Image Restoration By Removing Noise From Images

Gopal Prajapati(Assistant Professor), Diljeet singh chundawat(M.Tech Student)
Computer Science & Engg. Department, Central India Institute Of Technology, Indore, India

1 Introduction

The Images are blurred due to many reasons such as imperfections in capturing pictures, low intensity during camera exposure, atmospheric problems etc. Yet imaging, just as any other observation process, is never perfect: uncertainty creeps into the measurements, occurring as noise, and other degradations in the recorded images. The image is a projection of the real world onto the lower dimensional imaging medium, a procedure that intrinsically discards information. Sometimes the information lost may contain things we are interested in: it can be beneficial to try to recover these hidden details.

The current implementation of the RadonMAP algorithm requires quite a bit of memory. To solve for a 1 mega pixel image, it requires about 3-4GB of RAM memory[5].noise removal, often called motion deblurring or blind deconvolution, is challenging in two aspects. The first challenge is estimating blur kernels or point-spread functions (PSF) from blurred images. Because many noise–image pairs can explain the observed noisy image, blur kernel estimation is a difficult problem. Noise estimation can be especially difficult if the noise is spatially variant, for instance due to a dynamic scene or a camera rotation. The second challenge is removing the noise to recover a noise-free image. noise averages neighboring pixels and attenuates high frequency information of the scene. Consequently, the problem of recovering a noise-free image is ill-posed, and it needs to be addressed by deblurring systems or algorithms.

1.2 Research Objectives

First objective of this research is A Study of RadonMAP algorithm and factor affects to it. A RadonMAP algorithm is work efficiently in some circumstances but it also suffer from some aspects.
Second objective of this research is to Developed a modified approach to improve results that is Modified RadonMAP algorithm which work efficiently for specific type of images(such as .tiff). This proposed approach gives better result as compare to some previous algorithm of image deblurring.
Third objective of this research is the analysis of results and that results are to be compare with previously invented techniques.
Fourth objective of this research is the simulation of algorithm and results using Matlab. This proposed approach gives better performance in some specific circumstances for image deblurring.

2.1 Introduction to Blurring

Blurring in images arises from a variety of sources, like atmospheric scatter, lens defocus, optical aberration and spatial and temporal sensor integration. Human visual systems are good at perceiving it. But the mechanism of this processing is not completely understood. Therefore, it is difficult to come up with metrics to estimate blur in images .
Blur is unsharp image area caused by camera or subject movement, inaccurate focusing, or the use of an aperture that gives shallow depth of field. The Blur effects are filters that smooth transitions and decrease contrast by averaging the pixels next to hard edges of defined lines and areas where there are significant color transition .

1.4 2.2 Blurring Types

In digital image there are 3 common types of Blur effects:

2.2.1 Average Blur

The Average blur is one of several tools you can use to remove noise and specks in an image. Use it when noise is present over the entire image. This type of blurring can be distribution in horizontal and vertical direction and can be circular averaging by radius R which is evaluated by the formula:

$$R = \sqrt{g} + f \qquad (2.1)$$

Where: g is the horizontal size blurring direction and f is vertical blurring size direction and R is the radius size of the circular average blurring .

2.2.2 Gaussian Blur

The Gaussian Blur effect is a filter that blends a specific number of pixels incrementally, following a bell-shaped curve. The blurring is dense in the center and feathers at the edge. Apply Gaussian Blur to an image when you want more control over the Blur effect .

2.2.3 Motion Blur

The Motion Blur effect is a filter that makes the image appear to be moving by adding a blur in a specific direction. The motion can be controlled by angle or direction (0 to 360 degrees or –90 to +90) and/or by distance or intensity in pixels (0 to 999), based on the software used .

2.3 Introduction to Motion Blur

Motion blur is one of the salient sources of degradation in photographs. Although motion blur can sometimes be desirable for artistic purposes, it often severely limits the image quality. Blur artifacts result from relative motion between a camera and a scene during exposure. While blur can be reduced using a faster shutter speed, this comes with an unavoidable trade-off with increased noise .

