A Real-Time Method to Estimate Motion Blur Kernel on Discrete Wavelet Transform

Ting-Han Lin, Chiou-Shann Fuh

Graduate Institute of Biomedical Electronics and Bioinformatics, National Taiwan University

E-mail: {r01945021, fuh}@csie.ncku.edu.tw

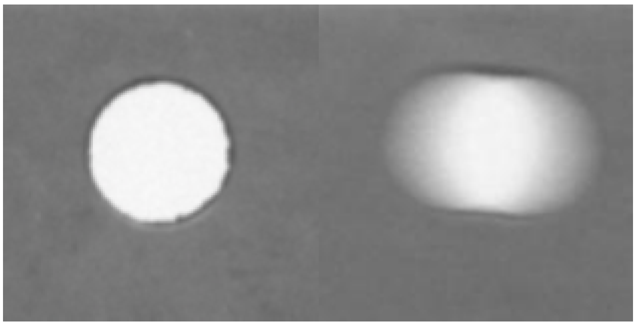
1. Abstract

Motion blur is a common phenomenon in photography. Blur may render the image useless. In many inspection industries, blur will cause large error and cannot be ignored. In this work, we simplify the problem of dealing with motion blur with constant velocity k and moving horizontally. Based on previous work, we propose a real-time method in wavelet domain to estimate the motion blur kernel on autocorrelation and hill climbing searching algorithm. With the blur kernel information, we are able to correct the proper coefficients and to remove motion blur for blurry image.

Our contribution in this work is we improve the searching algorithm instead of global searching. Compared with previous method, the improvement can reduce more than half of estimated time and get the same result. It is also the key factor to execute motion deblur in real time.

2. Introduction

Motion blur is a common phenomenon in photography caused by high relative velocity between detected object and camera. Blur may render the image useless and lead to inaccurate result. For instance, much useful information, such as edge and gradient information, will be partly distorted. In many inspection industries, blur will cause large error and cannot be ignored. For instance, in Fig 1, comparing two inspection snapshots, circle in Fig 1.b is obviously out of shape.



(a) Static object. (b) Moving object.

Figure 1. Inspection snapshots of static and moving object.

Our goal in this work is propose a real-time method to remove motion blur, especially for industry inspection system. Considering execution time, we implement our work in wavelet domain instead of Fourier domain. Besides, the key advantage of the wavelet transform over Fourier transform is that wavelet transform has the temporal information, thus the signal will not be influenced by the noise which is very far away. Since the motion blur appearance in Discrete Wavelet Transform (DWT) will be a sequence of continually increasing or decreasing signal, we use Double Discrete Wavelet Transform (DDWT) proposed by Zhang and Hirakawa [7]. The method of DDWT is literally wavelet transform twice. As long as the signal is invertible, the DDWT can be applied. Since DDWT has a very good ability to reveal the sparsity of the blur kernel, we can observe the blur appearance in the coefficients and correct the coefficients.

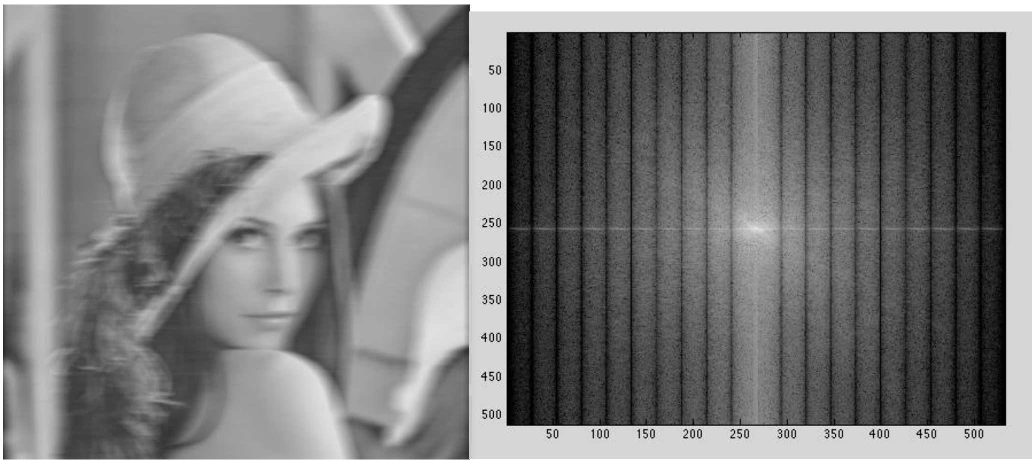
In motion blur image with DDWT analysis, double edges appear in DDWT coefficients, on the contrary, static image is not. The meaning of double edge is the same magnitude with different signs spaced in k pixels in the DDWT coefficients. Zhang proposes using autocorrelation analysis to decide the distance between double edge pair. Since autocorrelation is the cross-correlation of a signal/image with itself, Zhang considers the distance k of the minimum result of autocorrelation as the length of blur kernel. However, it is time-consuming to calculate all possible candates of autocorrelation. We use some optimal searching algorithm, hill climbing searching algorithm to reduce the execution time on estimation. Compared with previous work with this optimal strategy, we not only get the same result, but also reduce much of time.

