Gaussian blur identification using scale-space theory

Philip Robinson, Yuko Roodt and Andre Nel
Faculty of Engineering and Built Environment
University of Johannesburg
South Africa
philipr@uj.ac.za, yukoroodt@gmail.com, andren@uj.ac.za

Abstract—**Image deblurring algorithms generally assume that the nature of the blurring function that degraded an image is known before an image can be deblurred. In the case of most naturally captured images the strength of the blur present in the image is not known. This paper proposes a method to identify the standard deviation of a Gaussian blur that has been applied to a single image with no *a priori* information about the conditions under which the image was captured. This simple method makes use of a property of the Gaussian function and the Gaussian scale space representation of an image to identify the amount of blur. This is in contrast to the majority of statistical techniques that require extensive training or complex statistical models of the blur for identification.**

Keywords-Gaussian blur, blur identification, blur estimation, scale space.

I. INTRODUCTION

In almost all vision systems, biological or mechanical, the phenomenon of blur can be observed. Blur manifests itself as a degradation of spatial detail or high frequency visual information. This results in a reduction of edge sharpness and loss of the finer detail. There are many causes of blur but the most fundamental is the diffraction limit of a vision system that contains an aperture [1]. Some other causes of blur are defocus, motion during exposure, atmospheric turbulence and upscaling of images [1, 2, 3].

Blurring is a distortion of an image that reduces the amount of information contained in that image. While it is impossible to build a physical system that can capture arbitrarily sharp images it is mathematically possible to reconstruct a portion of the lost information [4]. This process is called image deconvolution or image restoration and is essentially an inverse filtering process. The blurring effect is modelled as a convolution of the original image with a blurring kernel or Point Spread Function (PSF) with some additive white Gaussian noise as shown in the following equation [5].

$$i(x,y) = f(x,y) * h(x,y) + n(x,y), \qquad (1)$$

Where $i(x,y)$ is the distorted 2D image with the 2 dimensions denoted by x and y, $f(x,y)$ is the undistorted image, $h(x,y)$ is the blurring function PSF which is convolved with the input image and $n(x,y)$ is the additive white Gaussian noise present in the scene [5].

Usually it is assumed that the PSF of the blurring distortion is known. An operation is then performed that is the inverse of that distortion to attempt to undo that distortion [4, 5].

The image deconvolution problem has been explored quite thoroughly in the literature. The basic approaches of inverse filtering, least squares filtering and iterative filtering can be found in most image processing textbooks such as [5, 6]. More modern methods have also been discussed in [7, 8, 9] to name a few.

When the parameters of the PSF of the blurring function is not known and has to be estimated from the input image the problem becomes known as a blind deconvolution problem [5]. Blur identification techniques need to be employed to estimate the nature of the blur in the observed image. Numerous approaches to this problem have been proposed in the literature. The vast majority of approaches make use of image statistics to provide an estimate of the blur. In [10] a maximum likelihood estimation technique is used, [3] uses an autoregressive–moving-average (ARMA) process and [11] uses a regularization approach.

Non-statistical approaches also exist, for instance in [12] the original unblurred image is estimated and used to estimate what blur was applied to result in the degraded image. The approach that most resembles ours is a parametric approach where the blur is considered to conform to an assumed blur model with a single parameter. A search space of possible blur parameters is traversed and the input image is deconvolved with each parameter value. A sharpness metric is used to determine which parameter results in the sharpest output image. In this case the sharpness metric used was kurtosis [13].

There are a variety of types of blur found in images but we will focus on Gaussian blur. This blur approximates the blur caused by upsampling an image fairly well and is a very good approximation of blur introduced to an image by capturing a scene through atmospheric turbulence [5].

The technique proposed in this paper focuses on identifying the standard deviation (σ) of a Gaussian blur applied to an input image. An interesting property of the Gaussian function is employed to identify the variance of the Gaussian blur in the input image by examining its scale-space representation [14]. The scale-space representation has been used previously in [15] to detect edges. In [15] edges are considered to be ideal step functions that have undergone blurring due to lighting and focal characteristics of the imaging system through which they

were captured. These blurs were modelled as Gaussian blurs. Through analysis of the derivatives of an image at various scales in the scale-space it was possible to locate blurred edges and identify the degree to which they were blurred.

The remainder of this paper will be structured as follows. Section II will present the theory employed in this algorithm. Section III will describe the algorithm itself. Section IV will present some experiments and discussion of their results and finally Section V will be the conclusion.

II. BACKGROUND

A. An Interesting Property of the Gaussian distribution

In image processing the most common operations use kernel filters that are panned around the image. The Gaussian equation used to produce these types of kernels is considered to have a zero mean. Thus the one dimensional Gaussian equation we are using is defined as follows:

$$G(x, \sigma^2) = a e^{\frac{-x^2}{2\sigma^2}}, \qquad (2)$$

Where a is the amplitude of the curve and σ^2 is the variance of the Gaussian and its square root σ is the standard deviation [14]. For generating kernels for image processing a is generally considered to be 1.

