

AN AUGMENTED-LAGRANGIAN APPROACH FOR THE BELTRAMI FRAMEWORK*

GUY ROSMAN[†], XUE-CHENG TAI[‡], AND RON KIMMEL[†]

Abstract. The Laplace-Beltrami framework provides useful tools for color image denoising and deblurring. The Beltrami filter is constructed by coupling, in a physically-motivated way, the color components of an image with its spatial coordinates. The image is thereby treated as a two dimensional surface embedded in the spatial-chromal hybrid space, for example (x, y, R, G, B) . The filter minimizes the area of the resulting surface that can be described by a Polyakov functional. This approach can be easily extended to higher dimensional domains and feature spaces.

The gradient descent process that defines the classical Beltrami filter can be replaced by a fast optimization approach for the associated Polyakov functional. Here, we demonstrate how an additional auxiliary field, in an augmented Lagrangian framework, defines a fast optimization method for Beltrami filtering and regularized deblurring. We demonstrate the efficiency of the method and its benefits in color image denoising and deblurring, including real-time implementation considerations on a parallel graphics processing unit.

Key words. Polyakov action, color images, PDE, diffusion, regularization

AMS subject classifications. 49M99,65K10,90C46

1. Introduction. Nonlinear diffusion filters 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. These filters are often related to the minimizing flow of a regularization function, with the relation between the PDE and variational view playing a significant role in optimization.

The Beltrami framework [37] describes a regularizing functional, well suited for color and vectorial image processing. In this framework, images are described as 2-manifolds embedded in a hybrid spatial-feature space.

Regularization of the image in this framework is expressed as minimization of area surface, for an area measure that stems from the embedding space metric. The Beltrami filter is strongly related to the bilateral filter (see [35], [2], [42], [36], [17], [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. Several extensions of the Beltrami framework consider different metrics by changing the embedding space and functional [6, 33, 39, 49], or the resulting minimizing flow [33, 5, 18, 44]. Minimization of the associated functional is usually obtained by considering its Euler-Lagrange equation as a gradient descent time evolution [37]. 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 by short time heat kernel [40], vector extrapolation techniques [30], or operator splitting methods [14] for solving the flow equations. For the case of gray-scale images, the projection-based

*

[†]Dept. of Computer Science, Technion - Israel Institute of Technology, 32000 Haifa, Israel,
rosman,ron@cs.technion.ac.il

[‡]Dept. of Mathematics, Bergen University, Johaness Brunsgate 12, 5007 Norway, tai@mi.uib.no

method of Chambolle [11] has been extended to the Polyakov function [8], but no suggestions were made for the vectorial case.

In [41], the augmented Lagrangian method [21, 29] is used to perform total-variation (TV, [34]) regularization of images. This inspires a similar approach for the Beltrami framework and its associated functional, which we describe in this paper. Instead of discretizing the continuous optimality condition or the resulting minimizing flow, we discretize and minimize the Polyakov action itself. We do so by adding an auxiliary variable approximating the gradient and formulating smoothness in terms of this variable. 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.

This paper extends a previous conference submission ([31], see also [32]) with a more complete description of the method and with the addition of experiments demonstrating robust data terms and implementations on parallel hardware proving our efficiency claim. In Section 2 we review the Beltrami framework. In Section 3 we extend the constrained optimization approach demonstrated in [41] to regularize color images by the Polyakov action (or more specifically, the Nambu-Goto functional). In Section 4 we display results of using our method for deblurring color images. Section 5 concludes the paper.

2. The Beltrami Framework. The Beltrami framework for non-linear diffusion in computer vision [24, 37, 38, 47] provides a coherent and axiomatic description of multi-channel signal denoising, with strong ties to differential and Riemannian geometry of surfaces. We refer the reader to [25, 15] for an in-depth discussion of the basic notions of Riemannian geometry used in the following discussion of the Beltrami framework.

In the Beltrami framework we consider images to be manifolds embedded into a higher dimensional embedding space, usually a feature-coordinates space, endowed with some metric induced by that space.

We denote by (Σ, g) the image manifold and its metric for spatial coordinates. We assume global coordinates, for example (σ^1, σ^2) for the case of 2D images. We denote by M the spatial-feature manifold, embedded in $\mathbb{R}^{d+d'}$, where d is the number of image channels, and d' is the number of spatial coordinates. As in the case of Dirichlet energy [16], we define regularization on a map $X : \Sigma \rightarrow M$, where 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).

Next, we choose a Riemannian metric on this surface. Its components are denoted by g_{ij} . The canonical choice of coordinates σ^1 and σ^2 in image processing uses Cartesian coordinates aligned with the x and y directions. 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, we can measure the manifold area, using the metrics of the spatial and spatial-feature spaces. The functional $S[X]$ characterizes the mapping $X : \Sigma \rightarrow M$, and is defined to be

$$S[X, g_{ij}, h_{ab}] = \int d^m \sigma \sqrt{g} \|dX\|_{g,h}^2, \quad (2.1)$$

where m is the dimension of Σ , g is the determinant of the image metric, and the

range of indices is $i, j = 1, 2, \dots, \dim(\Sigma)$ and $a, b = 1, 2, \dots, \dim(M)$. The integrand $\|dX\|_{g,h}^2$ is given by $\|dX\|_{g,h}^2 = (\partial_{x_i} I^a) g^{ij} (\partial_{x_j} I^b) h_{ab}$. 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 and hence $g_{ij} = \delta_{ij}$) are

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

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,

$$0 = -\frac{1}{\sqrt{g}} h^{ab} \frac{\delta S}{\delta I^b} = \underbrace{\frac{1}{\sqrt{g}} \operatorname{div}(D \nabla I^a)}_{\Delta_g I^a}, \quad (2.2)$$

where the matrix $D = \sqrt{g} G^{-1}$. See [37] 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 Δ_g .

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

$$I_t^a = -\frac{1}{\sqrt{g}} \frac{\delta S}{\delta I^a} = \Delta_g I^a, \quad (2.3)$$

with reflective boundary conditions and a smooth initial solution $I^a|_{t=0} = I_0^a$. Evolution according to these equations gives us the Beltrami scale-space, and allows image filtering and denoising. For Euclidean embedding, the functional in Eq. 2.1 reduces to

$$S(X) = \int \sqrt{g} d\sigma^1 d\sigma^2, \quad g = \det(G).$$

The area element for the case of 2D images can also be written as following [24]

$$g = 1 + \beta^2 \sum_{a=1}^3 \|\nabla I^a\|^2 + \frac{\beta^4}{2} \sum_{a,b=1}^3 \|\nabla I^a \times \nabla I^b\|^2. \quad (2.4)$$

This form suggests a novel fixed-point method for minimizing the regularization term and associated inverse problems, as will be shown in Section 3.

A special note should be made on cross product term $\sum_{a,b=1}^3 \|\nabla I^a \times \nabla I^b\|^2$ in the minimization. Its role was explored in [24], see also [23]. This term gives a high cost to misalignments of the various gradient channels. In particular, it penalizes deviations from the Lambertian model of image formation, and is quite helpful in preventing compression artifacts. Taking large values of β 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.

