An Investigation of Image Deblurring Techniques

Mihai Zaharescu, Costin A. Boiangiu

Abstract— The Image Deblurring problem seems to be one of perpetual actuality. The research started in the Second World War for military purposes and continued until today proposing numerous techniques for a huge palette of applications, all aiming at recovering the original signal from a blurred one. The current paper synthesizes most of the efforts carried over the time, constructively comparing available approaches and offering a high degree of up-to-date completeness over this vast research subject.

Keywords—deconvolution, image deblurring, image enhancement, image filtering, image restoration, PSF

I. Introduction

BLURRING is the process of altering a region of a signal with weighted sums of neighboring regions of the same signal. In the case of image blurring, a pixel's value is affected by the adjacent pixels.

Blurring is usually caused by the acquisition of the same information from the scene on different receiver cells. To exemplify:

- echo is a kind of blurring, because the same sound can be localized in multiple time intervals;
- defocusing is a kind of blur because a single scene element is not found only on the pixels that is should activate, but also on neighboring pixels. It can either originate from wrongly adjusted focus distance in a camera, or the lack of focusing elements, like in the X-ray system;
- motion smudging is also a type of blur because the same signal lands on different receiver cells as the object or receiver is moving.

Important domains where deblurring is essential, are those in which a signal inherently cannot be physically focused (ex: high energy electromagnetic waves (X-rays), mechanical waves (sound/sonar)); a signal distortion varies over time (space images captured through the atmosphere; imperfect mirrors for the distances needed to be used); a signal's distortion varies in space (like a car moving in front of a surveillance camera). In these domains other methods for recovery of the original signal are not known.

With the popularization of cheap camera devices deblurring can now be integrated in nonessential desktop or mobile devices for recovering personal movies, photographs or audio recordings.

Though methods for software restoration exist dating back to the Second World War, other inseparable processes, like noise addition and PSF distortion, made them applicable only for special devices and in limited scenarios (like fixing the aberration of the mirror on the Hubble telescope). However, this changed at the beginning of the 21^{st} century, when research in the domain exploded.

The image deblurring problem can be split into two distinct problems: recovering the Point Spread Function (PSF) and recovering the initial estimate using a known PSF. Blind deconvolution methods focus on recovering the PSF while non-blind methods rely on a known PSF for performing robust deconvolution.

The PSF tells how a single point is spread on the receiver and it can be estimated, either from a single image or, more accurately, from multiple images. In multi-image PSF estimation methods, objects are either followed through the image sequence [1], or the problem is mathematically constrained to become less and less ill posed by using multiple blurred [2] or a blurred noisy image pair [3]. In single image PSF estimation, the blurred edges of objects represent the sources of motion information [4] at local level. At global level the comparison of the gradients of an entire image with a known general estimate [5] can be used to deduce the PSF.

On the other hand, non-blind deconvolution methods address the problems of minimizing the huge impact additive noise has in deblurring with a known PSF [6] or the elimination of artifacts originating from approximate PSF estimations [7], [8], [9] and truncation of data in the altered image [10], [11].

II. The Naïve Method

The definition of convolution in discreet space is:

$$(C*P)[x] = N[x] + \sum_{i=-\infty}^{\infty} C[i]P[x-i] \quad (1)$$

where C is the clear source signal, P the PSF and $*$ is the convolution operator. This means that every point in the discreet space is affected by its neighbors, weighted by elements of the PSF. Thinking back, the initial image can be estimated by removing the neighbors weighted by the PSF, from each pixel. But as the neighbors themselves are affected by the convolution, the mathematical solution is a linear system of equations.

The problems faced with the naive method are:

- the system is under-determined, thus pixels from borders cannot be calculated precisely, because a clear border is missing;
- the actual computation needs very accurate precision in order not to propagate errors through the very large system;
- the process involves determining the inverse of a very large matrix. One of the fastest methods is the conjugate gradient method, which converges after N iterations, where N is the number of unknowns. Thus, the minimal complexity, speed and memory, is:

$$O(M \cdot N \cdot \log(N \cdot M)) \tag{2}$$

where M is the size of the PSF kernel. Due to this complexity, finding the solution for a large image becomes virtually impossible.

