The Nature of Motion Blur

Abstract

The removal of motion blur induced into images is currently an active field of research. The analysis of motion de-blur algorithms has shown, that they perform differently on synthetic and real motion blur. Therefore we have build an apperatus which allows us to produce pictures with defined motion blur induced, so we can study these possible differences. In this paper we focus on clearly visible lines in the spectrum of synthetically blurred images. We investigate for such lines in real blur images and are able to show their existence with both blur types. Furthermore we give theoretical reasoning for these lines to appear. Hence this which is a strong indice that the basic blur model used to generate synthetic blur incoperates some truth.

1 Purpose

Motion Blur is a phenomena encountered in many application. Not only photographers, but also astronomers and operators of surveillance cameras are struggling with distorted images due to motion blur [5]. Several techniques to restore the image already exists, but none of the existing algorithms allows perfect restoration [5]. The suboptimal results are partly due to differences between the motion blur modeled and the motion blur encountered in real world images. Revealing the similarities and differences hopefully leads to a better understanding of the motion blur phenomena, thus allows to design better algorithms. Therefore a certain aspect of synthetically blurred image is considered here, namely the appearance of of lines in the spectrum of the blurred image.

2 Experimental Observations

Let's take a look at a picture (Figure 1) which got blurred (synthetically with a known horizontal displacement, in this case 10 pixels).

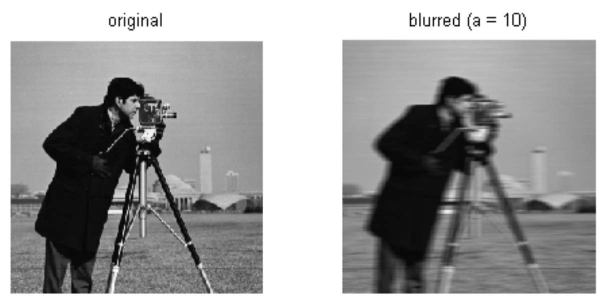

Figure 1:
Original and synthetically blurred image

Now the spectrum of both images is generated via a FFT and both spectra are compared (Figure 2). The upper row represents the spectrum of the unblurred, original image whereas the lower row show the spectrum of the blurred image.

The upper spectra look like random noise and there seems to be no additional information included in the picture. Whereas in the spectra of the blurred image clearly some lines get visible. These lines are visible at best in the magnitude spectrum, but also can be seen in the real or imaginary only spectrum. If we compute the average value of a vertical line and plot the result (Figure 3), we get a graph where the peaks represent the vertical lines in the magnitude spectrum.

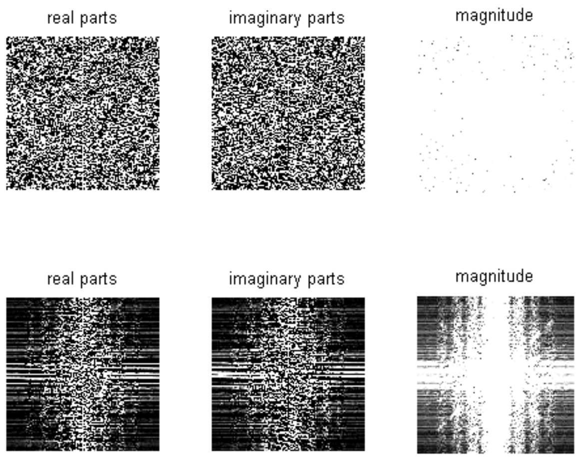

Figure 2:
Spectra of blurred and non-blurred image

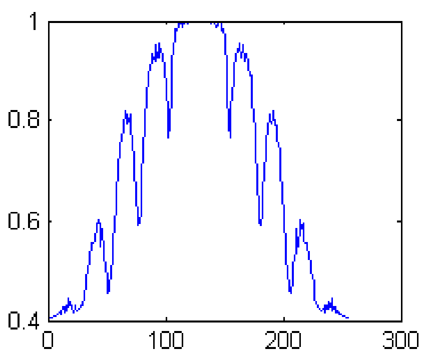

Figure 3:
Average magnitude along the horizontal axis of blurred image

Counting the number of peaks, we find 9 peaks, which is about the value of the blur parameter used to generate this image. If one now considers that the peak in the middle is about twice as wide as all the other peaks, we can count it as a double peak and therefore get 10 peaks, which is correspond to the blur parameter used earlier. We now assume as a thesis that the number peaks in the intensity graph of the spectrum of an blurred image corresponds to blur parameter used during its generation.