If we forgoe the differential geometric interpretation and focus on the image-processing outlook, a more general form of the functional can be given – of the form

$$\int \sqrt{\beta_1 + \beta_2 \sum_{a=1}^3 \|\nabla I^a\|^2 + \beta_3 \sum_{a,b=1}^3 \|\nabla I^a \times \nabla I^b\|^2} d\sigma^2, \quad (2.5)$$

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 allows us to tweak the smoothness and alignment terms in applications of color image restoration. Hence, we use it in the results shown hereafter.

The geometric functional S can be used as a regularization term for inverse problems in 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

$$\Psi = \frac{\alpha}{2} \sum_{a=1}^3 \|KI^a - I_0^a\|^2 + S(X),$$

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 α controls the smoothness of the solution. This functional have been used for image denoising [37, 3] and blind deconvolution [22], among other uses. Its relation to active contours models was explored, along with an extension for multiscale segmentation in [9]. We introduce an approach for finding an optimum for the functional Ψ using the augmented Lagrangian method.

3. An augmented Lagrangian approach for Beltrami regularization.

Several attempts have been made of optimizing total variation functionals using auxiliary variables. We refer the reader to [12, 10, 11, 26, 46, 48, 41] and references therein. In most of these techniques, the functional is treated as a optimization problem (with certain properties being exploited), instead of using the Euler-Lagrange equations to form the traditional minimizing flow [34].

Optimization methods for TV regularization achieved great accuracy and efficiency, and are considered to be among state-of-the-art methods for TV restoration. The special representation of the problem in terms of the auxiliary variables often leads to much simpler substeps in the algorithm, often allowing parallelization of the algorithm, and highly efficient implementation, for example on graphics processing units (GPUs).

Specifically, in [41], total-variation regularization is obtained by decoupling the optimization problem

$$\min_u \int |\nabla u| + \frac{\alpha}{2} \|Ku - f\|^2 \quad (3.1)$$

into a constrained optimization problem

$$\min_{u, \mathbf{q}} \int |\mathbf{q}| + \frac{\alpha}{2} \|Ku - f\|^2 \quad s.t. \quad \mathbf{q} = \nabla u, \quad (3.2)$$

where \mathbf{q} is a field that approximates to the gradient of u . This constraint is then enforced using an augmented Lagrangian penalty function of the form $\rho_{\mu,r}(u, \mathbf{q}) = \boldsymbol{\mu}^T(\nabla u - \mathbf{q}) + \frac{r}{2}(\|\nabla u - \mathbf{q}\|^2)$. The penalty is used to enforce the constraint $\mathbf{q} = \nabla u$, without making the problem severely ill-conditioned, resulting in the optimization problem

$$\max_{\mu} \min_{u, \mathbf{q}} \int \left\{ |\mathbf{q}| + \frac{\alpha}{2} \|Ku - f\|^2 + \boldsymbol{\mu}^T(\mathbf{q} - \nabla u) + \frac{r}{2} \|\mathbf{q} - \nabla u\|^2 \right\}, \quad (3.3)$$

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 [37, 22, 40, 3, 30, 14]. We treat the case of color 2D images, which is the classical setting for the Beltrami framework [37].

Specifically, we replace the gradient norm penalty used in total variation regularization by the area measure of the map, as described in Equation 2.1. This is done by replacing the TV term $|\mathbf{q}|$ in Equation 3.2 by the term

$$\int \underbrace{\sqrt{1 + \beta^2 \sum_{i \in \{R, G, B\}} \|\mathbf{q}_i\|^2 + \frac{\beta^4}{2} \sum_{i \in \{R, G, B\}} \sum_{j \neq i} \|\mathbf{q}_i \times \mathbf{q}_j\|^2 d\sigma^1 d\sigma^2}}_{\psi_{BEL}(\mathbf{q})}, \quad (3.4)$$

where β is the spatial-intensity aspect ratio. $\{\mathbf{q}_i\}_{i \in \{R, G, B\}}$ denote the components of the field \mathbf{q} that express the gradient of each of the image channels. The rest of the functional can then be trivially extended to multi-channel image regularization, obtaining the following functional

$$\begin{aligned} \mathcal{L}_{BEL}(u, \mathbf{q}, \mu^k) = & \\ & \int \psi_{BEL}(\mathbf{q}) + \sum_i \boldsymbol{\mu}_i^T(\mathbf{q}_i - \nabla u_i) + \frac{\alpha}{2} \|Ku - f\|^2 + \frac{r}{2} \sum_i \|\mathbf{q}_i - \nabla u_i\|^2 d\sigma^1 d\sigma^2, \end{aligned}$$

which corresponds to Beltrami regularization. Since the image is multi-channel, expressions optimizing u and $\boldsymbol{\mu}$ are replaced by their per-channel equivalents u_i and $\boldsymbol{\mu}_i$, for $i \in \{R, G, B\}$.

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

The variables u_i , and $\boldsymbol{\mu}_i$ are optimized as suggested in [41]. The Euler-Lagrange equation for u_i is given by

$$\operatorname{div} \boldsymbol{\mu}_i + \alpha K^* * (K * u - f) + \frac{r}{2} - 2 \operatorname{div}(\mathbf{q}_i - \nabla u_i) = 0 ,$$

where K^* is the adjoint operator of K – for the deblurring case, its kernel is given by $\bar{k}(x, y) = k(-x, -y)$. At each inner iteration k , $\{u_i\}_{i \in \{R, G, B\}}$ is updated using the optimality condition with respect to it [41],

$$\alpha K^* (Ku_i - f) + \operatorname{div} \boldsymbol{\mu}_i^k + r \operatorname{div} \mathbf{q}_i^k - r \Delta u_i = 0.$$

Algorithm 1 Augmented Lagrangian optimization of the Beltrami framework

```

1:  $\mu^0 \leftarrow 0$ 
2: for k=0,1,... do
3:   update  $\{u_i\}^k, \{\mathbf{q}_i\}^k$ :

$$(\{u_i\}^k, \{\mathbf{q}_i\}^k) \leftarrow \underset{\{u_i\}, \{\mathbf{q}_i\}}{\operatorname{argmin}} \mathcal{L}_{BEL}(\{u_i\}, \{\mathbf{q}_i\}, \{\mu_i^k\}) \quad (3.5)$$

   according to Equation 3.6 and Subsection 3.1.
4:   update the Lagrange multipliers  $\{\mu_i\}$  according to Equation 3.7.
5: end for

```

One way to perform the update is using the Fourier domain, leading to the update rule

$$u_i^k = \mathcal{F}^{-1} \left\{ \frac{\alpha \mathcal{F}\{K^*\} \mathcal{F}\{f_i\} - \mathcal{F}\{D_x^-\}((\mu_i^1)^k + r(p_i)^k) - \mathcal{F}\{D_y^-\}((\mu_i^2)^k + r(q_i)^k)}{\alpha \mathcal{F}\{K^*\} \mathcal{F}\{K\} - r \mathcal{F}\{\Delta\}} \right\}, \quad (3.6)$$

where D_x^- , D_y^- , Δ 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. Another possibility is to use a Gauss-Seidel solver in order to solve the optimality condition. Gauss-Seidel iterations have the advantage of easily incorporating non-repetitive boundary conditions such as Neumann BC.

