Bregman Approach to Single Image De-Raining

László Szirmay-Kalos and Márton Tóth

Budapest University of Technology and Economics, Dept. of Control Engineering and Information Technology (e-mail: szirmay@iit.bme.hu)

Abstract

Surveillance cameras are expected to work also in bad visibility conditions, which requires algorithmic solutions to improve the captured image and to eliminate image degradation caused by these weather conditions. Algorithms for such tasks belong to the field of computational photography and have been successful in eliminating haze, fog, motion blur, etc. This paper presents a simple algorithm to suppress rain or snow from single images. The algorithm uses energy minimization, and we propose a novel data term and a Bregman distance based regularization term reflecting the particular properties of precipitation.

1. Introduction

Rain or snow may degrade visibility conditions, which makes it difficult for automatic object detection algorithms to execute their tasks. Therefore, a pre-processing step is needed that eliminates these degradations and reconstructs the image of the scene which would be similar to the one made under ideal visibility conditions. Such approaches are called de-raining or de-snowing methods and are expected to remove rain or snow streaks from the image. As precipitation is a temporal phenomenon, de-raining or de-snowing can work in the temporal domain exploiting the correlation of video frames [GN04, KSK15]. However, if the scene is not static or the camera itself is moving, the temporal correlation of the colors of the same pixel disappears. If such correlation is still forced on the video sequence, motion blur artifacts may show up. One approach to attack this problem is to process single images based on spatial coherence only. Single image de-raining is difficult and less reliable because of the lacking temporal information [LAR*19].

2. Previous Work

Hase et al. [HMY99] used temporal median filter to eliminate snow fall. The main problem of median filtering is that rain or snow is not a point-like phenomenon, and they can occupy a larger portion of the image for multiple frames. Thus, a median filter would be able to eliminate precipitation if it considered a larger neighborhood, which would result in blurring artifacts. The size of the spatial neighborhood can be reduced by recognizing that rain droplets produce vertical streaks.

To reduce motion blur at non-rainy parts, Garg and Nayar [GN04] concluded that snowy or rainy pixels are brighter than the occluded background, and thus detected rainy pixels by checking if their intensity is higher than in adjacent frames. Zhang et al. [ZLQ*06] increased the robustness by introducing photometric constraints: a pixel color should be dominated by the background color, and the brightness changes of rainy pixels should be similar through an entire sequence. Motion blur artifacts can be reduced by optical flow estimation [MX11], but its implementation does not allow the processing of high resolution images in real-time.

Single image de-raining is an ill-posed decomposition problem. Non-perfect decomposition may only weaken streaks or over-smooth details. The missing information may be supplied by a prior. Prior-based techniques include sparse coding [KLF12, HKWL14, LXJ15], low-rank representation [CH13] and Gaussian mixture models [LTG*16]. These approaches work well as long as the prior assumption is valid, but fail in complex scenarios. Additionally, as they lead to non-convex optimization, the implementation is complicated and running times are long.

Deep learning approaches learn the mapping from the rainy to the de-rained image by training on rainy and ground truth pairs [ZP18]. Unfortunately, it is difficult to obtain such image pairs.

Non-linear or non-local filtering [KLSK13] or energy minimalization methods can also be used to de-raining, where the objective is to suppress vertical and elongated features [JHZ*19]. Oblique streaks can be attacked by rotating the image or by the application of directional total variation [WHZ*19].

This paper investigates single-image de-raining with energy minimization. The novelties are the data fidelity and smoothness terms taking the properties of precipitation into account, and the application of Bregman distance in the energy function.

3. Proposed de-raining algorithm

Snow crystals scatter light diffusely and their albedo is close to one on all wavelengths, i.e. they look brighter than other objects behind them. Rain droplets reflect and also refract light. As rain droplets gather light from a larger solid angle than they are visible from the camera, they also look brighter than occluded objects. Rainy or snowy pixels are also less saturated than pixels of no precipitation.

