR (for the whole image, compared to the uncorrupted image) plotted as a function of CPU time. The arrows in the images demonstrate gradient directions at each channel. The plot demonstrates the fast convergence of the augmented Lagrangian method, as well as a more accurate discretization.

Figure 4: Images used to compare the computational cost of the augmented Lagrangian and splitting-based Beltrami regularization. Left to right: (a) Astro image. (b) Fruits image. (c) Lion image (d) Monarch image.
Figure 5: Deblurring results obtained on an astronaut image with a disc blur filter of radius 5 and an added Gaussian noise of $\sigma = 5$. Left to right, top to bottom: (a) The original image. (b) The blurred image. (c) Deblurring using the Lucy-Richardson algorithm, as implemented by Matlab (d) BM3D-based deblurring. (e) Deblurring using the FTVd method. (f) Deblurring using the Beltrami / augmented Lagrangian algorithm.
Figure 6: Each row represents two regions zoomed-in from Figure 5. (a) The original image. (b) The blurred and noisy image. (c) TV-restoration results. (d) Beltrami / augmented Lagrangian restoration results.
Figure 7: Deblurring results obtained on the lion image blurred with a disc blur filter of radius 5 and an added Gaussian noise of $\sigma = 5$. Left to right, top to bottom: (a) The original image. (b) The blurred image. (c) Deblurring using the Lucy-Richardson algorithm, as implemented by Matlab. (d) BM3D-based deblurring. (e) Deblurring using the FTVd algorithm. (f) Deblurring using the Beltrami / augmented Lagrangian algorithm.
Figure 8: Deblurring results obtained on the Greek dome image (courtesy of [27]) blurred with a filter of a PSF as given in Equation 13 and an added Gaussian noise of $\sigma = 8$. Left to right, top to bottom: (a) The original image. (b) The blurred image. (c) Deblurring using a wiener filter, as implemented by Matlab. (d) BM3D-based deblurring. (e) Deblurring using the FTVd algorithm. (f) Deblurring using the Beltrami / augmented Lagrangian algorithm.
outlook does not pass in a straightforward manner into fourier domain solutions. Figure 9 demonstrates these limitations. When the differences between the color bands become too small, the bilateral filter fails to distinguish between them, resulting in a smeared appearance of the color bands.

Figure 9: Denoising results of striped color image corrupted by a Gaussian noise of $\sigma = 20$. Left to right, top to bottom: (a) The original image. (b) The corrupted image. (c) Denoising using a bilateral filter (d) Denoising using the Beltrami / augmented Lagrangian algorithm.

5 Conclusions

We presented an extension of the augmented Lagrangian method for color image processing with Beltrami regularization. Unlike existing techniques, the method discretizes the functional itself, rather than the resulting optimality conditions.
or minimizing flow. We present numerical examples demonstrating its efficiency and accuracy compared to existing techniques for variational regularization, and its effectiveness in image deblurring. In future work we intend to extend the use of Lagrange multipliers to allow a robust fidelity term [42], as well as explore other possible applications for our framework.

References