We explicitly write $\mathbf{q}_i = (p_i, q_i)$, $i \in \{R, G, B\}$, for the components of \mathbf{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 an updated auxiliary field, iteratively, its effect is suited to the nonlinear nature of the Beltrami flow. An update rule for the dual field \mathbf{q}_i of each channel is described in Subsection 3.1.

The Lagrange multipliers μ_i are updated so as to approximate the optimal Lagrange multipliers. Unlike the case of TV regularization, the cost function is sufficiently smooth for the relevant discussion in ([7], Section 2.),

$$(\mu_i)^k = (\mu_i)^{k-1} + r ((\mathbf{q}_i)^k - (\nabla u_i)^k). \quad (3.7)$$

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 $\mathcal{L}_{BEL}(u, \mathbf{q}, \mu)$.

3.1. Updating the dual field \mathbf{q} . When optimizing \mathbf{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 both on CPU and on GPU, optimization over \mathbf{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, as we will demonstrate.

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

$$\sqrt{1 + \beta^2 \sum (p_i^2 + q_i^2) + \frac{\beta^4}{2} \sum_i \sum_{j \neq i} (p_i q_j - q_i p_j)^2}$$

$$+ \frac{r}{2} \sum_i \| \mathbf{q}_i - (\nabla u_i) \|^2 + \sum_{i \in \{R, G, B\}} (\boldsymbol{\mu}_i^k)^T (\mathbf{q}_i - \nabla u_i),$$

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 \mathbf{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

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

and similarly for q_i . Next, we freeze the denominator and solve for each component of \mathbf{q} , thereby obtaining a fixed-point iteration of the form

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

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

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

and l denotes the IRLS iteration number. These iterations are repeated for all 6 elements of \mathbf{q} for several inner iterations, at each pixel.

4. Results. We now demonstrate the minimization of our geometric functional using the augmented Lagrangian method. We first demonstrate our algorithm for denoising, smoothing and deblurring. We then demonstrate the efficiency of denoising using this algorithm on a GPU.

Denoising

In Figure 4.1, we show results for smoothing an image using various values of α , with K equal to the identity operator, and $\beta = \sqrt{4000} \approx 63.25$. This approximates samples along the Beltrami scale-space. 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 \mathbf{q} were limited to 2 inner and 2 outer (IRLS) iterations for each cycle. The number of outer iterations, updating $\boldsymbol{\mu}$, in Figure 4.1 was 150, although fewer iterations suffice. The residual plot is shown in Figure 4.2, 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) [14] method shows that the augmented Lagrangian method

results in faster convergence, as can be seen in Figure 4.3. In this experiment, α was set for optimal results for both the augmented Lagrangian and the LOD method. 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 4.3. Experiments comparing our method to explicit scheme time evolution with backward-forward discretization led to similar conclusions.

Table 4.1 demonstrates the CPU-time required for several images (shown in Figure 4.4) for our algorithm, compared to Beltrami filtering with operator splitting techniques (in this case we used LOD splitting). The time step Δt 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 of two compared to LOD splitting, which is one of the fastest methods for Beltrami regularization. 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.

Deblurring

The result of deblurring a color image using the Beltrami framework are shown in Figures 4.5–4.8. In Figures 4.5–4.7, The color image is blurred using a disc kernel of radius 5 pixels. In Figure 4.8, the point spread function (PSF) of the filter is given by

$$k(x, y) = \frac{1}{1 + x^2 + y^2}, \quad x, y = -7, \dots, 7. \quad (4.1)$$

This filter is often used for comparing deblurring operators, see for example [19].

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 [43, 46]. 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 4.5–4.8, we have chosen to use the more general form of the function, given in Equation 2.5. We set β_1 set to a very small positive constant. This choice reflects a preference of a flow without a linear diffusion part. Furthermore, we set $\beta_2 = \frac{1}{\beta}$, $\beta_3 = \frac{1}{20}$. 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

Image	CPU time, AL	CPU time, splitting
Astro	1.77s	3.5s
Fruits	2.97s	7.36s
Lion	21.23s	59.66s
Monarch	3.63s	7.71s

Table 4.1: 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.

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 4.5, with magnified details shown in Figure 4.6. We note the same discrepancy between PSNR reading and visual results in color image processing has already been noted by Goldluecke et. al [20]. We simply iterate this word of caution here, and refer the reader to the images themselves.

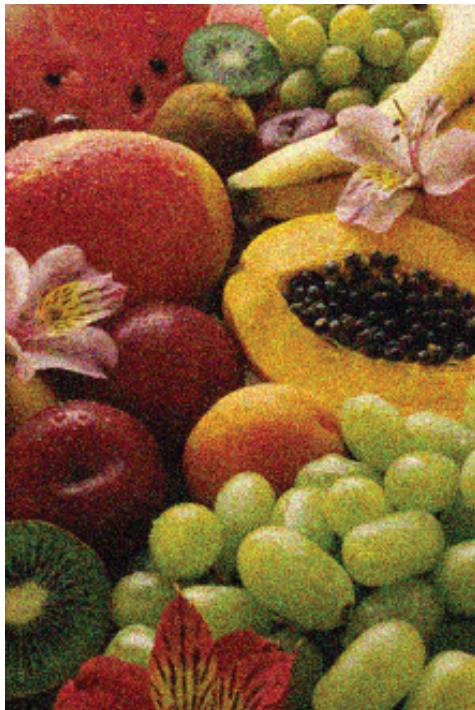
l_1 Denoising

An addition improvement in inverse imaging problems using augmented Lagrangian method can be seen in robust, non-quadratic fitting terms. In [45], an additional auxiliary variable was used to allow non-quadratic data terms such as l_1 fitting terms. In Figure 4.9 we demonstrate the addition of an l_1 fitting term to our functional, on a color image corrupted with 25% outlier noise (outlier pixel are given arbitrary uniformly sampled RGB values). The implementation using the same auxiliary variables and optimization done as in [45]. We refer the reader to [45] for more details. The results shows the suitability of our approach for outlier removal in multi-channel images.

GPU Implementation

In order to demonstrate the applicability of our method for parallel implementation, we have implemented it on graphics processing units using the CUDA toolkit. In Figure 4.10 we demonstrate the runtimes for image processing with the same parameters in Figure 4.1, and $\alpha = 0.4$, using an NVIDIA GTX680 card. The number of iterations was set to allow more than three and a half orders of magnitude drop in the residual norm – 50 inner iterations. As can be seen, the computational complexity scales linearly with image size, and for small images, results are obtained in realtime speeds.

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 showed numerical examples demonstrating its efficiency and accuracy compared to existing techniques for variational regularization, its effectiveness in image deblurring, and suitability for parallel implementations. In future work we intend to extend the use of our framework for other types of applications involving of multidimensional, multi-channel images, as well as extend it to other non-Cartesian sampled data.