3. Related Work

Nowadays, blur is still a challenging problem due to the unknown blur kernel estimation and noise factor. The simplest and most efficient way to reduce blur is upgrading the hardware, for instance, image stabilization, dual camera, coded shutter or coded aperture [3]. It is aslo a common way that using sequencial images to estimate the blur kernel and recontruct deblur image. All of these methods will cost extra money.

On the coding concepts, a blur image can model as clear image convolve with a blur kernel. Most deblur algorithms are implemented in the frequency domain. This makes sense since convolution operation in the spatial domain is mapped as multiplication operation in the frequency domain [6]. Over the past few decades, many frequency deconvolution researches are done [4]. We divide the methods into two categories according to the blur information: non-blind deconvolution and blind deconvolution. In non-blind method, we assume we have known the kernel information. On the other hand, in blind method, both kernel and latent images are unknown [4]. Both kinds of methods are very sensitive to noise, it causes ringing artifact and ill-posed easily. Although these approaches generate reasonable result, estimating the blur kernel and the transformation between frequency and spatial domain is time-consuming. That means, deblur in Fourier domain cannot be done in real time.

In Fourier frequency domain, we are able to observe obvious feature related to the blur kernel. In Figure 2.a, Lena image is convolved with a known motion kernel whose velocity is 20 pixels, and moves horizontally. Figure 2.b is the representation in Fourier domain of Figure 2.a. We are able to observe the dark line vertical with the object motion. Note that the number of the dark lines is exactly the velocity. That is, we can estimate the number and the angle of the dark lines in Fourier frequency representation as the motion blur kernel. There are many calculation methods to estimate the dark lines in computer vision and image processing. For instance, we can use Hough Transform or Radon transform to estimate the angle and the number of dark lines [1].



(a) Blurry image. (b) Blurry image magnitude in Fourier frequency domain.

Figure 2. Blurry Lena with k=20 pixels, angle=0^o and its Fourier representation.

There are also many deblur works in wavelet domain [2, 5, 7]. Thanks to multi-resolution and temporal information, we are able to observe blur appearance clearly in wavelet domain. In [2], Donoho et al. present a non-linear adaptive wavelet algorithm which deblurs an image with O(n2(log n)2)-steps. They exploit both the natural representation of the convolution in the Fourier domain and the typical characterization of Besov classes in the wavelet domain. In [7], Zhang proposed Double Discrete Wavelet Transform (DDWT), a new wavelet analysis method of blur kernel. The spirit of DDWT is convloving the image with two different wavelet analysis filters continuously. Due to the sparsity, DDWT differentiates the latent image and the blur kernel simultaneously after transformation. We are able to use the result of DDWT to observe the blur kernel and correct the biased coefficients. This work uses auto-correlation to estimate the size of blur kernel. However, it is time-consuming to estimate all possible candidate blur lengths. Based on this framework, we make some improvement about searching algorithm. Besides, Since we want our work to be done in real-time, some details can be simplified with simpler and more efficient way. We will discuss the methods in the following section.

4. Methodology

4.1 Blur Model

Relation between blur and clear image can be formulated in following equation:

$$y(n) = \{x * h\}(n) + \varepsilon(n) \qquad (1)$$

where * denotes convolution; x is the latent image; h is blur kernel, also called point spread function; ε is some unknown additive noise; and n is pixel coordinate in spatial domain.

