Notes on Photoshop’s Defect in Simulation of Global Motion-Blurring

Li-Dong Cai
Department of Computer Science, Jinan University
Guangzhou 510632, CHINA
ldcai@21cn.com

ABSTRACT
In restoration of global motion-blurred images, simulated blur images play an important role in the test of algorithms and assessment of quality. Photoshop is a popular tool to generate such simulated images. However, the motion-blur images produced directly by Photoshop may sometimes fail in image restoration. An analysis, with reference to the physical process of image formation, is thus made to find the root of the defect and to explain why periodic waves always appear within failed restorations. A pre-processing is then proposed to solve this problem by zero-padding the marginal scene areas in a width proportion to the motion distances. Experimental results show that by this pre-processing the defect of loss of motion information is precluded from the output simulated images, thus being qualified to use in motion-deblurring.

Keywords
motion-blur, Photoshop, simulation, defect, zero-padding.

1. INTRODUCTION
Motion may lead to an image blur during its formation. It is mainly caused by the relative motion between the camera and the scene, making the instantaneous exposure overlapped on a film over the time interval during which the shutter is open. And the image blur caused by camera’s shifting is called global motion blur, for which there are many methods to do the restoration job [Gon, Son, Lim].

Usually, the original version of a scene image is unknown to restoration. This brings inconvenience to the testing of restoration algorithms as well as the quality assessment of restored images. Simulation of blur images is thus needed to produce the common test data where original images are always known.

To do so, the software which produces simulated motion-blur images should satisfy four requirements as follows: 1) be universal and popular; 2) can treat various images; 3) only the motion parameters (direction and distance) are needed to generate a blur image; 4) be independent to image de-blurring methods.

Among numerous software of image processing, Photoshop is a widely used one satisfying the above requirements. It is a convenient and parameterized toolkit in simulation of motion-blur images; and as a source code encapsulated business software, its image blurring algorithm is a black box for all users, hence being fair to every restoration algorithm.

2. DEFECT IN PHOTOSHOP’S RESULT
According to Photoshop’s user guide, only two motion parameters, the direction \( \theta \) and distance \( D \) are needed to produce simulated motion-blur images. These images look pretty well for the purpose of watching, but will sometimes produce puzzling results when they are taken as the input for restoration, and things get apparent when the restoration is made using the Traveling Wave Deblurring (TWD) method [Cai]. As shown in Figures 1 to 3, these results can be roughly classified into three different cases:

Case 1: For scene images consists of a non-zero background and a central object (see Figure 1a), periodic waves appear in the restored version of the simulated blur images (see Figure 1c).
These inconsistent results give people an impression that image restoration’s success and failure could be content-dependent. For instance, Case 1 and 2 together suggest that the restored results are related to non-singleness of the background; Case 2 and 3 together suggest that the restored results are related to the composition of marginal areas in images; and Case 1, 2 and 3 altogether suggest that the restored results are related to the non-single composition of marginal areas in images. However, note that in the case of global motion, what an image undergoes is a rigid movement, therefore all the pixels have the equal status. So, the above suggestions are plausible, the real reason for appearing these inconsistent restored results is not due to the differences between scene images, but due to possible defects in the process of producing a simulated image.

3. ANALYSIS OF PHOTOSHOP’S DEECT IN SIMULATION

For simplicity, 1-dimensional case is discussed. Suppose that a scene image \( f(x) \), viewed through a window, is of an even intensity distribution, \( L \) in length, and with a horizontal moving rate \( v \) towards right; its images \( f(x - vt) \) at different instances \( t \) during the exposure time interval \([0, T]\) overlap together and form the motion-blurred image \( \tilde{g}(x) \):

\[
\tilde{g}(x) = \int_0^T f(x - vt) dt
\]

(1)

with an intensity distribution as shown in Figure 4 below:
Let the motion distance $D = vT$, then the length of the blurred image will be $L + D$, i.e., the global motion leads to the blurred image $\tilde{g}(x)$ being longer than the original scene image $f(x)$. However, the simulated (blurred) image $\hat{g}(x)$ output by Photoshop maintains the same size $L$. So, Photoshop must have made a re-location and/or a cutting to the blurred image according to the motion distance $D$. As the source code of Photoshop is encapsulated, the internal forms of re-location and cutting are unknown to users, so judgment of them has to be made only based on their external behavior in experiment.