Let's see if this holds true for other blur parameters. Figure 4 show the intensity graph of pictures blurred using different blur parameters. Counting the number of peaks, we get values which are round

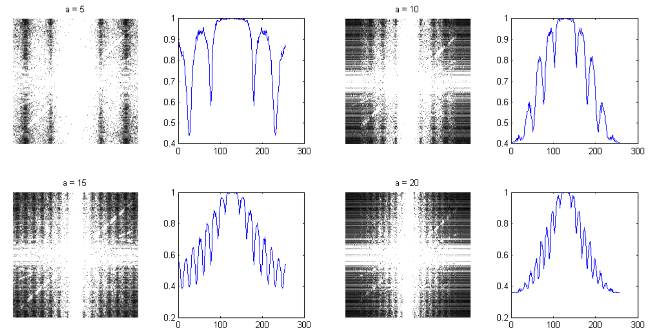

Figure 4:
Spectra of blurred images with different displacements

about the blur parameter, but differ to a little extend. This is possible due to the little amplitude near the ends.

Another important issue is, before questioning why these lines occur is, if our findings are not only of theoretical nature but are also visible in blurred pictures taken with a camera. For this purpose we have build an experimental setup, which allows to produce blurred images with a defined displacement. Furthermore the blurred images should be comparable to the synthetically created ones in terms of linear blur and horizontal-only blur.

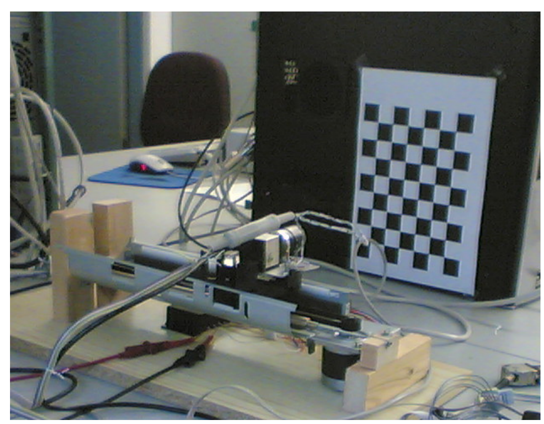

Figure 5:
Experimental setup

This unit is shown in Figure 5 and comprises a camera unit, a guiding rail and a stepper motor. The camera carriage is accelerated to a constant speed and the camera takes a photo with a medium exposure time (around 100ms) to allow significant motion blur appear in the picture. As a motif a checkerboard structure has been chosen, since this allows an easy method for estimating the associated blur parameters.

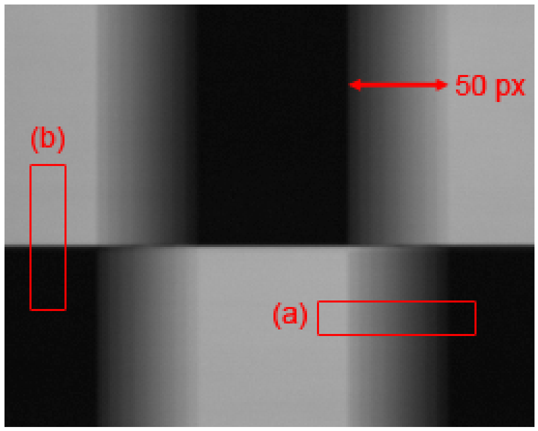

Figure 6:
Estimating the PSF with real motion blur

An image captured with our experimental setup is shown in Figure 6. From the way we have set up the capturing process we assume the motion path of the camera to be linear and uniform in the horizontal direction. The first assumption is verified by looking at the plot shown in Figure 7. The plot depicts the luminance of the picture, which taken across the blurred zone (marked with (a) in Figure 6). The luminance curve is very close to linearly decreasing.

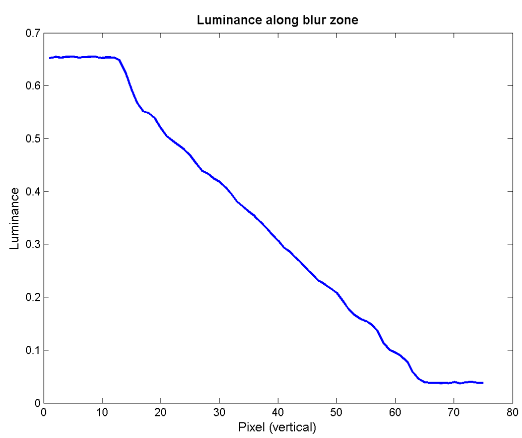

Figure 7:
Verification of linearity

The second assumption that the motion is only in the horizontal direction is verified with a second luminance graph (of region (b)), shown in Figure 8. The sharp decay of luminance at the border between the two boxes proves that there is almost no motion in the vertical direction (otherwise the graph must look like shown in Figure 7, where the decay is linear and spread about a significant number of pixels).