4.2 Double Discrete Wavelet Transform

Discrete Wavelet Transform (DWT) is a useful signal analysis tool due to its multi-resolution. It has been widely used in many image processing researches. With the proper wavelet function and scaling function, it decomposes the given signal into approximation coefficients and detail coefficients. Here, we consider Ψ(t), the mother wavelet function as the low-pass filter, and ϕ(t), the scaling function as the high-pass filter. Based on the scale of wavelet function and scaling function, DWT can be implemented into multiple scale levels. We observe different information in different levels.

$$w^j(n) = \{d^j * y\}(n) \qquad (2)$$

where d^j is the wavelet analysis filter of scale *j*; w^j is the detail coefficients of scale *j*; *y* is the input image; and * denotes a convolution operator.

After introducing DWT, Double Discrete Wavelet Transform (DDWT) is a new analysis method in wavelet domain proposed by Zhang [14]. The spirit of DDWT is to transform the signals/images with two different wavelet analysis filters continuously: d^i and d^j. We can allow d^i and d^j to be arbitrarily defined as long as both of them are invertible. DDWT is defined by the following relation:

$$v^{ij}(n) = \{d^i * w^j\}(n) \qquad (3)$$

where d^i is a high pass filter of scale *j*; v^{ij} is the DDWT coefficients of scale *j*; w^j is detail coefficients of discrete wavelet transform; and * denotes a convolution operator.

4.3 Haar Wavelet Transform

In this work, we use Haar wavelet as our basis wavelet. Haar wavelet transform is one of the earliest transform functions proposed. All of the wavelets are formed from mother wavelet function ψ(t) and scaling function ϕ(t) by convolve with the input signals. These functions aim to compute the difference and average respectively.

Haar wavelet's function ψ(t) can be described as following equations:

$$\psi(t) = \begin{cases} 1, & 0 \le t < 1/2 \\ -1, & 1/2 \le t < 0 \\ 0, & otherwise \end{cases}$$

And scaling function ϕ (t) can be described as:

$$\phi(t) = \begin{cases} 1, & 0 \le t < 1 \\ 0, & \text{otherwise} \end{cases}$$

4.4 DDWT analysis on Blur Model

Applying Equation (1) on Equation (2), we find that DDWT are made of the convolution result of two wavelet filters, blur kernel, and latent image. We organize one wavelet filter d^j and latent image x in a group, and another wavelet filter d^i and blur kernel *h* in a group:

$$\begin{aligned} v^{ij}(n) &= \{d^i * d^j * y\}(n) \\ &= \{d^i * h\} * \{d^j * x\}(n) + \{\{d^i * d^j * \varepsilon\}(n) \\ &= \{q^i * u^j\}(n) + \eta^{ij}(n) \end{aligned}$$

About motion kernel *h*, for a moving object with constant velocity and angle ψ, we can assume the motion kernel *h* as:

$$h(x,y)=\begin{cases}1/k & 0\le|x|\le k\cos\varphi,\ y=k\sin\varphi\\ 0 & \text{else}\end{cases}$$

where k is the length of blur kernel; and φ is the moving angle.

That means, if we assume the motion has a constant velocity k and moves horizontally, the blur kernel can be modeled as:

$$h(n)=\{\frac{1}{k},...,\frac{1}{k}\}$$

Using Haar Wavelet [1,-1] for d^i as the basis. The result of convolution with d^i and h can be considered as the difference of two impulse functions.

$$q(n)=\{\frac{-1}{k},0,...,0,\frac{1}{k}\}=\frac{\delta(n+\frac{k}{2})-\delta(n-\frac{k}{2})}{k}$$

About v^{ij}, for a filter supported with finite length, $n\in\{n_1, n_2,\ldots, n_k\}$, we can transform v^{ij} from the convolution term to the summation term, then we have:

$$v^{ij}(n)=\sum_{k=1}^{K}q^i(n_k)u^j(n-n_k).$$

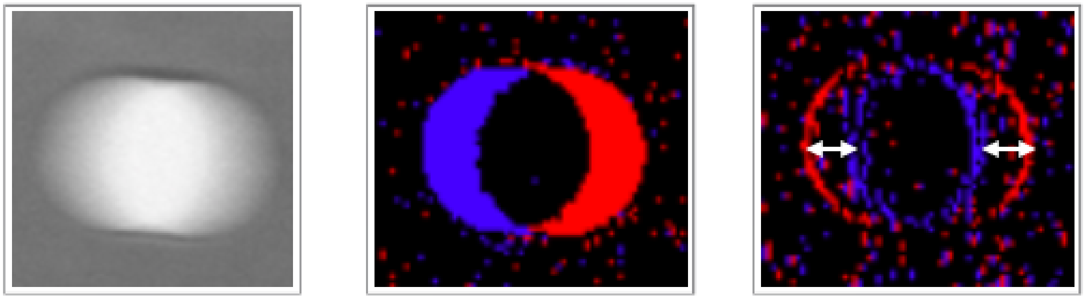
Because q^i is considered as the difference of two impulse functions, sparsity can infer $v=\{q*u\}$ to be a difference of two u^j coefficients placed k apart.

$$v(n)=\{q*u\}(n)=\frac{u(n+\frac{k}{2})-u(n-\frac{k}{2})}{k}$$

where v is DDWT; u is DWT of latent image; q is the convolution result of h and d^i; k is the blur kernel length; and n is the position in wavelet domain.

Suppose the influence of noise is very minor, and sparsity can separate u coefficients efficiently. We deduce that each v coefficient is generated from u coefficient which is apart from k/2 or -k/2 away. That means, without any overlapping, every u coefficient is exactly k times larger than v coefficient, which is apart from k/2 or -k/2 away. Besides, the relation between v(n+ k/2) and v(n- k/2) is that both of them have the same scalar with different sign bits. We call double edge for this property.

It is clearer in real signals illustration than text explanation. If we apply a practical blurry image into double discrete wavelet transform. Then we set a constant threshold σ to divide signals into three categories: blue for those are smaller than -σ, black for those are between -σ and σ, and red for those are larger than σ. We observe two continuous signals with different sign bits in DWT, Figure 3.b, and double edges appear DDWT in Figure 3.c, respectively.



(a) Original image. (b) DWT w^i. (c) DDWT v^{ij}.

Figure 3. Applying DDWT to the practical signals.

Since we can observe the phenomenon of double edge in v if the signal was contaminated by motion blur, the deblur method might seem intuitive at this step. Our goal is estimating the DWT coefficients of latent image u from DDWT coefficients v. In Figure 4.3, it is the representation of the relation of those coefficients. That means, if we estimate the correct u from v, we can recover the latent image successfully.

4.5 Autocorrelation and Hill Climbing Searching Algorithm

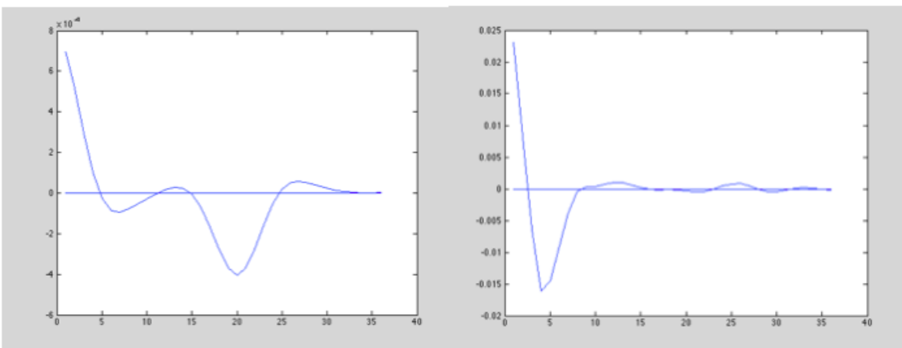
Based on above definition, double edge is the same magnitude with different signs revealed in k pixels. Suppose double edges appear in DDWT coefficients, we consider the distance k as the length of blur kernel. It is obvious two numbers with the same value but different sign bits will get the smallest result by multiplying together. That means, suppose r>1, even we have no idea about the exact value of r and –r, it is guaranteed that r*(-r) is the smallest result since the negative sign remains. Using this property, we assume the position proper k has the smallest minimum result by summation of every pair of multiplications. We are able to use this simple mathematical relationship on double edge analysis to find proper length k by enumerating all possible candidates and recording the summation of every result.

In this sub-problem, Zhang proposed autocorrelation to determine the final k [14].