For example, draw a one-pixel thick, vertical line at the center of a blank background as the scene image, then make it motion-blurred with different shifting values, Photoshop’s behavior in re-location and cutting can be observed by a comparison of a specific object’s position in the scene image and the simulated image.

As shown in Figure 5, for different shifting values, each blurred line maintains its center position invariant. This means that when the distance of a forward motion is $D = vT$, Photoshop will shift the blurred image $\tilde{g}(x)$ 0.5D backward so as to ensure the blurred image’s center maintains at the window’s center. So the blurred image is now 0.5D excess beyond the window at both sides, and they will be cut out before the image is output as simulated image. Therefore, motion information within both image margins is thus possible to be lost, leading to failure of image restoration.

With this observation, a reasonable explanation to Cases 1, 2 and 3 is now available by appealing to the Traveling Wave Deblurring, where the scene image satisfies the so-called recursion equation [Cai] below:

$$f(x) = v \frac{\partial \hat{g}}{\partial x} + f(x - D)$$

(2)

with its different forms over sub-intervals of $[0, L + D]$:

$$f(x) = v \frac{\partial \hat{g}}{\partial x} + f(x - D) \quad 0 \leq x < D$$

(3)

$$f(x) = v \frac{\partial \hat{g}}{\partial x} + f(x - D) \quad D \leq x < L$$

(4)

$$f(x) = 0 \quad else$$

(5)

Note that in order to output a simulated blurred image $\hat{g}(x)$, Photoshop needs to re-locate the blurred image $\tilde{g}(x)$, cut out both sides 0.5 D each. This is actually equivalent to zero-padding the blurred image there

$$\tilde{g}(x) = 0, \quad 0 \leq x < 0.5D \cup L + 0.5D \leq x \leq L + D$$

(6)

So, the restoration to the simulated image $\hat{g}(x)$ is in fact a computation to the zero-padded, blurred image. From Eq. (3), (5), there must be:

$$f(x) = v \frac{\partial \hat{g}}{\partial x} = 0, \quad 0 \leq x < 0.5D \cup L - 0.5D \leq x \leq L$$

(7)

Therefore, different results will appear in Cases 1, 2 and 3 above:

1) For any motion-blurred image $\tilde{g}(x)$ which results from a scene image composed of a complex
background and a central object as shown in Figure 1a, there will be \( \tilde{g}(x) \neq \text{const} \) in both 0.5D width margins, hence \( f(x) \neq 0 \). But this is in contradiction with Photoshop’s zero-padding result, \( f(x) = 0 \) as presented by Eq. (8), and will bring forwards errors into consequent recursions, thus leading to failure of image restoration.

2) For any motion-blurred image \( \tilde{g}(x) \) which results from a scene image composed of a unitary (zero) background and a central object as shown in Figure 2a, because the overlapping of zero background creates nothing but zero background in both 0.5D width margins. It is true from the beginning, that \( \tilde{g}(x) = 0 \) hence \( f(x) = 0 \). Therefore, Photoshop’s zero-padding brings no effect to producing a proper simulated image as long as the width of zero background is not less than 0.5D at both margins of the motion-blurred image, and a successful restoration can be expected.

3) For any motion-blurred image \( \tilde{g}(x) \) which results from a scene image composed of a unitary (zero) background and a marginal object as shown in Figure 3a, because the width of zero background at both margins is less than 0.5D, the situation is similar to Case 1, \( i.e. \), there exists \( \tilde{g}(x) \neq \text{cons} \), thus \( f(x) \neq 0 \), and image restoration will fail. However, image restoration can succeed as long as the width of zero background at both margins is adjusted to 0.5D as shown in Figure 6.