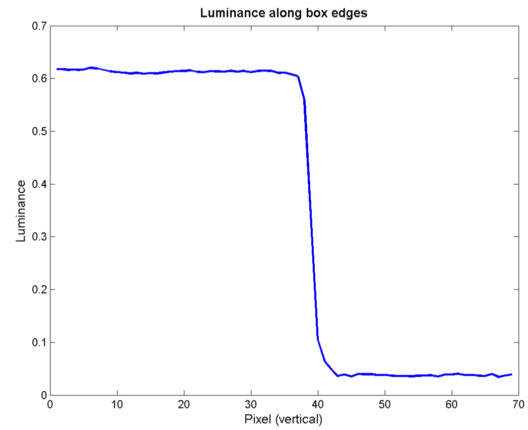

Figure 8: Verification of horizontal motion

Measuring the blur area in Figure 7 allows us to directly infer the length of the PSF (the length has been visualized in Figure 6).

Another picture taken by this unit is shown in Figure 9, which is used to see if your assumption of the characteristic spectrum holds true.

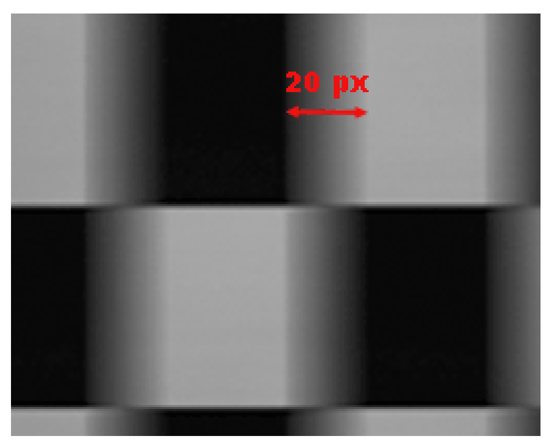

Figure 9:
Real blur of a checker board structure

Using the same technique as for the synthetic image, we can compute the intensity graph (Figure 10). Clearly the graph shows much more noise than the one of the synthetic blurred images. In this case we calculate the period of the peaks, since it is constant in the graphs above and we need only a few peaks to calculate the total number. The marked data points show a period between 5 and 6. This gives us 21 to 23 peaks which is about the same as the blur parameter determined earlier.

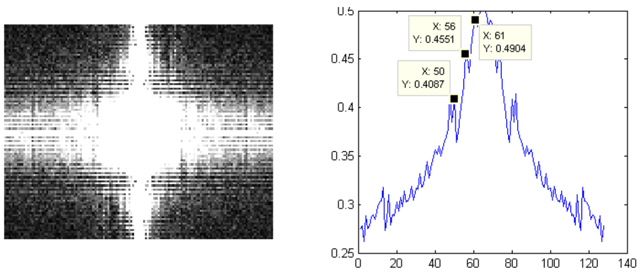

Figure 10:
Spectrum of checker board structure

3 Theoretical Reasoning

Commonly a linear, non-recursive (FIR) is used to model the degradation of digital (sampled) images caused by motion blur. Let's consider the original, blur-free $M \times N$-image $\boldsymbol{f}$ to be convolved with a convolution kernel $\boldsymbol{h}$, referred to as the Point Spread Function (PSF). Additionally, some noise is introduced during the capturing process, which is modeled with the additive noise term $\boldsymbol{n}$. Hence, the blurred $M \times N$ image $\boldsymbol{b}$, as it is captured by the moving camera, is modeled as

$$\boldsymbol{b} = \boldsymbol{h} \star \boldsymbol{f} + \boldsymbol{n} \tag{1}$$

where the symbol $\star$ represents the convolution operator.

The most simple form of the PSF consists of two Heaviside functions, defining a rectangular filter:

$$\boldsymbol{h} = u(0) - u(a)$$

with a being the blur parameter or respectivly the length of the blur zone.