In order to address these problems the following has been done:

- To avoid introducing any artifacts from the unknown borders, a test image was generated from a clear photo by applying convolution only to the center of the image. The result is a blurred image with known borders.
- The complexity was minimized by using a programmatic approach with events: The algorithm tries to find the solution of the top left pixel. Every time a pixel value is inquired, it recursively tries to find the solution to the inquired pixel, thus propagating to the bottom right corner. Every time the program finds the solution to a pixel, it generates an event telling that the respective pixel is now known. So, every pixel that needed that value, can now update it, thus generating more known pixels. For a fast propagation, every equation holds the unknowns in a hash table along with their weights, ensuring $O(1)$ retrieval. Likewise, every time an unknown is assigned to an equation, the unknown pushes in a stack the reference to the respective equation, in order to inform it. The total complexity resembles that of a simple convolution: $O(N*M)$ (N, the number of equations multiplied by M, the number of pixels added to each equation, plus $N*M$, the number of propagations).
- The numeric instability problem was solved by choosing the fraction as number element. The fraction's nominator and denominator are also integers unlimited in size. The method avoids all truncation. The result is that either the program finds a solution without any loss of precision, or it runs out of memory.

The results show that this method can give results in a timely manner; comparable to the simple naive convolution method (deconvolution takes 8 seconds, while the convolution takes 20 seconds, on a 2.67GHz machine) and that it is very robust to truncation errors.

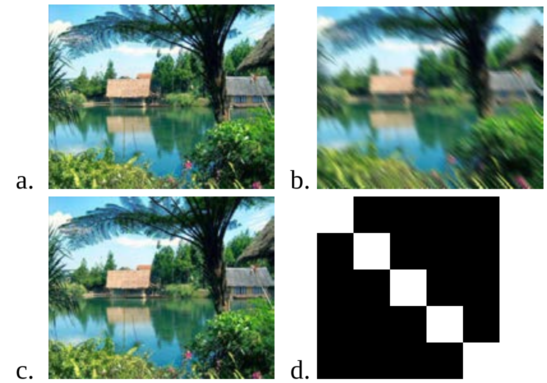


Fig. 1 a. Initial image, b. blurred 5 times, c. deblurred 5 times, d. the PSF. No loss of precision is evident

Even though the method works inside the program, as soon as the image is exported, additive noise affects it, thus, the equation becomes:

$$(C*P)[x] = N[x] + \sum_{i=-\infty}^{\infty} C[i]P[x-i] \tag{3}$$

where N is random noise. Even if N is very small, it propagates through the system, generating significant alterations. With every propagation, the error accumulates, the final pixels ending up with a small signal to noise ratio, dependent of the number of pixels that propagated the result multiplied by the error each pixel applies.

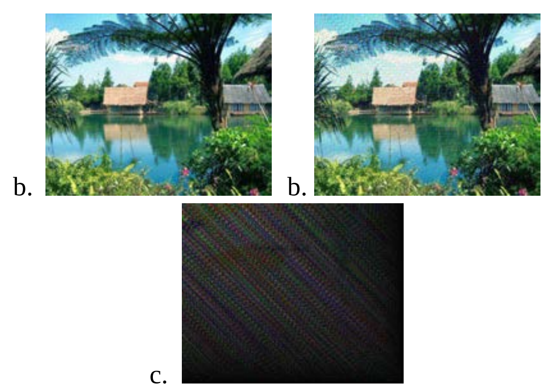
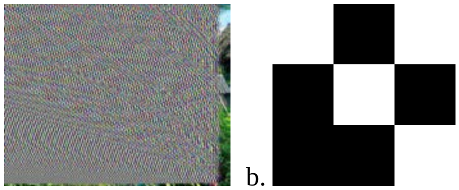


Fig. 2 a. Deblurring after bmp compression; b. Deblurring after jpg compression (quality 90); c. The propagation of noise

If a more complicated kernel is used, the errors propagate not only along lines, but also from line to line, thus the image ends up being indistinguishable after just a few propagations.



d. b.

Fig. 3 Deblurring using a more complicated PSF

In conclusion, the naive algorithm cannot be used without a regularization method.

III. Regularization Techniques

A. Introduction

It was seen that deconvolution is an ill posed problem, either because the convolution kernel cannot be deduced exactly, because clear signal information is missing from the edges of the image or because additional unknown noise signal is present in the source image.

Even the smallest perturbation propagates and accentuates in the deconvolution process.

Regularization techniques aim to attenuate the great impact these unknowns have, by introducing additional information in the system.

One example is introducing a constraint in the equations, so the result has small total variance. Signals with excessive and possible fake detail have great total variation (the integral of the absolute gradient). The last result is a clear illustration of this case: the wanted signal was too faint compared to the excessive generated detail. Introducing a variation constrain in the system can generate pleasing results.

The work done by Jalobeanu et all on recovering signal from satellite images [12] showed that even in the highly noisy photo resulted from deconvolution resides a recoverable, separable and powerful enough useful signal, which they obtained by using complex wavelet packets. The reason why they used oriented wavelet packets and not the standard wavelet transform is because noise generated from deconvolution is not white noise but colored. It can be seen that every error propagates up and down through the image at regular intervals. Any thresholding on a normal wavelet transform is unable to eliminate such strong elements.