One source of a motion blur is camera shake. When a camera moves during exposure, it blurs the captured image according to its trajectory. We can mitigate the camera shake blur using mechanical image stabilization hardware. However, when a camera takes a long exposure shot of a dark scene and/or when a camera uses a telephoto lens, the camera shake can be too large for assistive devices to accommodate. Another source of blur is a movement of objects in the scene, and this type of blur is harder to avoid. Therefore, it is often desirable to remove blur computationally .

2.4 Models of Motion Blur

2.4.1The general Model

As the aperture time of a camera is non zero, moving objects tend to smear in the image. The effect is more pronounced as the relative motion is faster, and the aperture time is longer.

If we think about the original and observed images as continuous functions of location (x, y) then the motion blur degradation can be modeled by:

$$g(x,y,t) = \frac{1}{Ta}\int_{t-Ta}^{t} f(x,y,t)dt \qquad (2.2)$$

Where Ta is the aperture time, f is the original image, and g is the observed motion blurred image. This formulation is commonly describes in both space and time domain .

The general motion blur is hard to treat, since each pixel can undergo a different degradation. It is even harder when the motion parameters are unknown, and there is no way to predict the motion blur. For this reason, simplified blur models are used. Two popular options are to use the constant velocity assumption or to use the uniform motion blur model.

2.4.2The Uniform Motion Blur Model

In this model the motion blur degradation can be described by a convolution with an arbitrary one dimensional kernel. This model can be formulated in the following way:

Let g denote the observed image, degraded by a motion blur with a one dimensional kernel m = (mi,..., m^) at an angle a. Let f be the original image. We assume that f was degraded in the following way:

$$g(x,y) = f \overset{\alpha}{*} m = \sum_{k=0}^{k-1} mk.f(x + k\cos(\alpha), y + k\sin(\alpha)) \qquad (2.3)$$

This assumption is valid when the motion blur is uniform over the entire image. Otherwise, the image can be divided into regions having approximately a constant motion blur. In each region, one can assume a uniform motion blur, and apply the methods based on this assumption. For a discrete image f, interpolation is used to estimate the gray levels at non-integer locations.

2.4.3 Spatially Invariant Blur

Spatially invariant blur arises when the scene is static and the camera undergoes a small out-of-plane rotation or a translation (for a constant depth scene.) A spatially invariant blur model is popular because one can exploit a simple global convolution model to describe an image formation process. Even with the spatially invariant blur assumption, however, estimating the correct blur from a single image is a challenging task due to inherent ambiguities: the observed blurry input image can be interpreted as a blurry image of a sharp scene or a sharp image of a blurry scene. This ambiguity can be address by taking multiple photos, each of which contains different blur. Taking images with modified cameras can also improve the kernel estimation performance.

2.5 DeNoising

2.5.1 Denoising Model

A blurred or degraded image can be approximately described by this equation : g = h * f + N, where g the blurred image, h the distortion operator called Point Spread Function(PSF), f the original true image and N additive noise, introduced during image acquisition, that corrupts the image.

2.5.2.1 Point Spread Function

Point spread function(psf) is the degree to which an optical system blurs(spreads) a point of light. The PSF is the inverse Fourier transforms of Optical Transfer Function (OTF) in the frequency domain, the OTF describes the response of a linear, position-invariant system to an impulse. OTF is the Fourier transform of the PSF.

Image deblurring is an inverse problem which is used to recover an image which has suffered from the linear degradation. The blurring degradation can be space-invariant or space-in variant. Image deblurring methods can be divided into two classes: nonblind, in which the blurring operator in known and blind, in which the blurring operator is unknown .

2.6 Deblurring Techniques

There are so many deblurring techniques are available to restore or to deblur the noisy image. Below listed are some techniques used for the deblurring .