Thus, the added effect of rain or snow is positive on all wavelengths and natural, which enables us to define a criterion for potentially useful pixels to filter the target pixel.

Let us consider the input image and denote the color of a pixel at position (i,j) by $\mathbf{I}_{i,j} = (\mathbf{I}_{i,j,r}, \mathbf{I}_{i,j,g}, \mathbf{I}_{i,j,b})$ on the wavelength of primary colors. The output color of the same pixel is denoted by $\mathbf{f}_{i,j} = (\mathbf{f}_{i,j,r}, \mathbf{f}_{i,j,g}, \mathbf{f}_{i,j,b})$. We apply an energy minimization method. First a functional is established that is large when the output is far from the input or it has high variation due to precipitation, then the output image is found by minimizing the functional. Our energy functional has a *data term* $D(\mathbf{f}_{i,j}, \mathbf{I}_{i,j})$ penalizing that difference between input $\mathbf{I}$ and output $\mathbf{f}$ which cannot be caused by rain, and a regularization aka *smoothness term* $R(\mathbf{f})$ penalizing the variation caused by precipitation:

$$E(\mathbf{f}) = \sum_{i,j} D(\mathbf{f}_{i,j}, \mathbf{I}_{i,j}) + \lambda R(\mathbf{f})$$

where λ is the regularization parameter. In order to allow an efficient solution of this minimization problem, we require the energy be a convex function of optimization variables $\mathbf{f}_{i,j,c}$ where c is the index of color channels r,g,b.

Both the data and the smoothness terms contain differences of pixel colors. The data term depends on the difference of the input and output values of the same pixel, the smoothness term on the difference of the output values of neighboring pixels. In both cases, we need to consider what kind of differences precipitation can cause.

3.1. Data term

We follow the observation that the added effect of rain or snow is positive on all wavelengths and natural. Let us consider the difference of the observed and true color values

$$\mathbf{I}_{i,j} - \mathbf{f}^{\text{true}}_{i,j} = (\Delta r, \Delta g, \Delta b), \quad \Delta m = \min(\Delta r, \Delta g, \Delta b). \tag{1}$$

If minimal gray difference Δm is positive, then rain of luminance at most Δm could be the cause of the increase from $\mathbf{f}^{\text{true}}$ to $\mathbf{I}$. The data term should only minimally penalize those differences between the input $\mathbf{I}$ and the output $\mathbf{f}$ which are caused by the precipitation. However, spectral changes $(\Delta r - \Delta m, \Delta g - \Delta m, \Delta b - \Delta m)$ cannot be explained with added rain, thus these should contribute to the penalty. If $\Delta m \geq 0$, then an appropriate penalty is

$$D = (\Delta r - \Delta m)^2 + (\Delta g - \Delta m)^2 + (\Delta b - \Delta m)^2 + \xi(\Delta m)^2. \tag{2}$$

Factor ξ controls the strength of penalizing added natural color. As we wish to preserve variations except those caused by precipitation, penalty of color changes and luminance decrease should be much higher than that of luminance increase, thus parameter ξ must be close to zero. We used $\xi = 0.01$ in our experiments.

If Δm is negative, then color $\mathbf{I}$ cannot be the rainy version of $\mathbf{f}^{\text{true}}$, thus the total difference $(\Delta r, \Delta g, \Delta b)$ is due to factors other than rain. In such cases the optimal choice is $\mathbf{f} = \mathbf{I}$, so other values should be penalized based on their distance from the optimum:

$$D = (\Delta r)^2 + (\Delta g)^2 + (\Delta b)^2 \tag{3}$$

Note that this is an asymmetric but a convex function of the optimization parameter.

3.2. Smoothness term

The smoothness term depends on the changes i.e. gradients caused by the precipitation. As rain adds neutral colors, it modifies all channels in the same direction causing changes of luminance and not chroma. Therefore, we use the gradient of the luminance

$$l = (r + g + b)/3 \tag{4}$$

in our smoothness term if the gradient of all color channels have the same sign. If color channels change in different directions, this difference is not caused by rain, and therefore not penalized setting the gradient to zero. Note that with this decision, the changes due to the regularization are always gray, thus hues of the original image are preserved.