Fruits, noisy

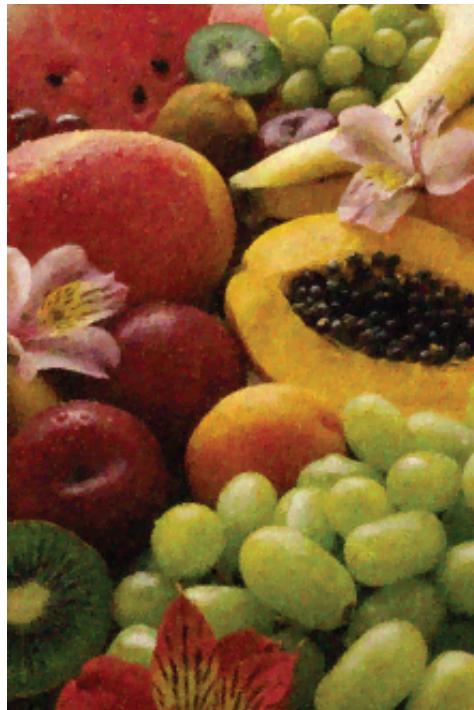
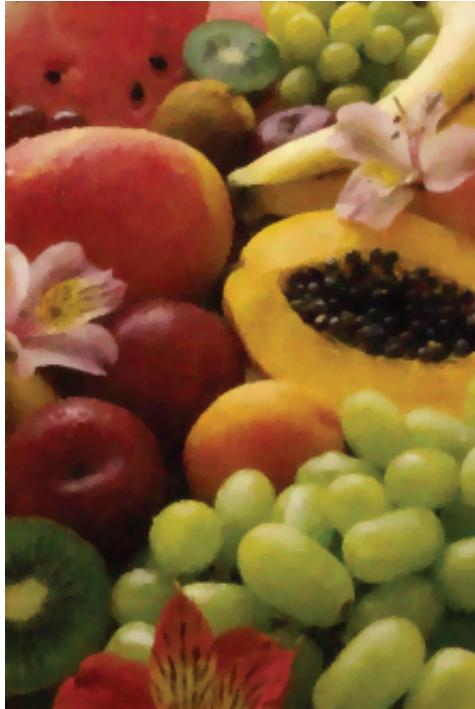
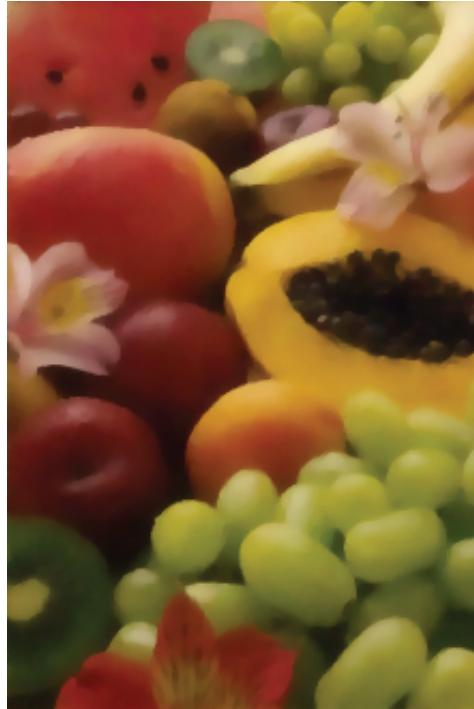
Fruits, denoised, $\alpha = 1.00$ Fruits, denoised, $\alpha = 0.40$ Fruits, denoised, $\alpha = 0.15$

Fig. 4.1: Smoothing, under various α values, of the Fruits image, with added Gaussian noise with $\sigma = 20$ intensity levels per channel.

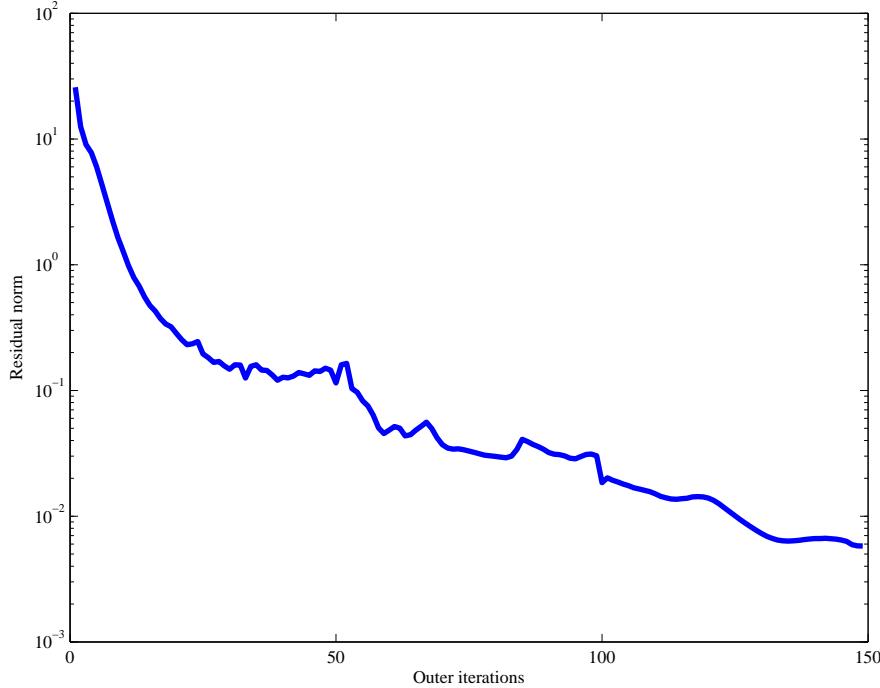
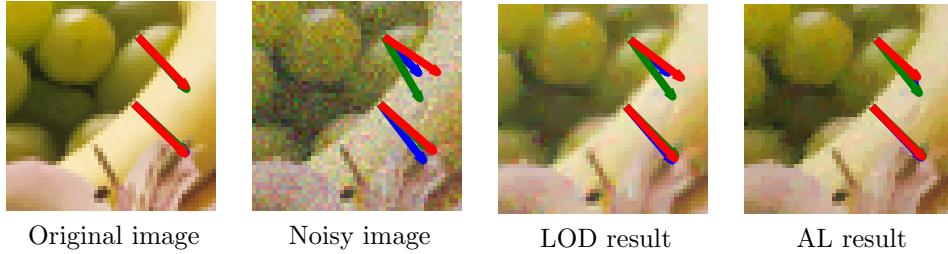


Fig. 4.2: The residual norm, $\|u^{k+1} - u^k\|_2^2$, as a function of iterations, for the augmented Lagrangian method for Beltrami regularization in Figure 4.1 ($\alpha = 0.15$).