2.6.1 Blind Deconvolution Algorithm Technique

The Blind Deconvolution Algorithm can be used effectively when no information about the distortion (blurring and noise) is known. The algorithm restores the image and the point-spread function (PSF) simultaneously. The accelerated, damped Richardson Lucy algorithm is used in each iteration. Additional optical system (e.g. camera) characteristics can be used as input parameters that could help to improve the quality of the image restoration. PSF constraints can be passed in through a user-specified function.

Definition of the blind deblurring method can be expressed by:

$$g(x, y) = PSF * f(x,y) + \eta(x,y) \quad (2.1)$$

Where: g (x, y) is the observed image, PSF is Point Spread Function, f (x,y) is the constructed image and η (x,y) is the additive noise term.

2.6.2 Wiener Filter Deblurring Technique

Wiener filter is used for deblurring an image in the presence of blur and noise. The frequency-domain expression for the Wiener filter is:

W(s) = H(s)/F+(s), H(s) = Fx,s (s) eas /Fx(s)

(2.2)

Where: F(s) is blurred image, F+(s) causal, Fx(s) anticausal.

2.6.3 Regularized Filter Deblurring Technique

Regularized deconvolution can be used effectively when constraints are applied on the recovered image (e.g., smoothness) and limited information is known about the additive noise. The blurred and noisy image is restored by a constrained least square restoration algorithm that uses a regularized filter. Regulated filter is the deblurring method to deblur an Image by using deconvlution function deconverge which is effective when the limited information is known about additive noise .

2.6.4 Lucy-Richardson Algorithm

The Lucy-Richardson algorithm can be used effectively when the point-spread function PSF (blurring operator) is known, but little or no information is available for the noise. The blurred and noisy image is restored by the iterative, accelerated, damped Lucy-Richardson algorithm. The additional optical system (e.g. camera) characteristics can be used as input parameters to improve the quality of the image restoration.

The Richardson–Lucy algorithm, also known as Richardson–Lucy deconvolution, is an iterative procedure for recovering a latent image that has been the blurred by a known PSF.

3.1Proposed Algorithm

As discussed in the previous section, our kernel estimation algorithm is less stable when there are not enough edges in many orientations. To handle images that do not have enough isolated Edges, we develop a method that incorporates kernel projection constraints within a MAP estimation framework.

In a MAP framework, we maximize the posterior probability with respect to the blur *k* and the image *I* jointly :

$$[k, I] = \text{argmax } p(k, I|B) = \text{argmax } p(B|k, I)\, p(k)p(I) \quad (3.6)$$

$[k, I]$ is called a maximum-a-posteriori (MAP) of the joint distribution $p(k, I|B)$. One often models the likelihood term $p(B|k, I)$ using the image observation model.

The Radon transform term relies on a strong assumption that natural images consist of step edges and that every detected edge should be an ideal step edge.

For l = 1 to 5 **do**
$I \Leftarrow \text{argmax}I\ p(k, I|B)$ % *Latent image estimation*
I $\Leftarrow$ bilateral Filtering(*I*)
$k \Leftarrow \text{argmax}k\ p(k, \mathbf{I}\,|B)$ % *Kernel re-estimation*
End for
$I \Leftarrow \text{argmax}_I\ p(k, I|B)$

Figure 3.1: The Blur Estimation Algorithm

It essentially penalizes the no-blur solution, and steers the joint distribution $p(k, I|B)$ to favour the correct solution. Algorithm shows the pseudo code for the joint estimation algorithm.

Notice that we filter the latent image estimate *I* using a bilateral filter before re-estimating the kernel. The bilateral filter step is important for improving the kernel estimation performance. *I* usually contains visually disturbing ringing and noise because the initial kernel estimate is inaccurate. If we directly use *I* to refine the blur kernel, we would be seeking a blur kernel that would reconstruct the visually disturbing image *I* from *B*. To improve the blur kernel, we bilateral-filter the latent image so that the refined blur kernel restores an image with less ringing and noise.

SIMULATION AND RESULTS

4.1 Result Analysis

This section provides experimental results that illustrate the performance of our proposed deblurring algorithm. This work, our proposed algorithm's checks the performance over RadonMAP algorithm. In order to compare just the kernel estimation performance, it has used the same deconvolution algorithm to restore images.