Autocorrelation is useful analysis tool for finding repeating patterns by calculating the cross-correlation of a signal with itself. It is the similarity between observations as a function of the time lag between them. In our case, the goal is finding the correct distance between double edges. We are able to consider the double edges as the repeating patterns with different sign bits. After that, we can estimate the proper k by getting the result from the position of minimum result in autocorrelation. In Figure 4, two autocorrelation results from different scales of DDWT signals.

$$R(\tau)=\frac{E[(X_t-\mu)(X_{t+\tau}-\mu)]}{\sigma^2}$$

where E is expected value operator; X is the position of signal; t is the position in the spatial domain; τ is the time lag; μ is mean value; and σ^2 is standard deviation.



(a) Autocorrelation k=20. (b) Autocorrelation k=5.
Figure 4. Autocorrelation on different blur scales of image.

However, it is time-consuming to compute all possible candidates to decide proper k. Although global searching can always get the specific answer, its corresponding execution time is also very long. In this work, we use hill climbing searching algorithm instead of global searching. Hill climbing is a mathematical local search optimization. It attempts to find a better solution by continuously changing a single element of the solution. If the change produces a better solution, an incremental change is made to the new solution, repeating until no further improvements can be found. To avoid local minimun, we set some checking mechanisms to decide whether calculating continuously or break the search.

Based on the observation, we get two conclusion: 1. global minimum is same as local minimum in small scales(k < 14), 2. Global minimum always appears behind positive candidates. We use above conclusions to set three following mechanisms to avoid traping into local minimum and deciding calculating going or not.

Based on the above conclusion, we set three detecting mechanisms to find the global minimum with the prior knowledge:

1.A minimum candatate must appear behind some positive candidates.
2. A minimum candatate must following with decreasing trend.
3. A minimum candatate must appear in front of increasing trend.

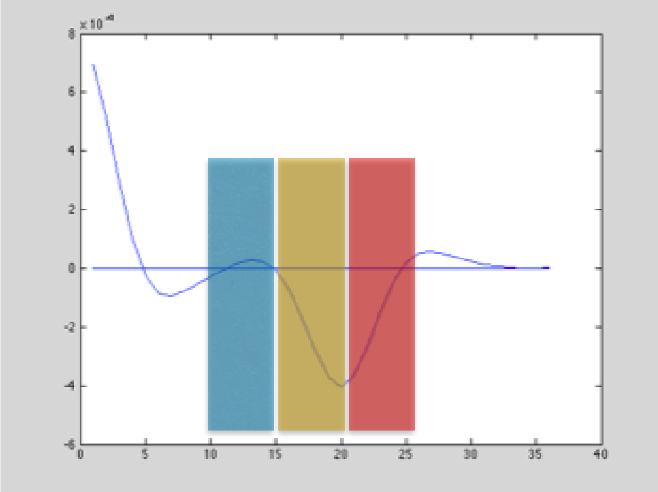


Figure 5. Presentation of detecting mechanisms. Blue for mechanism 1, yellow for mechanism 2, and red for mechanism 3.

Pseudo Code

```
bool positive=false;
bool decreasing=false;
bool increasing=false;
for(k=1; k < s*2; k++)
{
  record[k]= autocorr(k);
  if(record[k]<minimum) minimum=record[k],answer=k;
  if(record[k]>0) positive=true;
}
```

```
if(positive)
{
    while(k<range && !increasing)
    {
    record[k]= autocorr(k);
    if(record[k]<minimum) minimum=record[k],answer=k;
    If(record[k]>0) positive=true;
    if(positive && record[k-1] > record[k]) decreasing=true;
    if(decreasing && record[k-1]< record[k]) increasing=true;
    }
}
```

4.6 Deducing latent coefficients

After estimating a proper k, the final step of deblur method is correct the blurry coefficients u. In Section 4.4, we define the relation of v and u. Suppose the coefficients v are separated well due to sparsity, Zhang proposes a deducing strategy to decide latent u coefficients [14]. Since nonzero v only appears in the situation of motion blur. We can deduce well u, since it will get the same value whether choosing from $v(n-\frac{k}{2})(-k)$ or from $v(n+\frac{k}{2})k$.

$$u(n)=\begin{cases} v(n+\frac{k}{2})k & \text{if} \left\| v(n-\frac{k}{2}) \right\| > \left\| v(n+\frac{k}{2}) \right\| \\ -v(n-\frac{k}{2})k & \text{else} \end{cases}$$

5. Experiments

5.1 Equipments

Matlab 2013a is used at the simulating and verifying state. After determining algorithm and parameter, we implement it into C++ code language with OpenCV 2.4.7, and use Basler Pylon camera capturing images and simulating the industry inspection lines. The testers are 3M strips.