B. Richardson-Lucy Deconvolution

As seen previously, a way of smoothing out pixels that explode numerically is needed. One way of doing this is by processing the image iteratively and stopping the iteration process when the photo becomes unstable.

Now, since a clear image C exists, both an estimate P PSF and the resulting blurred image B are known. For every pixel, its equation can be written as:

$$B_k = \sum_j P_{kj} C_j \tag{4}$$

meaning that the k-th pixel of the blurry image is the weighted sums of the neighbors, the weights being read from N.

Because the kernel moves over every pixel, it is necessary to use an iterative method that restores little by little the information, as modifying a neighbor pixel influences the current pixel as well.

The first step of the algorithm generates the correlation between a n initial image, which can be noise, and the PSF. The next step is to find the difference between the estimated image obtained at the first step and the actual blurred image. These differences will tell us how wrong we were when we picked an estimation of the clear image. In order to correct the original estimate, the obtained error is correlated with the inverse kernel. The iterations can continue, until the PSF is eliminated from the image completely or they can be stopped when noise becomes too evident.

$$R_j^{(i+1)} = R_j^{(i)} \sum_k \frac{B_k}{CB_k} \tag{5}$$

where $R^{(i)}$ is the result of iteration i, B is the blurred image and CB is the correlated image.

```
Image observed;
Image psf;

Image clear = random();
Image inverse_psf = inverse(psf);

Image calculated_blur, difference,
        projected_error

While (iterations > 0)
  calculated_blur = conv(clear,psf);
  difference = observed / calculated_blur;
  projected_error = conv(difference,
                              inverse_psf);
  clear = clear * projected_error
```

An article that can estimate when deconvolution method should stop is presented here [13]; if the algorithm runs for too long, noise begins to emerge, if it runs too little, then the blur is not completely eliminated.

This is the Richardson-Lucy deconvolution method and it gives similar results to the Wiener deconvolution, which will be presented in the following section.

C. Inverse Filtering

The convolution operation generates a signal that repeats the PSF characteristics over the entire input function. Analyzing the frequency spectrum, the influence of the convolution can be clearly seen. Another way of performing convolution is by applying the frequencies characteristic to the PSF on the initial function, and this can be done by multiplying the two spectra in frequency domain:

$$b = c * p \tag{6}$$

where c is the resulting convoluted image, f the initial image and g the PSF, all in frequency domain. In this context, $*$ becomes normal multiplication in frequency domain.

Deconvolution is calculated the other way around:

$$c = b / p \tag{7}$$

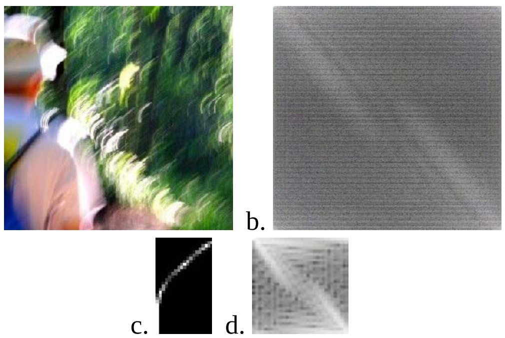


Fig. 4 A blurred image and its Fourier transform. The estimated PSF and its Fourier transform.

The observed noise in the naive method is very strong and has high frequency. Strong frequency elements are obtained when N is very small, thus, an idea of stabilizing the solution is to cut the small frequencies from the division.

In order not to eliminate the small frequencies completely, which are an important detail factor in the final image, a slightly modified threshold function, called Inverse Filtering, is employed:

$$g(t) = \begin{cases} \frac{1}{p[t]} \gamma |p[t]|, |p[t]| < \gamma \\ \frac{1}{p[t]}, |p[t]| \geq \gamma \end{cases} \tag{8}$$

where g represents the $1/n$ factor.

This method shows great improvements, but generates unwanted waves, along strong edges, because of the missing frequencies and is still badly affected by additive noise.

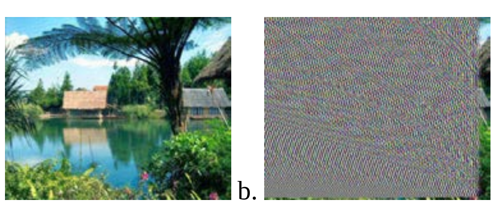


Fig. 5 a. Result of inverse filtering opposed to b. The naive method in deblurring with a complicated PSF

This method is also more stable in the case of approximate PSF functions, as in reality one cannot find the exact camera trajectory.

An experiment was performed, with a few images affected by motion blur, taken with a normal camera. The user draws with the mouse an estimation of the movement, following bright spots in the picture and also has the ability to set the time in each point of the motion curve. Using this curve, the program generates a PSF and passes it to the above mentioned method.