REFERENCES

- [1] Bartomeu Coll Antoni Buades and Jean-Michel Morel, *A review of image denoising algorithms, with a new one*, SIAM Interdisciplinary Journal **4** (2005), 490–530.
- [2] Volker Aurich and Jörg Weule, *Non-linear Gaussian filters performing edge preserving diffusion*, Mustererkennung 1995, 17. DAGM-Symposium (London, UK), Springer Verlag, 1995, pp. 538–545.
- [3] Leah Bar, Alexndar Brook, Nir Sochen, and Nahum Kiryati, *Deblurring of color images corrupted by impulsive noise*, IEEE Trans. Image Process. **16** (2007), no. 4, 1101–1111.
- [4] Danny Barash, *A fundamental relationship between bilateral filtering, adaptive smoothing and the nonlinear diffusion equation*, IEEE Trans. Pattern Anal. Mach. Intell. **24** (2002), no. 6, 844–847.
- [5] Thomas Batard and Nir A. Sochen, *Polyakov action on $(?, g)$ -equivariant functions application to color image regularization*, SSVM, 2011, pp. 483–494.
- [6] Rami Ben-Ari and Nir Sochen, *A geometric framework and a new criterion in optical flow modeling*, Journal of Mathematics in Imaging and Vision **33** (2009), no. 2, 178–194.
- [7] Dimitri P. Bertsekas, *Constrained optimization and Lagrange multiplier methods*, 1st ed., Athena Scientific, 1982.
- [8] Xavier Bresson and Tony Chan, *Fast dual minimization of the vectorial total variation norm and applications to color image processing*, CAM-Report 07-25, UCLA, 2007.
- [9] Xavier Bresson, Pierre Vandergheynst, and Jean-Philippe Thiran, *Multiscale active contours*, International Journal of Computer Vision **70** (2006), no. 3, 197–211.
- [10] Jamylle Laurice Carter, *Dual methods for total variation-based image restoration*, CAM-Report 02-13, UCLA, April 2002.
- [11] Antonin Chambolle, *An algorithm for total variation minimization and applications*, J. Math. Imaging Vis. **20** (2004), no. 1-2, 89–97.
- [12] Tony F. Chan, Gene H. Golub, and Pep Mulet, *A nonlinear primal-dual method for total variation-based image restoration*, SIAM J. Sci. Comput. **20** (1999), 1964–1977.



Original image Noisy image LOD result AL result

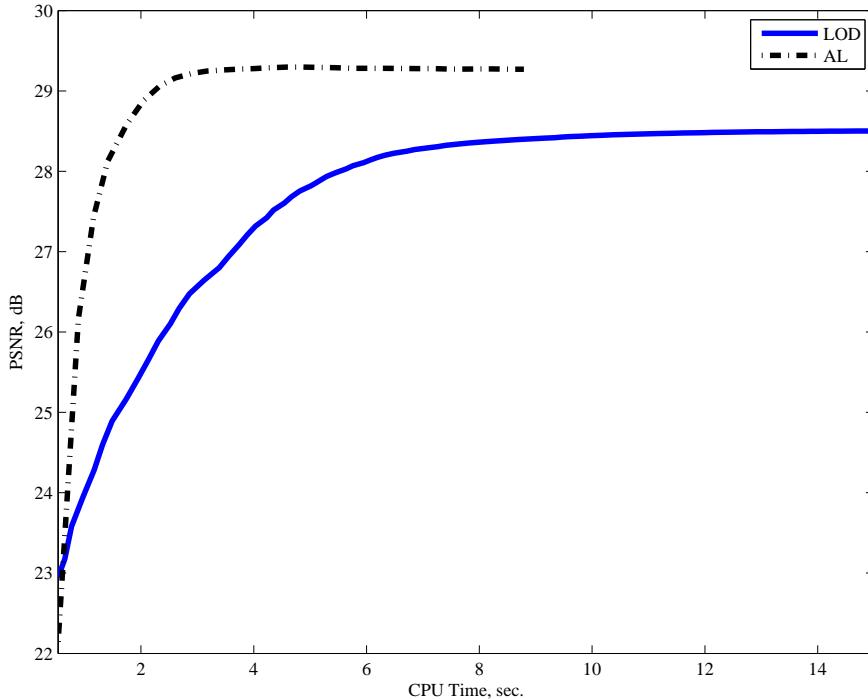


Fig. 4.3: A comparison of the results for LOD, and the augmented Lagrangian method, as well as the PSNR (for the whole image, compared to the uncorrupted image) plotted as a function of CPU time. The arrows in the images demonstrate gradient directions at each channel. The plot demonstrates the fast convergence of the augmented Lagrangian method, as well as a more accurate discretization.

- [13] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian, *Image restoration by sparse 3D transform-domain collaborative filtering*, Proc. SPIE (Jaakko T. Astola, Karen O. Egiazarian, and Edward R. Dougherty, eds.), vol. 6812, 2008.
- [14] Lorina Dascal, Guy Rosman, Xue-Cheng Tai, and Ron Kimmel, *On semi-implicit splitting schemes for the Beltrami color flow*, Scale Space and Variational Methods in Computer Vision (Berlin, Heidelberg), Springer Verlag, 2009, pp. 259–270.
- [15] Manfredo Perdigao do Carmo, *Riemannian geometry*, Birkhäuser Verlag, Boston, MA, 1992.
- [16] James Jr. Eells and J. H. Sampson, *Harmonic mappings of Riemannian manifolds*, American J. of Math **86** (1964), no. 1, 106–160.
- [17] Michael Elad, *On the bilateral filter and ways to improve it*, IEEE Trans. Image Process. **11**



Fig. 4.4: 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.

- (2002), no. 10, 1141–1151.
- [18] Micha Feigin, Nir A. Sochen, Baba C. Vemuri, and Baba C. Vemuri, *Anisotropic alpha-kernels and associated flows.*, 2010, pp. 904–925.
 - [19] Mário A. T. Figueiredo and Robert D. Nowak, *A bound optimization approach to wavelet-based image deconvolution*, ICIP (2), 2005, pp. 782–785.
 - [20] Bastian Goldluecke and Daniel Cremers, *An approach to vectorial total variation based on geometric measure theory*, Computer Vision and Pattern Recognition, 2010.
 - [21] Magnus R. Hestenes, *Multipliers and gradient methods*, Journal of Optimization Theory and Applications **4** (1969), 303–320.
 - [22] Ran Kaftry, Nir Sochen, and Yehoshua Y. Zeevi, *Variational blind deconvolution of multi-channel images*, International Journal of Imaging Science and Technology **15** (2005), no. 1, 56–63.
 - [23] Ron Kimmel, *Numerical geometry of images: Theory, algorithms, and applications*, Springer Verlag, 2003.
 - [24] Ron Kimmel, Ravi Malladi, and Nir Sochen, *Images as embedding maps and minimal surfaces: Movies, color, texture, and volumetric medical images*, International Journal of Computer Vision **39** (2000), no. 2, 111–129.
 - [25] Erwin Kreyszig, *Introduction to differential geometry and Riemannian geometry*, University of Toronto Press, 1969.
 - [26] Stanley Osher, Martin Burger, Donald Goldfarb, Jinjun Xu, and Wotao Yin, *An iterative regularization method for total variation-based image restoration*, Simul **4** (2005), 460–489.
 - [27] Sylvain Paris and Frédo Durand, *A fast approximation of the bilateral filter using a signal processing approach*, International Journal of Computer Vision **81** (2009), no. 1, 24–52.
 - [28] A. M. Polyakov, *Quantum geometry of bosonic strings*, Physics Letters **103 B** (1981), no. 3, 207–210.
 - [29] Michael J.D. Powell, *Optimization*, ch. A method for nonlinear constraints in minimization problems, pp. 283–298, Academic Press, 1969.
 - [30] Guy Rosman, Lorina Dascal, Ron Kimmel, and Avram Sidi, *Efficient beltrami image filtering via vector extrapolation methods*, SIAM J. Imag. Sci. (2008), no. 3, 858–878.
 - [31] Guy Rosman, Xue-Cheng Tai, Lorina Dascal, and Ron Kimmel, *Polyakov action for efficient color image processing*, ECCV workshop on color and reflectance in comp. vision, 2010.
 - [32] ———, *Polyakov action for efficient color image processing*, ICNAAM, AIP, vol. 1281, 2010, pp. 1018–1021.
 - [33] Anastasios Roussos and Petros Maragos, *Tensor-based image diffusions derived from generalizations of the total variation and beltrami functionals*, ICIP, 2010, pp. 4141–4144.
 - [34] Leonid I. Rudin, Stanley Osher, and Emad Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D Letters **60** (1992), 259–268.
 - [35] Stephen M. Smith and J.M. Brady, *SUSAN – A new approach to low level image processing*, International Journal of Computer Vision **23** (1997), 45–78.
 - [36] Nir Sochen, Ron Kimmel, and Alfred M. Bruckstein, *Diffusions and confusions in signal and image processing*, Journal of Mathematics in Imaging and Vision **14** (2001), no. 3, 195–209.
 - [37] Nir Sochen, Ron Kimmel, and Ravi Maladi, *From high energy physics to low level vision*, Proceedings of the First International Conference on Scale Space Theory in Computer