Figure 4.1[c], figure 4.2[c] and figure 4.3[c] shows Filtered images. In most test images, our proposed algorithm performs favourably compared to prior art. As long as we can find enough stable edges in many orientations, our algorithm can reliably estimate the kernel.

This proposed algorithm sometimes recovers blur kernels with spurious "islands" when the edge selection algorithm erroneously includes unstable edges at which edge slices intersect other neighbouring edges. A better edge selection algorithm should reduce such error.

This proposed algorithm can also be unstable when there are not enough edges, and/or when there are not enough edges in different orientations. When there are not enough edges, there simply isn't much information to estimate the kernel with; when there are only few dominant orientations in selected edges, this is only constrain the blur in those orientations and cannot recover meaningful blur kernel in other orientations. In some cases, this is less problematic. An interesting aspect of estimating the blur kernel explicitly from blurred edge profiles is that the estimated blur kernel contains enough information to properly deblur edges in those orientations, even if the blur kernel is not entirely correct.

This work recovers the horizontal component of the blur kernel, and this is what our proposed algorithm does. Kernel projection constraints assume that the image *B* is a "linear" image. In other words, the blurred image *B* is not processed by any non-linear operators such as nonlinear tone maps. This observes experimentally that the proposed algorithm is vulnerable to non-linearity in *B*, therefore it is important to properly linearize the input image *B*. In this work, this approach used only raw images as our test set in order to factor out artifacts from non-linearity. This approach observes that while competing algorithms are less susceptible to non-linearity, using raw images also improves their performance.

Chromatic aberration from a camera lens may also affect kernel estimation performance. When chromatic aberration is significant, the edge selection algorithm will discard most edge samples because an edge slice would not be explain by two dominant colours.

Here we have shown some of the results based on our proposed algorithm simulation. Below figures are results of simulation performed on MATLAB tool.

Case 1: Experiment based on image resolution 1000x553 and consider mean **μ = -0.2**

[a]

[b]

[c]

Figure 4.1: RGPV Image [a] Original Image [b] Noisy Image [c] Filtered Image.

Case 2: Experiment based on image resolution 1000x553 and consider mean **μ = -0.4**

[a]

[b]

[c]

Figure 4.2: RGPV Image [a] Original Image [b] Noisy Image [c] Filtered Image.

Case 3: Experiment based on image resolution 1000x553 and consider mean **μ = -0.6**

[a]

[b]

[c]

Figure 4.3: RGPV Image [a] Original Image [b] Noisy Image [c] Filtered Image.

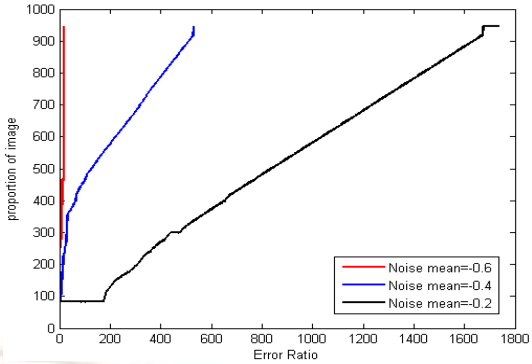

Figure 4.4: Error Ratio for estimation algorithm

Above shown figure 4.1[c], figure 4.2[c] and figure 4.3[c]shows the output of images using proposed algorithm here we are using some specific type of images (.tiff) as well as with some specific noise in each image. We have done the simulation on specific images not on all types of images so with some specific type of images we can achieve good performance compare to other algorithms but it might be perform less for all type of images. Figure 4.4 shows error ratio for estimation algorithm for multiple results for mean values □= -0.2, □= -0.4, □= -0.6 as shown in figure all together. These results are simulated using MATLAB Tool.

CONCLUSION

This thesis investigated new ideas to address to a long-standing problem in photography: Noise removal. Noise removal is challenging because many noise image pairs can explain the noisy photograph and we need to pick the correct pair from them. The challenge is aggravated since the blur can be spatially variant depending on the relative motion between the camera and the scene.