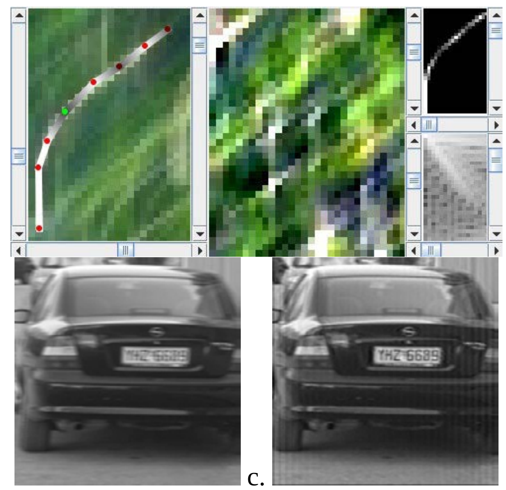


Fig. 6 a. User interface for drawing an estimate PSF, b. and c. Input and output images of inverse filtering

The results show that even the user can estimate a good enough PSF in order to recover an image.

D. *Wiener Deconvolution*

During World War II, Norbert Wiener was seeking for a way of receiving as much useful signal as possible from radar machine:

$$r(t) = g(t) * [c(t) + n(t)] \tag{9}$$

where $c(t)$ is the original clear signal, $n(t)$ is noise, $r(t)$ is the function intended to equal $c(t+a)$ and $*$ is normal multiplication, all functions being in frequency domain. $g(t)$ is the function that has the role of transforming the received signal into a close estimate of the original signal.

The error can be calculated as the difference between the initial signal, delayed by the time taken by the signal to arrive at the destination, and the original transmitted signal:

$$e(t) = c(t+a) - r(t) \tag{10}$$

Minimizing the square error

$$e^2(t) = c^2(t+a) - 2c(t+a)r(t) + r(t) \tag{11}$$

generates a filter that can restore most of the stationary signal corrupted by stationary noise, as long as the signal and noise spectra are known.

Later, this filter was adapted to work for functions like:

$$b(t) = c(t) * p(t) + n(t) \tag{12}$$

where p is a PSF. This is a convolution affected by additive noise. And the solution is the Wiener Deconvolution:

$$g(t) = \frac{p(t)s_c(t)}{\left|p^2(t)\right|s_c(t) + s_n(t)} \tag{13}$$

where s_c is the clear image spectrum and s_n the noise spectrum.

Given that g has a greater power at the denominator, the function acts as a deconvolution, but has an extra filter meant for removing noise with a known spectrum.

And the estimate clear image is:

$$c = b * g \tag{14}$$

IV. Automatic PSF Estimation

There are images where the user cannot find a clear element to follow, or the PSF isn't even a camera path, but a combination of defocus, movement and intersections. To solve this, a robust automatic method has to be developed.

This problem has its origins in space observation research, where the solution is relatively easy because stars are point-like elements. The telescopes' PSH can therefore be deduced just by photographing a distant star.

However, for a natural image, a solution could not be found, up until 2006, when Fergus's [5] research opened a big door in kernel estimation. He noticed that all natural clear photographs share a similar histogram of gradients. A blurred image changes the shape of the histogram. His approach estimates the PSF by going from small resolution to great resolution and tries to fit the resulting latent image to the mathematical gradient distribution, varying the PSF.

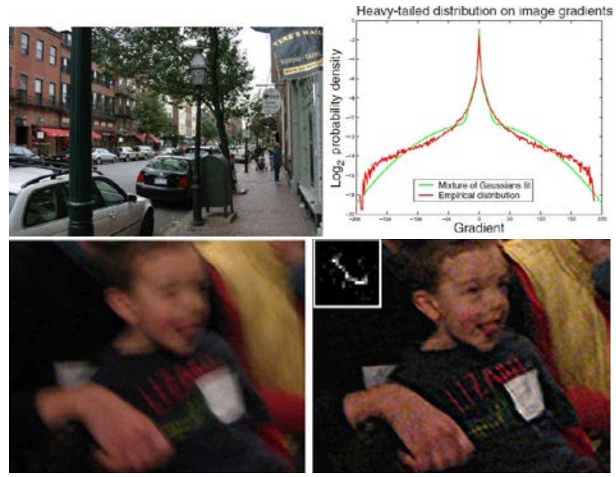


Fig. 7 Top to bottom and left to right: a natural image; its gradients (gradient and probability) compared to a general natural image gradient distribution; blurry image; recovered image and kernel using Fergus' method. Image from "*Removing Camera Shake from a Single Photograph*" [5]

An addition to the original idea is the observation that not all gradients are good for estimating the PSF [6]. Contrary to intuition, objects smaller than the kernel degenerate the prediction, thus, they should be ignored. Another small contribution is the usage of a better refinement method in kernel generation between resolutions that keep the connectivity of the pixels, which should happen when the trajectory of the camera is a connected curve.