As the rain streak is strongly directional phenomenon, we apply a direction sensitive gradient, which emphasizes horizontal changes by factor 20. The gradient at pixel (i,j) can be estimated by finite differences in the discrete data. Here the crucial choice is the norm used in the regularization term. L_1 or *Total Variation regularization* would take the sum of absolute values:

$$R(\mathbf{f}) = \sum_{i,j} |\vec{g}_{i,j}|. \tag{5}$$

Although, Total Variation regularization does not blur edges as L_2 regularization does, but it also has problems. When its derivatives are computed, we should solve the problem that the absolute value function has no derivative at zero. We can use the following approximation where β is a small positive constant:

$$|\vec{g}_{i,j}| \approx \sqrt{\vec{g}^2_{i,j,x} + \vec{g}^2_{i,j,y} + \beta}. \tag{6}$$

Total Variation measures the distance between the constant function and the actual estimate. The variation of the true solution is also penalized, so the optimum would be modified. As a result, Total Variation regularization results in reduced contrast solutions having stair-case artifacts. In theory, the optimal weight can be obtained with *Hansen's L-curves* [Han00] but it has no practically feasible generalization to larger scale problems. Setting the regularization term is a trial and error process, and the algorithm is quite sensitive to it. These problems can be attacked by Bregman iteration, which makes the method less sensitive to the regularization parameter.

3.3. Bregman iteration

An optimal regularization term would have its minimum at the ground truth solution when $\mathbf{f} = \mathbf{f}^{\text{true}}$, and would measure the "distance" $B(\mathbf{f}, \mathbf{f}^{\text{true}})$ between $\mathbf{f}$ and $\mathbf{f}^{\text{true}}$. An appropriate distance function is the *Bregman distance* [Bre67, YODG08, YBCV13, SKTJ14] that can be based on an arbitrary convex penalty term, for example, on the Total Variation:

$$B(\mathbf{f}, \mathbf{f}^{\text{true}}) = R(\mathbf{f}) - R(\mathbf{f}^{\text{true}}) - \langle \mathbf{p}, \mathbf{f} - \mathbf{f}^{\text{true}} \rangle \tag{7}$$

where $\langle \cdot, \cdot \rangle$ is the $3N$ dimensional scalar product where N is the number of pixels in the image, and $\mathbf{p}$ is the gradient of $R(\mathbf{f})$ at $\mathbf{f}^{\text{true}}$, i.e. $\mathbf{p}_{i,j,c} = \partial R / \partial \mathbf{f}_{i,j,c}$ if it exists. Note that R is not required to be differentiable everywhere, thus we can use inequalities in the definition of the smoothness term.

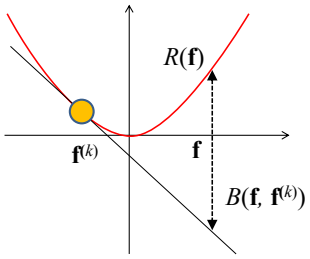

Figure 1: *Bregman distance in 1D.*

In practice, we do not know the true solution, so it is replaced by an earlier estimate $\mathbf{f}^{(k)}$. Note that if the regularization term is linear between the true solution and $\mathbf{f}^{(k)}$, then this approximation is precise. Total variation is based on the absolute value function, which results in a piece-wise linear regularization term, making it particularly attractive for Bregman iteration.

Inserting the Bregman distance into the energy, we get

$$E = \sum_{i,j} D(\mathbf{f}_{i,j}, \mathbf{I}_{i,j}) + \lambda B(\mathbf{f}, \mathbf{f}^{(k)}). \tag{8}$$

This energy term is minimized with gradient descent, thus in iteration step n, processed image $\mathbf{f}^{(n)}$ is updated as

$$\mathbf{f}^{(n+1)} = \mathbf{f}^{(n)} - \alpha \nabla E \tag{9}$$

where α controls the step size.