This Proposed modified RadonMAP algorithm used for removing noise from photographs. This approach showed that (I) it is possible to estimate blur kernel projections by analyzing blurred edge profiles and that (ii) it can reconstruct a blur kernel from these projections using the inverse Radon transform. This method is conceptually simple and computationally attractive, but is applicable only to images with many edges in different orientations.
This work showed that it is possible to recover a blur kernel by analyzing blurred edge profiles. One of the assumptions is that an image consists of isolated step edges oriented in different directions, and this assumption limits the algorithm's applicability. In fact, even piecewise smooth logos sometimes have thin bands at boundaries, making this algorithm inappropriate.

BIBLIOGRAPHY

[1] M. Banham and A. Katsaggelos, "Digital image restoration," IEEE Signal Processing Magazine, vol. 14, no. 2, 1997.

[2] A. Jalobeanu, J. Zerubia, and L. Blanc Féraud, "Bayesian estimation for blur and noise in remote sensing imaging," in Blind image deconvolution: Theory and Applications, P. Campisi and K. Egiazarian, Eds. CRC, 2007, Ch.-6.

[3] Rob Fergus, Barun Singh, Aaron Hertzmann, Sam Roweis, and William T. Freeman. Removing camera shake from a single photograph. ACM Transactions on Graphics (Proc. of SIGGRAPH).

[4] Anat Levin, Yair Weiss, Frédo Durand, and William T. Freeman. Understanding and evaluating blind deconvolution algorithms. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009. 31, 46, 47, 114, 129.

[5] Taeg Sang Cho, Anat Levin, Frédo Durand, and William T. Freeman. Motion blur removal with orthogonal parabolic exposures. In Proceedings of the IEEE International Conference on Computational Photography (ICCP), 2010. 28.

[6] G. Ayers and J. Dainty, "Iterative blind deconvolution method and its applications,"Optics Letters, vol. 13, no. 7, pp. 547–549, 1988.

[7] T. G. Stockham, Jr., T. M. Cannon, and R. B. Ingebretsen, "Blind deconvolution through digital signal processing." Proceedings of the IEEE, vol. 63, no. 4, pp. 678–692, 1975.

[8] A. M. Tekalp, H. Kaufman, and J.W.Woods, "Identification of image and blur parameters for the restoration of non causal blurs," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 34, no. 4, pp. 963–972, Aug 1986.

[9] "Blur identification using the expectation maximization algorithm," ICASSP,IEEE International Conference on Acoustics, Speech and Signal Processing Proceedings, vol. 3, pp. 1397–1400, 1989.

[10] R. G. Lane and R. H. T. Bates, "Automatic multidimensional deconvolution," Journal of the Optical society of America A, vol. 4, no. 1, pp. 180–188, 1987.

[11] J. Krist, "Simulation of HST PSFs using Tiny Tim," in Astronomical Data Analysis Software and Systems IV, R. A. Shaw, H. E. Payne, and J. J. E. Hayes, Eds. San Francisco, USA: Astronomical Society of the Pacific, pp. 349–353, 1995.

[12] Jian-Feng Cai, Hui Ji, Chaoqiang Liu, and Zuowei Shen. Blind motion deblurring from a single image using sparse approximation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009. 30.

[13] O. Michailovich and D. Adam, "A novel approach to the 2Dblind deconvolution problem in medical ultrasound," IEEE Trans. Med. Imag., vol. 24, no. 1, pp. 86–104,2005.

[14] O. V. Michailovich and D. R. Adam, "Deconvolution of medical images from micro scopicto whole body images," in Blind image deconvolution: Theory and Applications,P. Campisi and K. Egiazarian, Eds. CRC, 2007,Ch.-5.

[15] S. Dai, M. Yang, Y. Wu, and A. K. Katsaggelos, "Tracking motion blurred targets in video," in International Conference on Image Processing (ICIP'06), Atlanta, GA, October2006.