(a)Balser Pylon camera. (b) 3M strips.
Figure 6. Basler Pylon camera and 3M strips.

5.2 Comparison of estimation

Tables 1,2,3,4 are the comparison of Zhang's and our result. The input is generated by a known motion blur kernel, plus random noise. Zhang uses the global searching algorithm to find the best kernel length. On the contrary, our method only moves with optimal strategy. Therefore, the searching times and execution time in our method are fewer than Zhang's method and also achieve the same result.

Table 1. Comparison of searching times from k=11~20.

Times\Kernel	11	12	13	14	15	16	17	18	19	20
Zhang's method	42	42	42	42	42	42	42	42	42	42
Our method	26	25	23	33	22	23	25	26	27	28

Table 2. Comparison of searching times from k=21~30.

Times\Kernel	21	22	23	24	25	26	27	28	29	30
Zhang's method	42	42	42	42	42	42	42	42	42	42
Our method	16	27	27	29	32	33	22	23	24	25

Table 3. Comparison of execution times on estimation from k=11~20.

Seconds\Kernel	11	12	13	14	15	16	17	18	19	20
Zhang's method	0.069	0.070	0.066	0.065	0.067	0.066	0.065	0.066	0.065	0.065
Our method	0.036	0.034	0.032	0.045	0.030	0.032	0.034	0.036	0.037	0.039

Table 4. Comparison of execution times on estimation from =21~30.

Times\Kernel	21	22	23	24	25	26	27	28	29	30
Zhang's method	0.066	0.065	0.068	0.069	0.066	0.069	0.066	0.066	0.065	0.069
Our method	0.022	0.037	0.037	0.042	0.052	0.045	0.030	0.031	0.033	0.034

5.3 Deblurring with real data

In this section, we show some real data captured by the industrial camera, Basler Pylon. The detection object is a 3M strips with unknown velocity.

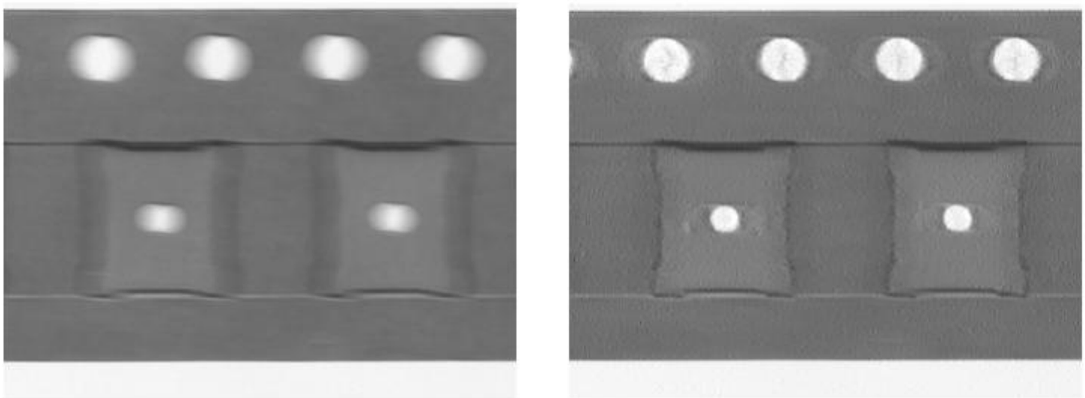


Figure 7. Blurry image and deblur result, blur length = 20 pixels.