As for the phenomenon of periodic waves always appearing in the failed restoration, it is not difficult to make a similar and reasonable explanation. Let both cut-out margins of the blurred image \( \tilde{g}(x) \) be \( \delta(x) > 0 \), \( x \in [0,0.5D) \cup (L+0.5D,L+D) \), then extend it to \( \delta(x) = 0 \), \( x \in [0.5D,L+0.5D] \) so that the simulated image \( \tilde{g}(x) = \tilde{g}(x) - \delta(x) \), \( x \in [0,L+D] \). Therefore, from Eq. (3), the restored result of \( \tilde{g}(x) \) over \( 0 \leq x < D \) can be seen as that of the sum of \( \tilde{g}(x) \) and \( -\delta(x) \):

\[
\begin{align*}
  f(x) &= \frac{\partial \tilde{g}}{\partial x} - \frac{\partial \delta}{\partial x} \quad 0 \leq x < 0.5D \\
  f(x) &= \frac{\partial \tilde{g}}{\partial x} \quad 0.5D \leq x < D 
\end{align*}
\]

(3’)

Afterwards, according to Eq. (4), the influence of \(-\delta(x)\) will be introduced into \( f(x) \) over \( [D,L] \) with a period of \( D \), making the cut-out left margin present in a white and black reversed waving texture till \( x = L \) where the processing of whole simulated image finishes.

4. DEFECT PRECLUSION BY ZERO-PADDING

The discussion so far shows that simulated motion-blurred images resulting directly from Photoshop are not always qualified for testing of algorithms. Restoration of these images sometimes fails for the reason that while motion-blurring makes an image be larger, Photoshop’s re-location and cutting treatment are used to retain the image size, and this puts the image at the risk of losing the motion information in marginal areas for image restoration.

To deal with this situation, the simplest way is to directly take the motion-blurred image as the simulated image without any cutting. This means that the simulated image no longer keeps the same size to that of the input image, and will change size from time to time corresponding to the motion distance. Therefore, it is not only troublesome to do but also no way to do, since this will involve the revision of Photoshop’s source code though users have no access to those encapsulated code, let alone to change it.

Note that the simulated image \( \tilde{g}(x) \) is a 0.5D margin-cut version of the blurred image \( \tilde{g}(x) \), and according to Eq. (8), such a cut is equivalent to a 0.5D zero-padding to both margins of the original scene image \( f(x) \). If both image margins of \( f(x) \) are set to zero in 0.5D width from the beginning, then there will be actually no loss of motion information in the

(a) left margin is 10 pixels (b) blurred (\( D = 20 \) pixels)

(c) restored image

Figure 6. So long as the left margin expands to 0.5D=10 pixels in width, deblurring of simulated image succeeds.
simulated image, thus being qualified to be used in image restoration.

Therefore, given the simulated motion distance D, an effective pre-processing which is consistent with Photoshop’s re-location and zero-padding, is first to make a 0.5D zero-padding at both margins of the scene image, then use Photoshop to do motion blurring so as to produce a simulated image in the same size.

An example is shown in Figure 7aa, where a zero-padding pre-processed scene image is in content a zero-padding version of the original scene image. When it is used as the input to Photoshop’s simulation for motion-blurring, there is no loss of motion information in the output image, thus being a qualified simulated image as desired in both appearance and application.

(a) original scene image  (aa) zero-padded version  
(b) motion-blurred (20 pixels)  (c) restored

Figure 7: With a 0.5D zero-padding at both margins as the pre-processing for Photoshop’s treatment, a qualified simulated version of motion-blurred image is properly produced.

5. SUMMARY
Motion blur images resulting directly from Photoshop’s simulation may sometimes fail in image restorations, thus being not qualified to be used for testing of algorithms or performance assessment. This paper first exposes that this defect has its root in information loss caused by Photoshop’s re-location and cutting of the motion-blurred image; then explains why periodic waves always appear when image restoration fails; and proposes to apply a certain width zero-padding to both margins of a scene image as a pre-processing of Photoshop’s treatment. Experimental results show that by this pre-processing; motion blur images generated by Photoshop’s simulation are now qualified to be used in motion-deblurring.

6. REFERENCES