The gradient of the energy term is

$$\frac{\partial E}{\partial \mathbf{f}_{i,j,c}} = \frac{\partial D(\mathbf{f}_{i,j}, \mathbf{I}_{i,j})}{\partial \mathbf{f}_{i,j,c}} + \lambda \left(\frac{\partial R}{\partial \mathbf{f}_{i,j,c}} - \mathbf{p}^{(k)} \right). \tag{10}$$

Using our definitions, the partial derivative of the data fidelity term with respect to channel r can be computed as:

$$\frac{\partial D(\mathbf{f}_{i,j}, \mathbf{I}_{i,j})}{\partial \mathbf{f}_{i,j,r}} = 2 \begin{cases} \Delta g + \Delta b - (2+\xi)\Delta r & \text{if } \Delta r = \Delta m \geq 0, \\ \Delta m - \Delta r & \text{if } \Delta r \neq \Delta m \geq 0, \\ -\Delta r & \text{otherwise.} \end{cases} \tag{11}$$

where $\mathbf{I}_{i,j} - \mathbf{f}_{i,j} = (\Delta r, \Delta g, \Delta b)$. Other color channels can be handled similarly. The partial derivative of the smoothness term is

$$\frac{\partial R}{\partial \mathbf{f}_{i,j,c}} = \frac{\vec{g}_{i,j-1,x}}{\sqrt{\vec{g}^2_{i,j-1,x} + \vec{g}^2_{i,j-1,y}}} + \frac{\vec{g}_{i-1,j,y}}{\sqrt{\vec{g}^2_{i-1,j,x} + \vec{g}^2_{i-1,j,y}}} + \frac{\vec{g}_{i,j,x} + \vec{g}_{i,j,y}}{\sqrt{\vec{g}^2_{i,j,x} + \vec{g}^2_{i,j,y}}} \tag{12}$$

if gradient $\vec{g}$ is non-zero.

When k is incremented, the gradient vector of the total variation, $\mathbf{p}$, should be updated. This can be done directly considering the criterion of optimality, i.e. the gradient of the energy is zero:

$$(\nabla E)(\mathbf{f}^{\text{true}}) = (\nabla D)(\mathbf{f}^{\text{true}}) + \lambda \left((\nabla R)(\mathbf{f}^{\text{true}}) - \mathbf{p}^{(k)} \right) = 0 \tag{13}$$

and $\mathbf{p}$ is the gradient of the Total Variation R:

$$\mathbf{p}^{(k+1)} = (\nabla R)(\mathbf{f}^{\text{true}}). \tag{14}$$

From these, we can obtain

$$\mathbf{p}^{(k+1)} = \mathbf{p}^{(k)} - \frac{1}{\lambda} \nabla D. \tag{15}$$

The number of k increments should be small, otherwise the rain is introduced back into the solution.

4. Results and discussion

The proposed algorithm has been implemented in OpenGL/GLSL. As a single step of the gradient descent is similar to convolution, the algorithm is particularly suitable for GPU implementation. The method has been evaluated on the image database of [LTG*16]. Some representative images are shown in Figure 2. Note that the new method, called Bregman, is superior to TV regularization and effectively removed rain streaks while only moderately blurring the target image. For the Bregman method, we set $\lambda = 0.4$, for the TV regularization, $\lambda = 0.2$ since TV has stronger blurring when the regularization factors are similar. Comparing to the Deep learning solution [ZP18], Bregman is more aggressive, so it leaves less streaks on the image, but also reduces the sharpness of rain-free image parts. The running time of the algorithm depends on the number Bregman iterations (we used $k \leq 5$) and the number of conjugate gradient subiteration steps (we took $n \leq 10k$).

The proposed algorithm relies on the rain model stating that rain adds gray to the ground truth. There are cases, when this is not true, thus the rain is not removed. On the other hand rain free images can also produce gray variations, which should be preserved, but the algorithm reduces them.