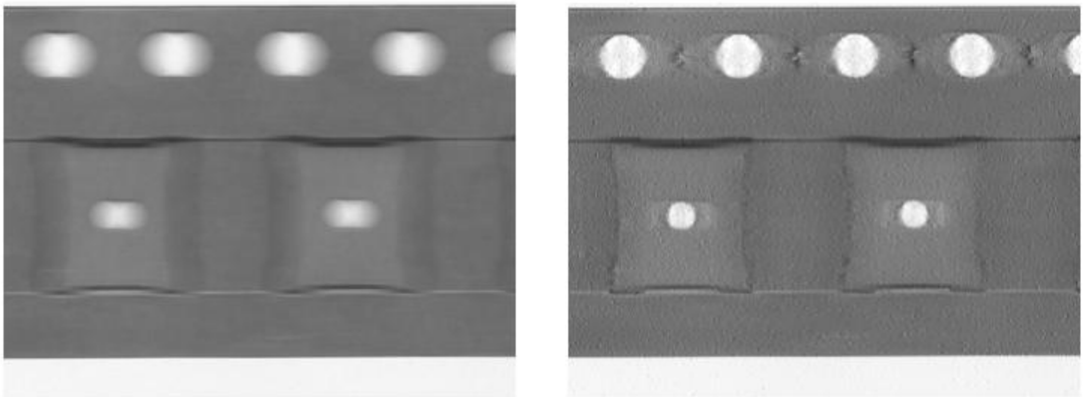


Figure 8. Blurry image and deblur result, blur length = 28 pixels.

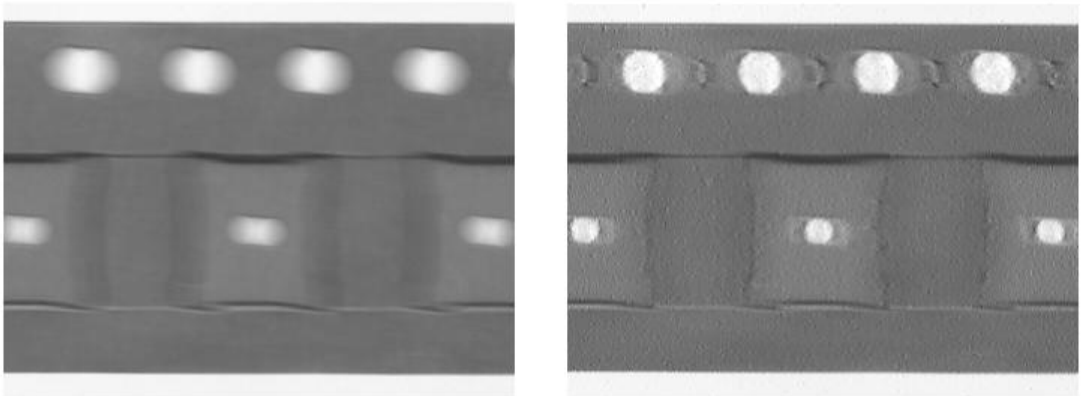


Figure 9. Blurry image and deblur result, blur length = 29 pixels.

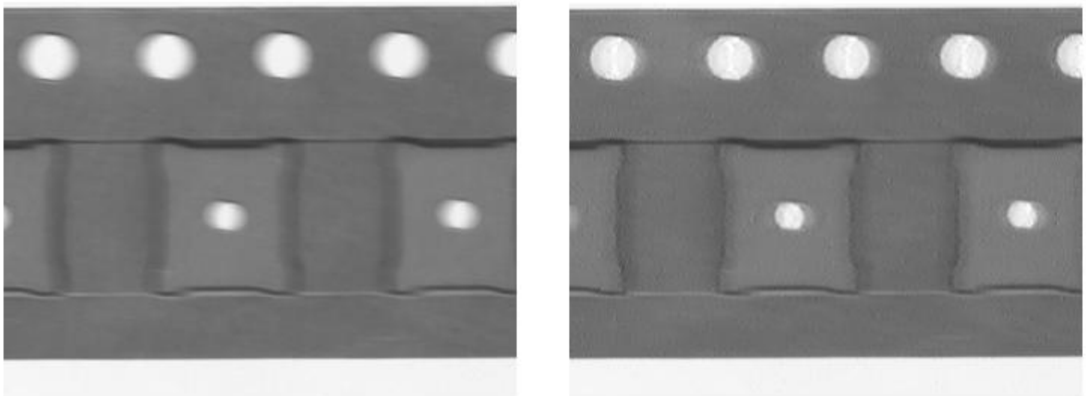


Figure 10. Blurry image and deblur result, blur length = 15 pixels.