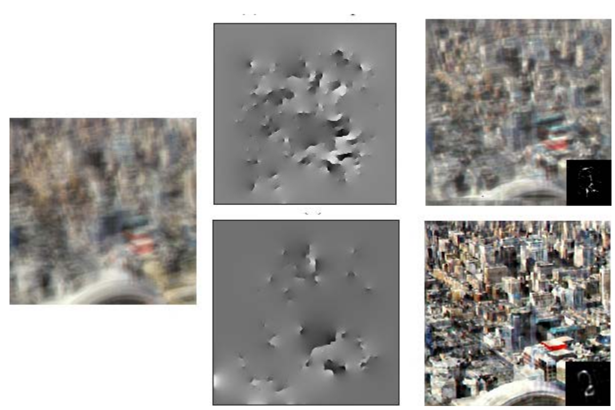


Fig. 8. Left to right: Input image; gradients used in the estimation phase; deblurring result. Image from "*Two Phase Kernel Estimation For Robust Deblurring*" [6]

Iterative methods use the result from the last step in order to compute the next image. A denoised image contains clear edges in well defined positions. It also doesn't contain wave artifacts generated from approximating missing elements from the image or the kernel, thus it is a great estimate for following iterations. The noisy/blurry image pair method [3] can give very good PSF estimates. The noise filtered sharp image is the latent image in the iterative kernel estimation algorithm. As the result converges, the deblurred image can be used to clean the noise from the sharp image. As well, the ill conditioned problem can become much less ill conditioned if more blurred images are given as input, for example, from a burst shooting or a video [2].

Methods in cepstral domain have also been proposed. They usually rely on interpreting the shape of the amplitude cepstrum. The amplitude cepstrum is obtained by taking the inverse Fourier Transform of the logarithmized amplitude of the Fourier transform. The idea is that a Fourier transform generates from a square wave, a sinc. The logarithm converts the sinc to a sin and the inverse Fourier converts from a sin into a square wave. This means that similar elements from the image will end up one over the other in the same place. As the movement is present everywhere on the image, the PSF should be evident on the cepstrum. The problem is that the phase element is thrown away, so it can have any orientation. Thus methods like this usually try to recover simple shapes from interpreting the geometrical shapes they can see in the center of the cepstrum [14], [15].

The deconvolution algorithms usually don't work well with Gaussian blurring, as high frequencies (which become very faint in the blurring phase) are eliminated in the regularization phase. This is an example aimed at resolving this particular problem, by introducing in the equations the presumption that the kernel is actually Gaussian. [16]

Most of the deblurring methods assume a shift invariant linear blur model, which means that the image is blurred the same way everywhere. This is true only if the photographed objects are at the same distance, or at great distances from the camera, in order not to introduce perspective blur, and the camera follows only a translational movement in a plane parallel to the objects. As seen in the description, not very many images fall in this class of alterations. Rotational motion blur is the simplest example to show that the blur kernel changes at every pixel of the image (fragments of concentric circles). Blur caused by individual moving objects is even harder to describe.

Two approaches [4], [17] try to deblur moving objects from static backgrounds. Firstly they separate the blurred elements and use only the transparent edges for estimating the motion direction. They cut out the moving objects by means of spectral mating [18], thus preserving the transparent shading left by the blur. The authors of the first article try to automatically deduce the movement in a simple manner (reducing the local kernel to a line), whilst the others need the user input in order to get an estimate of some local motions, which they interpolate.

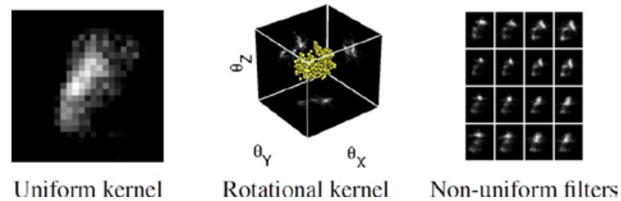


Fig. 9 3D kernel used to estimate nonuniform kernel shapes over a blurry image. Image from "*Non-uniform Deblurring for Shaken Images*" [HHH]

Both methods give good results, with the first being able to correctly estimate localized movement whereas the second uses a better deconvolution method.

Another solution, which is used in multiple uniform moving objects, is to break all the moving elements into layers, using their motion print, deblurring each element separately and combining the fragments in the final image. [19]

V. Artifact Minimization

A. Deringing

There are now available methods for estimating the PSF and removing much of the amplified noise. Another artifact that is most unpleasant in image deblurring is ringing. Because the PSF mostly has null values, the inverses are very large values which amplify in excess frequencies, especially at borderlines, generating a periodic ripple near them. In spatial domain iterative methods, the initial estimation error propagates and accumulates through iterations, becoming more visible near the edges, where the correlation was most intense. Moreover, the PSF cannot be accurately estimated in reality.