5. Conclusions

This paper proposed an energy minimization approach to de-raining images. We defined the data and regularization terms to penalize variations caused by rain and changes from the image that cannot be explained by added rain. To handle the non-differentiability of the regularization term and to address the problems of Total Variation regularization, we used Bregman distance in our energy functional.

6. Acknowledgements

This project has been supported by OTKA K-124124.

References

[Bre67] BREGMAN L.: The relaxation method for finding common points of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics 7* (1967), 200–217.

[CH13] CHEN Y., HSU C.: A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In *2013 IEEE International Conference on Computer Vision* (2013), pp. 1968–1975.

[GN04] GARG K., NAYAR S. K.: Detection and removal of rain from videos. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* (2004), vol. 1, pp. I–I.

[Han00] HANSEN P. C.: The L-curve and its use in the numerical treatment of inverse problems. In *Computational Inverse Problems in Electrocardiology, ed. P. Johnston, Advances in Computational Bioengineering* (2000), WIT Press, pp. 119–142.

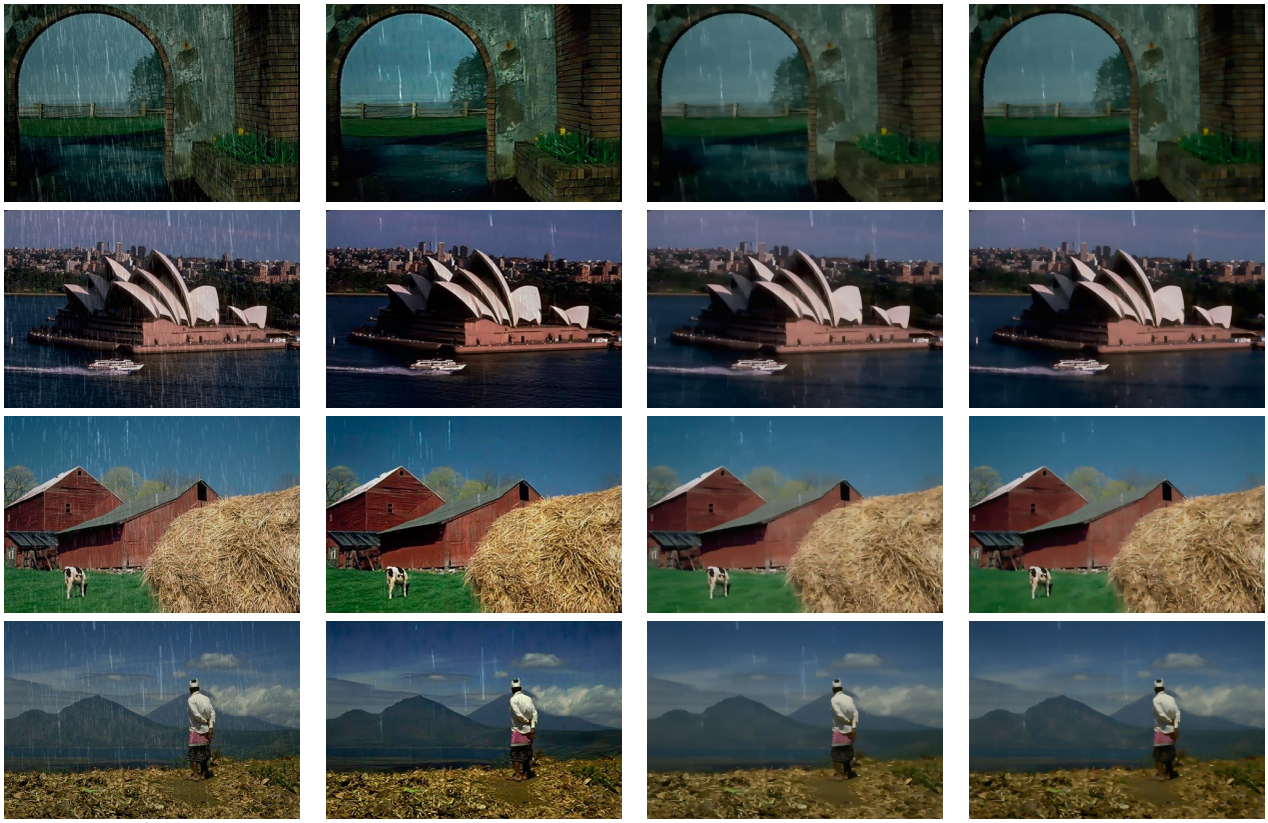

Figure 2: *Rainy and de-rained images obtained with Deep learning, TV regularization and with the proposed Bregman type approach.*