- Vision, Lecture Notes on Computer Science, vol. 1252, July 1997, pp. 237–247.
- [38] ———, *A general framework for low level vision*, IEEE Trans. Image Process. **7** (1998), no. 3, 310–318.
 - [39] Nir A. Sochen and Leah Bar, *The beltrami-mumford-shah functional.*, SSVM, 2011, pp. 183–193.
 - [40] Alon Spira, Ron Kimmel, and Nir A. Sochen, *A short-time Beltrami kernel for smoothing images and manifolds.*, IEEE Trans. Image Process. **16** (2007), no. 6, 1628–1636.
 - [41] Xue-Cheng Tai and Chunlin Wu, *Augmented Lagrangian method, dual methods and split Bregman iteration for ROF model*, Scale Space and Variational Methods in Computer Vision, 2009, pp. 502–513.
 - [42] Carlo Tomasi and Roberto Manduchi, *Bilateral filtering for gray and color images*, International Conference on Computer Vision (1998), 836–846.
 - [43] Yilun Wang, Junfeng Yang, Wotao Yin, and Yin Zhang, *A new alternating minimization algorithm for total variation image reconstruction*, SIAM J. Imag. Sci. **1** (2008), no. 3, 248–272.
 - [44] A. Wetzler, G. Rosman, and R. Kimmel, *Patch-space beltrami denoising of 3D point clouds*, IEEE Israeli chapter conference, 2012.
 - [45] Chunlin Wu, Juyong Zhang, and Xue-Cheng Tai, *Augmented Lagrangian method for total variation restoration with non-quadratic fidelity*, CAM Report 09-82, UCLA, December 2009.
 - [46] J.F. Yang, W.T. Yin, Y. Zhang, and Y.L. Wang, *A fast algorithm for edge-preserving variational multichannel image restoration*, SIAM J. Imag. Sci. **2** (2009), no. 2, 569–592.
 - [47] Anthony J. Yezzi, *Modified curvature motion for image smoothing and enhancement*, IEEE Trans. Image Process. **7** (1998), no. 3, 345–352.
 - [48] X. Zhang, M. Burger, X. Bresson, and S. Osher, *Bregmanized nonlocal regularization for deconvolution and sparse reconstruction*, CAM-Report 09-03, UCLA, 2009.
 - [49] Dominique Zosso, Xavier Bresson, and Jean-Philippe Thiran, *Geodesic active fields - a geometric framework for image registration.*, IEEE Transactions on Image Processing **20** (2011), no. 5, 1300–1312.

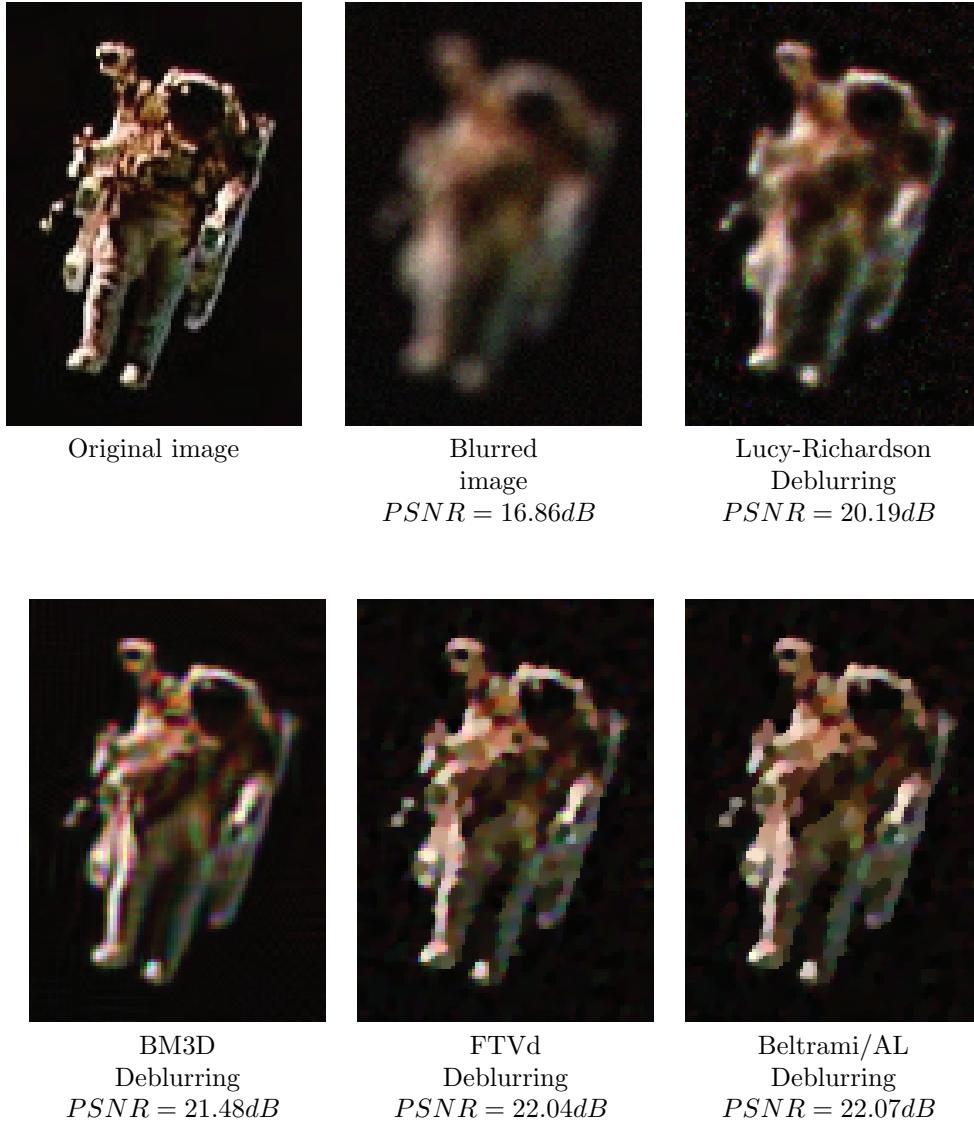


Fig. 4.5: Deblurring results obtained on an astronaut image with a disc blur filter of radius 5 and an added Gaussian noise of $\sigma = 5$. Left to right, top to bottom: (a) The original image. (b) The blurred image. (c) Deblurring using the Lucy-Richardson algorithm, as implemented by Matlab (d) BM3D-based deblurring. (e) Deblurring using the FTVd method. (f) Deblurring using the Beltrami / augmented Lagrangian algorithm.