Photographing in dim light conditions is difficult, as the signal is too low compared to noise [3]. If one increases the exposure time in order to receive more useful signal, camera movement blurs the photograph. Various methods which correct one of the two exist, but with limitations: noise reduction algorithms eliminate fine detail whilst deblur algorithms generate the artifacts mentioned earlier. One interesting approach is to use an iterative method which takes what is good from each image, using the other as reference [3]. In the first iteration, a general denoise algorithm cleans the noisy image. The deblur algorithm deblurs the moved image using the cleaned image as base. The difference between the clean and noisy image generates a noise layer and the difference between the deblurred and clean image reveals the artifacts, or wave layer. The rings can now be eliminated without losing precious texture information.

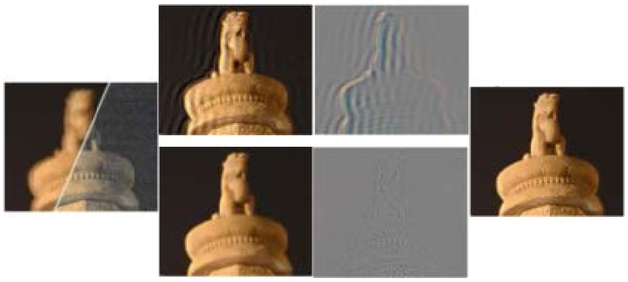


Fig. 10 Description of the iterative process of the deblurring method which uses a blurred/noisy image pair. Image from “*Image Deblurring With Blurred/Noisy Image Pairs*” [3]

Another similar approach is to use just the blurred image as a base for estimating ringing artifacts[9]. After a general deblur algorithm generates the sharp result, the deringing algorithm takes into consideration only the initial, affected image and the clarified image. Using the unclear photo, it deduces uniform patches that are likely to suffer from long range ringing resulting from far away strong edges. Afterwards it identifies small regions around edges in the clarified image, which can suffer from short range ringing. The waves are then removed by a filter that is dependent on the wave size, or the distance from the edge.

Fig. 11 From left to right: input blurred image; recovered image with Richardson-Lucy algorithm; the RL result cleaned with the mentioned algorithm. Image from “*Dual range deringing for Non-blind image deconvolution*” [9]

One great idea that makes the ringing problem obsolete is to use both intra-scale (the deconvolution is being fine tuned inside the respective resolution) and inter-scale (using the result from precedent resolution) elements in the deconvolution process [8]. The method starts with a small resolution that represents the base clarified image for the next greater resolution. Afterwards it computes the greater resolution by an iterative Joint Bilateral Richardson Lucy deconvolution. The resulting edge detections from the coarser resolution image is a base for a more accurate edge detection in the finer resolution image. With the aid of accurate edge detections, a regularization method removes unwanted artifacts in uniform areas. Moreover, using the smaller resolution as guide, and a residual deconvolution algorithm, more and more details can be recovered. This method eliminates ringing entirely and also generates a sharp image with insignificant texture loss.

Fig. 12 Left: input blurred image and kernel; right: result of interscale-intrascale algorithm. Image from “*Progressive Inter-Scale Non-Blind Image Deconvolution*” [8]

B. Outliers Handling

The mathematical model presented before takes into account only Gaussian additive noise. In reality, there are other aberrations that can disturb the convolved image. For example: when taking pictures during night time, some bright spots, where the lights are present, appear on the photo. Those bright spots have intensities whose values go beyond the limited range of values provided by the image format specification, and are thus are clipped to the greatest value. This clipping, along with dead pixels or hot pixels are not taken into account in the original theoretical model. Other influences are color curves introduced by software in order to capture an image more similar to what can be seen.

One proposed solution is to first remove the color curve by applying a gamma correction, so the colors vary linearly. Afterwards, the outliers elimination algorithm separates pixels that respect the model from those that could be possible errors (saturated and dark pixels). An Expectation Maximization method fills the areas where pixels were removed [10].

This model removes the very evident repetitive and wave like artifacts that originate from software truncations and hardware errors. It also generates far fewer rings caused by nonlinear color transformations, present in all photos taken by ordinary cameras today.