If we consider discrete values, the PSF consists of a series of Diracs:

$$\boldsymbol{h} = \sum_{k=0}^{a} \delta(k)$$

It is furthermore known, that a convolution with a Dirac is a shifts by the amount the Dirac's argument of the original function. Assuming no noise ($\boldsymbol{n} = 0$), the blurred image can be written as

$$\boldsymbol{b} = \boldsymbol{h} \star \boldsymbol{f} = \sum_{k=0}^{a} \delta(k) \star \boldsymbol{f} =$$

$$\boldsymbol{f}(x,y) + \boldsymbol{f}(x+1,y) + ... + \boldsymbol{f}(x+a,y) = \sum_{k=0}^{a} \boldsymbol{f}(x+k,y)$$

Clearly the resulting image increased in intensity, which has to be compensated by a factor $c = \frac{1}{a+1}$, which has been omitted here fore simplification purposes.

As we are interested in the spectrum of the blurred image, we will use the Fourier transformation (FT) $\mathcal{F}\{\}$ to achieve this. Since the FT is a linear operation, we can move the FT-operation into the sum.

$$\mathcal{F}\{\boldsymbol{b}\} = \mathcal{F}\{\sum_{k=0}^{a} \boldsymbol{f}(x+k,y)\} \overset{linearity}{=} \sum_{k=0}^{a} \mathcal{F}\{\boldsymbol{f}(x+k,y)\}$$

Now let's look at a single, displaced image. The rule of circular displacement allows us, to rewrite the expression in that manner, that it only contains the original image.

$$\mathcal{F}\{\boldsymbol{f}(x+k,y)\} = e^{-j\frac{2\pi k}{N}} \cdot \mathcal{F}\{\boldsymbol{f}(x,y)\}$$

where N denotes the total Length of the image. Now we make again use of linearity and factor out the spectrum of the original image:

$$\mathcal{F}\{\boldsymbol{b}\} = \sum_{k=0}^{a} e^{-j\frac{2\pi k}{N}} \cdot \mathcal{F}\{\boldsymbol{f}(x,y)\}$$

Assuming that the spectrum of the motion blur free image consists only of random noise, we can neglect its influence on the blurred image spectrum. Plotting the sum of exponential function yields a graph (Figure 11) which is somewhat similar in its peak structure to the one of the real blur image. The plot has been shifted by half its width to correspond to MATLAB's way of computing the spectrum. Furthermore this graph has been translated into an image, with the characteristic line structure. Here it is harder to see the corresponding lines in the real blur spectrum, but they can be seen even though they are dominated by non-equally distributed noise. We can infere, since lines are present in both synthetic and real spectrum, convolution is a valid method to model motion blur.

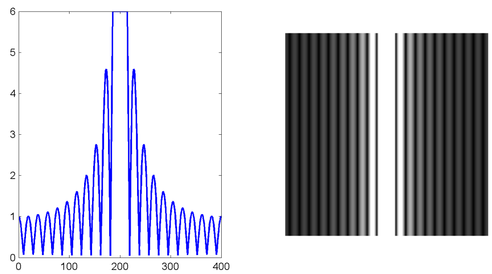

Figure 11:
Series of exponential functions

4 Conclusion

We have seen that synthetic and real blur seems to be quite similar in terms of lines in the spectra. Thus this can not be a reason for restoration algorithms to fail miserably on real world pictures. Taking a look at the spectrum of the real blur image gives a indication that noise might be the reason. In comparison to the synthetic spectrum it comprises more noise. Furthermore comparison tests have shown increasing restoration problems when noise is added to the synthetic blur [5]. Hence more research into the noise issue is required.

Furthermore one might be tempted to use the lines appearing in the spectrum to determine the blur parameters [3]. Quick tests on real pictures taken by cameras with long exposure time reveal that the relevant motion is much more complex and can therefore not be identified by just looking at the lines in the spectra. But only recently new findings [1, 2, 4] offer promising ways to acquire the blur parameters.

References

[1] M. Ben-Ezra and SK Nayar. Motion deblurring using hybrid imaging. *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, 1, 2003.

[2] M. Ben-Ezra and S.K. Nayar. Motion-Based Motion Deblurring. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):689–699, 2004.

[3] MM Chang, AM Tekalp, and AT Erdem. Blur identification using the bispectrum. *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, 39(10):2323–2325, 1991.

[4] R. Fergus, B. Singh, A. Hertzmann, S.T. Roweis, and W.T. Freeman. Removing camera shake from a single photograph. *ACM Transactions on Graphics (TOG)*, 25(3):787–794, 2006.

[5] S. Schuon and K. Diepold. Comparison of Motion Deblur Algorithms and Real World Deployment. *Paper on IAC 2006*, 2006.