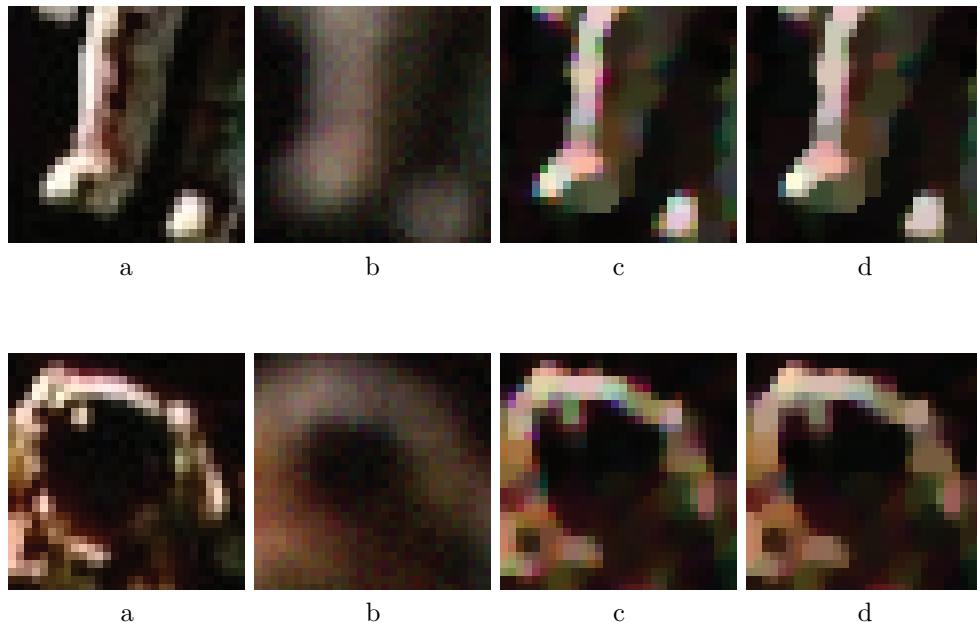


Fig. 4.6: Each row represents two regions zoomed-in from Figure 4.5. (a) The original image. (b) The blurred and noisy image. (c) TV-restoration results. (d) Beltrami / augmented Lagrangian restoration results.

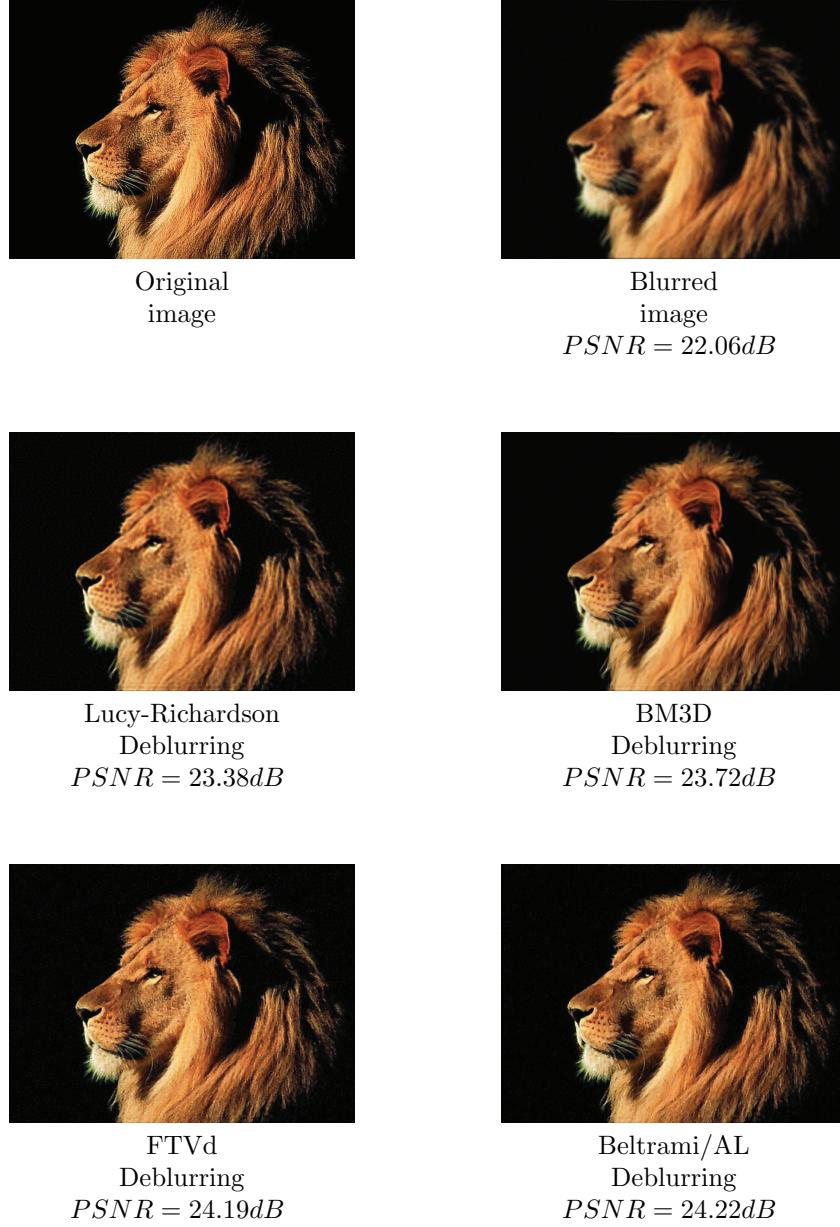


Fig. 4.7: Deblurring results obtained on the lion image blurred with a disc blur filter of radius 5 and an added Gaussian noise of $\sigma = 5$. Left to right, top to bottom: (a) The original image. (b) The blurred image. (c) Deblurring using the Lucy-Richardson algorithm, as implemented by Matlab (d) BM3D-based deblurring. (e) Deblurring using the FTVd algorithm. (f) Deblurring using the Beltrami / augmented Lagrangian algorithm.