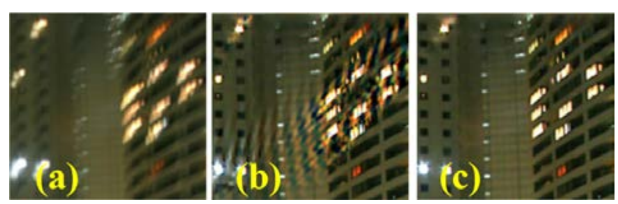


Fig. 13 Left to right: input image; standard deconvolution (waves propagate from areas where information is lost); outliers handling. Image from “*Outliers in Non-Blind Image Deconvolution*” [10]

C. Noise Reduction

The regularization techniques presented before have the principal role of minimizing the influence of small noise signals in convolved images. The practical problem is that the majority of blurred images have a significant amount of noise, because they are captured in a medium where the signal is weak over a large period of time (space telescopes have the signal source very far away, medicinal imaging use a small quantity of radiation in order to minimize its impact on the patient, the photo camera compensates with time for a night scene). The result is that noise is comparable to the signal power. In these conditions, the regularization techniques are inefficient at providing a good result.

Wohlberg and Rodrigues developed a mathematical model which deals with impulse noise alone. [20] The solution is a modified Total Variance (TV) regularization, which generates an image with the smallest variations between pixels that still follow the original signal's shape. The variance is defined as:

$$\frac{\lambda}{q}\left|\sqrt{(D_x u)^2+(D_y u)^2}\right|_p^p \tag{15}$$

where D is the derivative and lambda the power of the filter. And the measure of how close the generated signal is to the original one is the p norm of:

$$\frac{1}{p}\left|K_u - s\right|_p^p \tag{16}$$

where K is a linear operator representing the forward problem and s is the altered signal.

Both of these functions are modified in order to accommodate pixels that fall over of below a threshold, in order to locate and eliminate salt and pepper noise.

The authors of "Two-Phase Kernel Estimation for Robust Motion Deblurring" [6] use a similar but faster technique, which still produces good results for impulse noise and moderate results for Gaussian noise.

Knowing that a natural image has most derivates round 0, a sparse prior that opts to concentrate the derivates at a small number of pixels, the rest leaving almost unchanged in the deconvolution process. [21] This way, the image has sharp edges, less noise and smaller ringing artifacts, but fine texture details are lost due to the convolution.

One very interesting solution [12] does not use regularization at all and generates impressive results. The simplest deconvolution algorithm generates an unregularised result which contains the entire, unfiltered signal hidden in an image that looks just like noise. The novel idea is in filtering the result with a special kind of wavelet packets. Instead of using wavelets on lines or columns, which can only detect horizontal or vertical signal orientations, the authors developed 26 orientated wavelets for different scales and orientations. Being able to characterize the signal in 26 different ways, the noise at known power was very well separated from the orientated texture.

VI. Conclusions

In the past years, deconvolution proved to be a resolvable problem that can aid in many domains, from medical imaging to space photography. The theoretical problems that made deconvolution be overlooked for everyday photography until the third millennium, like ill conditioned systems, high ratios of noise amplification because of inversion of small values in the blur kernel, artifacts originating from real values truncation and others, found their solutions with practical approaches in a very short time. Another very interesting domain, super resolution can now be enhanced by the aid of this new technology, by removing the blur that inherently is generated when the combination of multiple images ends [22], or by reading more information from the larger space occupied by the moved object on the image [23], [24] .

This domain has proven that is now ready to be used in everyday applications, like introducing special camera aperture [25] or coded camera exposures [23] that can aid the software editing of blur in photographs, modifying the medical instruments so that they incorporate these algorithms in order to give clearer results.

References

[1] M. Ben-Ezra, S. K. Nayar, “Motion-based motion deblurring,” *Pattern Analysis and Machine Intelligence*, IEEE Transactions, vol. 26, no. 6, pp. 689 – 698.

[2] J.-F. Caia, H. Jib, C. Liua, Z. Shenb, “Blind motion deblurring using multiple images,” *Journal of Computational Physics*, vol. 228, no. 14, Aug., 2009, pp. 5057-5071.

[3] L. Yuan, J. Sun, L. Quan, H.-Y. Shum, “Image deblurring with blurred/noisy image pairs,” *ACM Transactions on Graphics (TOG)* - Proceedings of ACM SIGGRAPH 2007 TOG Homepage, vol. 26, no. 3, July 2007, article no. 1.

[4] S. Dai, Y. Wu, “Motion from blur,” *Computer Vision and Pattern Recognition*, 2008. CVPR 2008. IEEE Conference, 23-28 June 2008, pp. 1-8.

[5] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, W. T. Freeman, “Removing camera shake from a single photograph,” *ACM Transactions on Graphics (TOG)* - Proceedings of ACM SIGGRAPH 2006 TOG Homepage, vol. 25, no. 3, July 2006, pp. 787 – 794.

[6] L. Xu, J. Jia, “Two-phase kernel estimation for robust motion deblurring,” *ECCV'10 Proceedings of the 11th European conference on Computer vision*: Part I, pp. 157-170.