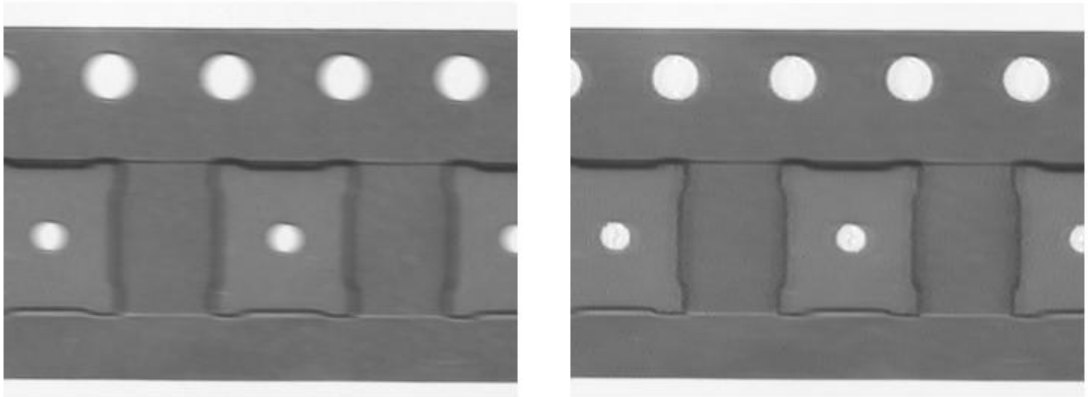


Figure 11. Blurry image and deblur result, blur length = 10 pixels.

6. Conclusion and Future Work

Nowadays, blur is still a challenging problem due to the unknown blur kernel estimation and the noise contamination. In our work, we propose a real-time method in wavelet domain to estimate the motion blur kernel on autocorrelation and hill climbing searching algorithm. With the blur kernel information, we are able to correct the proper coefficients and to remove motion blur for blurry image. Our contribution in this work is we improve the searching algorithm instead of global searching. Although global searching can always find the best answer, it spends much time enumerating all candidates. We propose using hill climbing searching algorithm to find the position of global minimum but also avoiding the local minimum instead of global searching. The method is fast since it only needs to visit fewer candidates and also have high accuracy. Compared with previous method, the improvement can reduce more than half of estimated time and get the same result.

Deblurring in wavelet domain is a novel framework with excellent result. This analysis technique is useful in many image processing and signal processing fields, for instance, the detection of the road speeding camera, the inspection of industry production system, the moving athletes in online broadcast tournament, and some medical online detection system. Our result might have not reached the quality of some specific measurement in medical field now, after improving the quality with the fast execution time, no doubt this deblurring method will be involved in more fields. Let us leave this research to the future work.

7. Acknowledgement

This research was supported by the Ministry of Science and Technology of Taiwan, R.O.C., under Grants NSC 101-2221-E-002 -194 and NSC 102-2221-E-002 -177, and by Winstar Technology, Test Research, Qisda, Lumens Digital Optics, PolarLink, and Lite-on.

8. Reference

[1] T. S. Cho, S. Paris, B. K. P. Horn, and W. T. Freeman, "Blur Kernel Estimation Using the Radon Transform," Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Colorado Springs, Colorado, pp. 241-248, 2011.

[2] D. L. Donoho and M. E. Raimondo, "A Fast Wavelet Algorithm for Image Deblurring," ANZIAM Journal, Vol. 46, pp. C29-C46, 2004.

[3] R.Fergus,"Computational Photography," http://cs.nyu.edu/~fergus/teaching/comp_photo/, 2013.

[4] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, "Removing Camera Shake from a Single Photograph," Proceedings of ACM SIGGRAPH, New York, Vol. 25, Issue 3, pp. 787-794, 2006.

[5] J. Lu, E. Poon, and K. N. Plataniotis, "Restoration of Motion Blurred Images," Proceedings of ACM Symposium on Applied Computing, Dijon, France, pp. 301-305, 2006.

[6] D. Miller and W. Scott, "Deconvolution with Inverse and Wiener Filters," http://cnx.org/content/m13144/latest/?collection=col10380/latest, 2014.

[7] Y. Zhang and K. Hirakawa, "Blur Processing Using Double Discrete Wavelet Transform," Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Portland, Oregon, pp. 1091-1098, 2013.