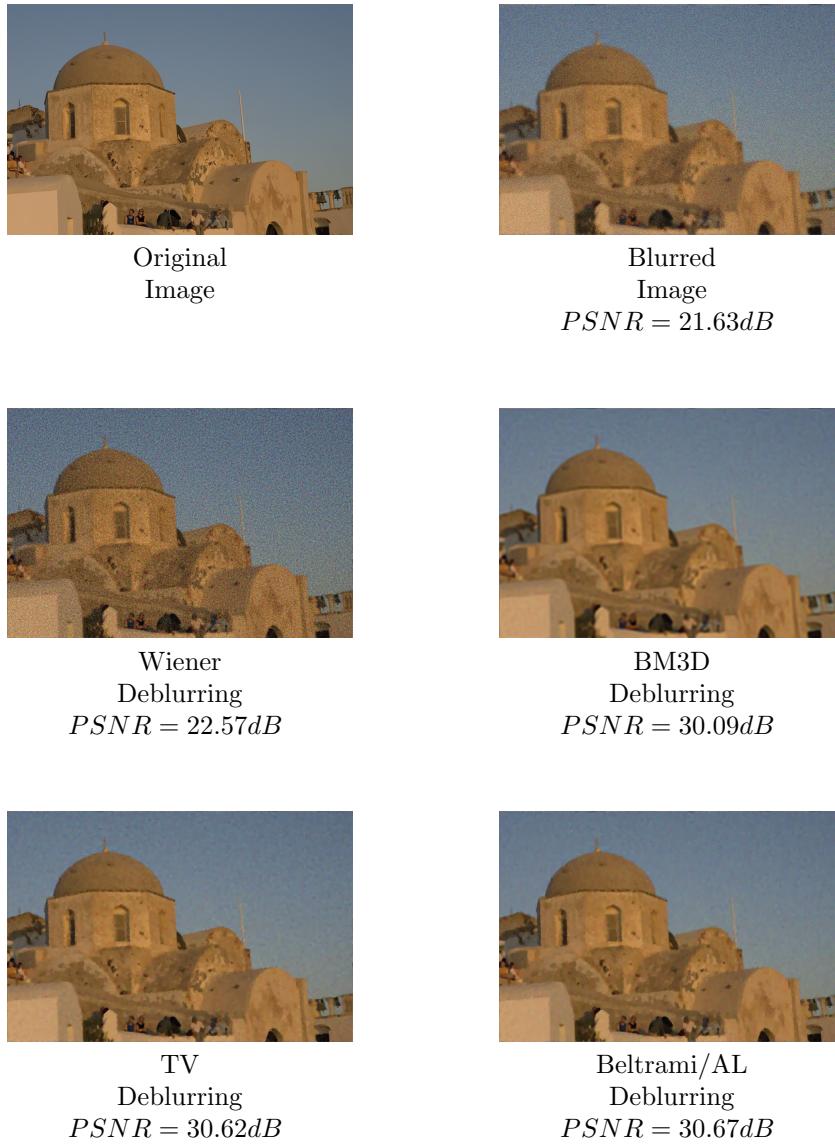


Fig. 4.8: Deblurring results obtained on the Greek dome image (courtesy of [27]) blurred with a filter of a PSF as given in Equation 4.1 and an added Gaussian noise of $\sigma = 8$. Left to right, top to bottom: (a) The original image. (b) The blurred image. (c) Deblurring using a Wiener filter, as implemented by Matlab (d) BM3D-based deblurring. (e) Deblurring using the FTVd algorithm. (f) Deblurring using the Beltrami / augmented Lagrangian algorithm.

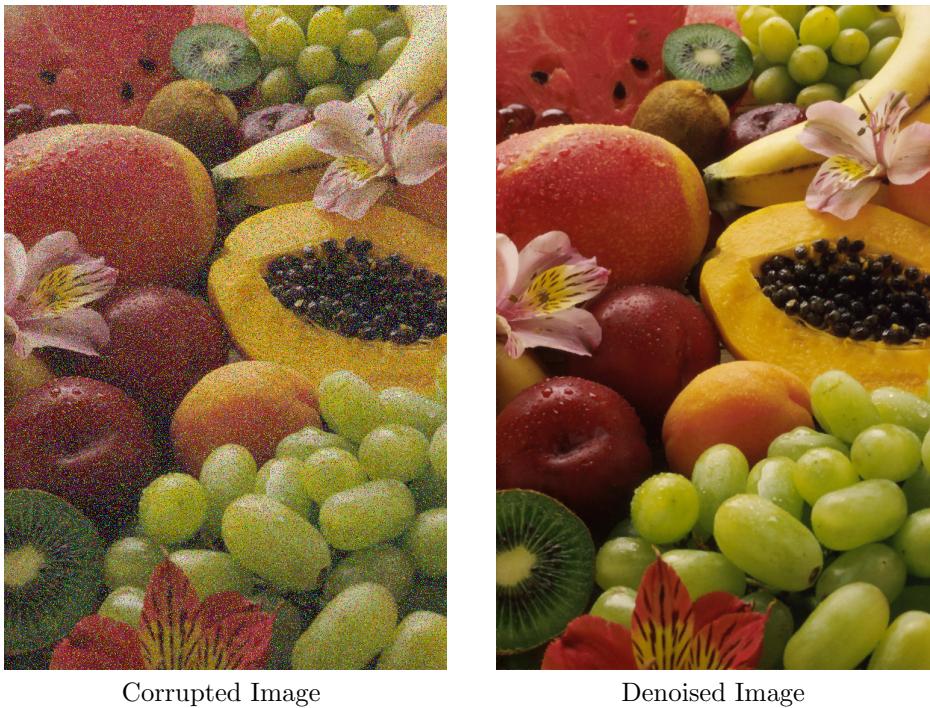


Fig. 4.9: Outlier noise removal results on the fruits image, with 25% outliers. Left to right: The corrupted image, the denoised image.

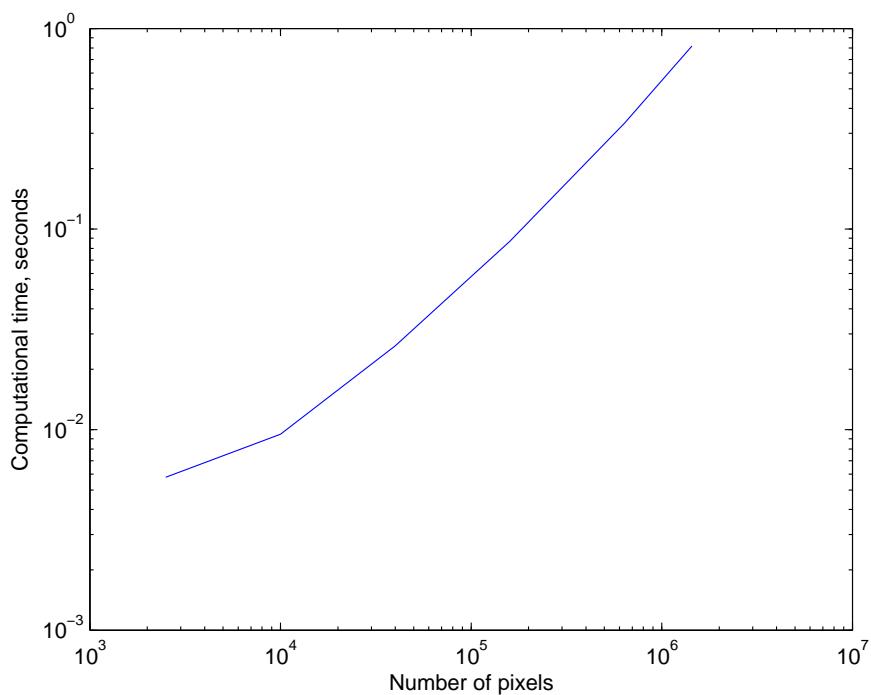


Fig. 4.10: Runtimes as a function of image size, using an NVIDIA GTX680 GPU.