[7] Q. Shan, J. Jia, A. Agarwala, “High-quality motion deblurring from a single image,” *ACM Transactions on Graphics (TOG)* - Proceedings of ACM SIGGRAPH 2008 TOG Homepage, vol. 27, no. 3, August 2008, article no. 73.

[8] L. Yuan, J. Sun, L. Quan, H.-Y. Shum, “Progressive inter-acale and intra-scale non-blind image deconvolution,” *ACM Transactions on Graphics (TOG)* - Proceedings of ACM SIGGRAPH 2008 TOG Homepage, vol. 27, no. 3, Aug. 2008, article no. 74.

[9] L. Zouy, H. Zhouz, S. Chengx, C. Heyy, “Dual range deringing for Non-blind image deconvolution,” *Image Processing (ICIP)*, 2010 17th IEEE International Conference, 26-29 Sept. 2010, pp. 1701 – 1704.

[10] S. Cho, J. Wang, S. Lee, “Handling outliers in non-blind image deconvolution,” *Computer Vision (ICCV)*, 2011 IEEE International Conference, 6-13 Nov. 2011, pp. 495 – 502.

[11] J.-H. Lee, Y.-S. Ho, “Non-blind image deconvolution with adaptive regularization,” *PCM'10 Proceedings of the 11th Pacific Rim conference on Advances in multimedia information processing*: Part I, pp. 719-730.

[12] A. Jalobeanu, L. Blanc-Féraud, J. Zerubia, “Satellite image deconvolution using complex wavelet packets,” *Image Processing, 2000*. Proceedings. 2000 International Conference, 2000, vol. 3, pp. 809-812.

[13] P. Marcelo, J. Daniel, “An iterative SNR estimation algorithm for wiener deconvolution of self-similar images distorted by camera shake blurring,” *8th WSEAS International Conference on Signal, Speech and Image Processing* (SSIP '08) Santander, Cantabria, Spain, September 23-25, 2008, pp.97-100.

[14] W. Fawwaz, A. Maki, S. Shimahashi, M. Matsubara, S. Sugimoto, “PSF estimation and image restoration for noiseless motion blurred images,” in *Proc. 7th WSEAS International Conference on Signal,*

Speech and Image Processing (SSIP'07), Beijing, China, September 15-17, 2007, pp. 1-7.

[15] W. Fawwaz. A. Maki, S. Sugimoto, "Blind deconvolution algorithm for spatially-invariant motion blurred images based on inverse filtering and DST," *International Journal Of Circuits, Systems And Signal Processing*, vol. 1, no. 1, 2007, pp. 92-100.

[16] R. C. Bilcu, M. Trimeche, S. Alenius, M. Vehvilainen,, "Blur Space Iterative De-blurring," *ICCOM'05 Proceedings of the 9th WSEAS International Conference on Communications*, 2005, article no. 16.

[17] Q. Shan, W. Xiong, J. Jia, "Rotational motion deblurring of a rigid object from a single image," *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference*, pp. 1 – 8.

[18] A. Levin, A. Rav-Acha, D. Lischinski, "Spectral matting," *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference*, 17-22 June 2007, pp. 1 – 8.

[19] S. Cho, Y. Matsushita, S. Lee, "Removing non-uniform motion blur from images," *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference*, 14-21 Oct. 2007, pp. 1 – 8.

[20] B. Wohlberg, P. Rodriguez, "An L1-Tv algorithm for deconvolution with salt and pepper noise," *ICASSP '09 Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 1257-1260.

[21] A. Levin, R. Fergus, F. Durand, W. T. Freeman, "Deconvolution using natural image priors," *ACM Trans. Graphics*, vol. 26, no. 3, 2007, pp. 0-2.

[22] M. Irani, S. Peleg, "Super resolution from image sequences," *10th ICPR*, vol. 2, Jun 1990, pp. 115-120.

[23] A. Agrawal, R. Raskar, "Resolving objects at higher resolution from a single motion-blurred image," *Computer Vision and Pattern Recognition*, 2007. *CVPR '07. IEEE Conference*, 17-22 June 2007, pp. 1 – 8.

[24] M. Zaharescu, C.-A. Boiangiu, "Image deblurring: challenges and solutions," in *Proc. of 12th WSEAS International Conference on Circuits, Systems, Electronics, Control & Signal Processing* (CSECS '13), Budapest, Hungary, December 10-12, 2013, pp. 187-196.

[25] A. Levin, R. Fergus, F. Durand, W. T. Freeman, "Image and depth from a conventional camera with a coded aperture," *ACM Transactions on Graphics (TOG)* - Proceedings of ACM SIGGRAPH 2007 TOG Homepage, July 2007, vol. 26, no. 3, article no. 70.