[HKWL14] HUANG D., KANG L., WANG Y. F., LIN C.: Self-learning based image decomposition with applications to single image denoising. *IEEE Transactions on Multimedia 16*, 1 (2014), 83–93.

[HMY99] HASE H., MIYAKE K., YONEDA M.: Real-time snowfall noise elimination. In *International Conference on Image Processing (Cat. 99CH36348)* (1999), vol. 2, pp. 406–409.

[JHZ*19] JIANG T., HUANG T., ZHAO X., DENG L., WANG Y.: Fast-derain: A novel video rain streak removal method using directional gradient priors. *IEEE Transactions on Image Processing 28*, 4 (2019), 2089–2102.

[KLF12] KANG L., LIN C., FU Y.: Automatic single-image-based rain streaks removal via image decomposition. *IEEE Transactions on Image Processing 21*, 4 (2012), 1742–1755.

[KLSK13] KIM J., LEE C., SIM J., KIM C.: Single-image deraining using an adaptive nonlocal means filter. In *IEEE International Conference on Image Processing* (2013), pp. 914–917.

[KSK15] KIM J., SIM J., KIM C.: Video deraining and desnowing using temporal correlation and low-rank matrix completion. *IEEE Transactions on Image Processing 24*, 9 (Sep. 2015), 2658–2670.

[LAR*19] LI S., ARAUJO I. B., REN W., WANG Z., TOKUDA E. K., JUNIOR R. H., CESAR-JUNIOR R., ZHANG J., GUO X., CAO X.: Single image deraining: A comprehensive benchmark analysis. *CoRR abs/1903.08558* (2019).

[LTG*16] LI Y., TAN R. T., GUO X., LU J., BROWN M. S.: Rain streak removal using layer priors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 2736–2744.

[LXJ15] LUO Y., XU Y., JI H.: Removing rain from a single image via discriminative sparse coding. In *IEEE International Conference on Computer Vision (ICCV)* (2015), pp. 3397–3405.

[MX11] MINMIN SHEN, XUE P.: A fast algorithm for rain detection and removal from videos. In *IEEE International Conference on Multimedia and Expo* (2011), pp. 1–6.

[SKTJ14] SZIRMAY-KALOS L., TÓTH B., JAKAB G.: Efficient Bregman iteration in fully 3d PET. In *IEEE Nuclear science symposium and medical imaging conference* (2014), MIC'14.

[WHZ*19] WANG Y., HUANG T., ZHAO X., DENG L., JIANG T.: Rain streaks removal for single image via directional total variation regularization. In *IEEE International Conference on Image Processing (ICIP)* (2019), pp. 2801–2805.

[YBCV13] YAN M., BUI A., CONG J., VESE L.: General convergent expectation maximization (EM)-type algorithms for image reconstruction. *Inverse Problems and Imaging 7* (2013), 1007–1029.

[YODG08] YIN W., OSHER S., DARBON J., GOLDFARB. D.: Bregman iterative algorithms for compressed sensing and related problems. *IAM Journal on Imaging Sciences 1*, 1 (2008), 143–168.

[ZLQ*06] ZHANG X., LI H., QI Y., LEOW W. K., NG T. K.: Rain removal in video by combining temporal and chromatic properties. In *2006 IEEE International Conference on Multimedia and Expo* (2006), pp. 461–464.

[ZP18] ZHANG H., PATEL V. M.: Density-aware single image de-raining using a multi-stream dense network. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), pp. 695–704.