The Gaussian equation exhibits self-similarity and thus the cascade property where if two Gaussians are convolved with each other they produce a new Gaussian as follows [14]:

$$G(x, \sigma_A^2) \otimes G(x, \sigma_B^2) = G(x, \sigma_A^2 + \sigma_B^2), \qquad (3)$$

In this paper we exploit an interesting feature of the Gaussian equation. Given a Gaussian with a constant standard deviation σ_1, if we convolve this Gaussian with another Gaussian with standard deviation σ_2 we get a resulting Gaussian with the standard deviation of $\sqrt{\sigma_1^2 + \sigma_2^2}$. If we then subtract the resulting Gaussian from the original Gaussian with standard deviation σ_1 and absolute the result we get a measure of the difference or error between the original Gaussian and the new Gaussian. This process is described in the equations below and figure 1.

$$E = |G(x, \sigma_1^2) - G(x, \sigma_1^2) \otimes G(x, \sigma_2^2)|, \qquad (4)$$

$$E = |G(x, \sigma_1^2) - G(x, \sigma_1^2 + \sigma_2^2)|, x \in [-B; B] \ \& \ \mathbf{Z}, \qquad (5)$$

Where x is an integer that ranges between integer bounds defined by $-B$ and B.

If you perform this process using a chosen value for σ_1 and a range of values for σ_2 and then plot the resulting errors you will find the response shown in figure 2. What is interesting is that the error curve contains a point of inflection where the concavity of the curve changes. To find the exact point of inflection we must look for extrema in the first derivative of the error curve which is shown in figure 3.
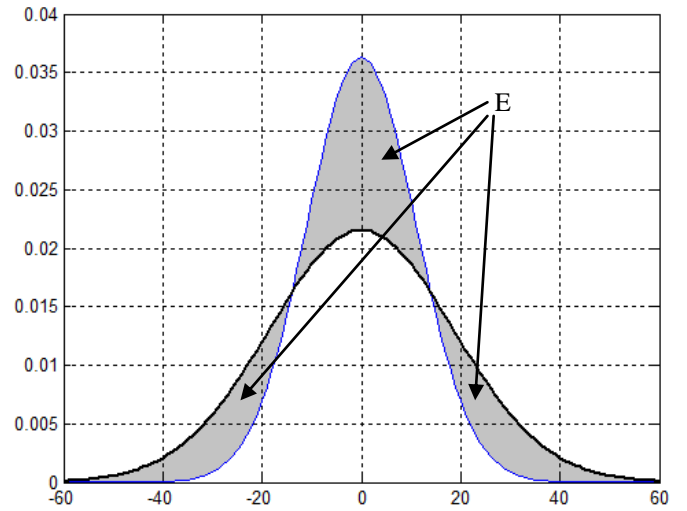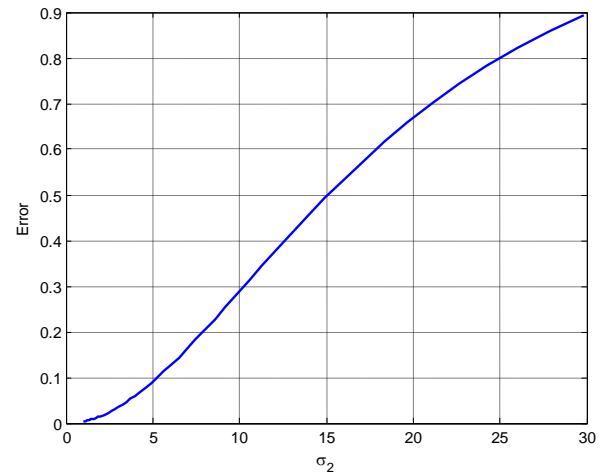

Figure 1: Error between 2 Gaussians

Figure 2: Error between a Gaussian with constant standard deviation $\sigma_1=11$ and a second Gaussian with standard deviation $\sqrt{\sigma_1^2 + \sigma_2^2}$ where σ_2 is varied over a range.
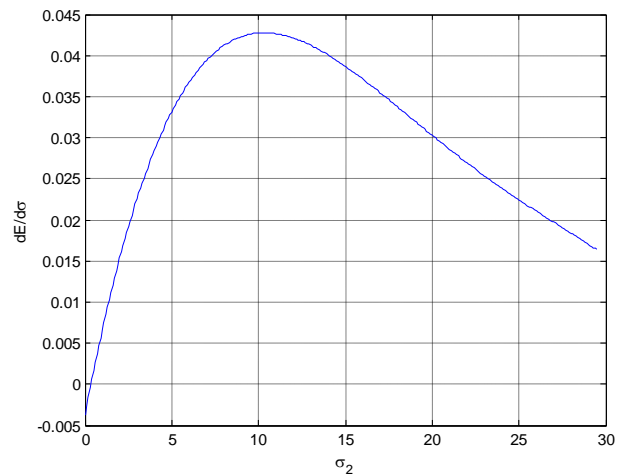

Figure 3: First derivative of the Error with respect to σ_2

As can been seen from figure 3 the maximum value of the first derivative of the error corresponds to the point of inflection in the error curve. This also corresponds with the chosen value for σ_1 which in this case was 11. This shows that while the error is increasing monotonically when σ_2 is smaller than σ_1 the error increases at a faster rate than when σ_2 is larger than σ_1. This phenomenon can be used to determine the value of σ_1 by only varying the value of σ_2 and searching for the point of inflection on the error curve.

B. The scale space

Scenes in the world appear very different when viewed from varying scales. For example a tree viewed from 1 meter away would be made up of individual branches, a trunk and leaves while if it was viewed from 1 km away it would appear to be a single solid object. The fact that scale is so important in describing the structure of objects being observed has led to the development of multi-scale representations of images. Being able to isolate the structures contained in an image at a given scale is an immensely powerful tool in being able to extract useful information from an image [14].

A large number of multi-scale representation techniques have been proposed in the literature. One of the first was the quad-tree representation which iteratively divides an image into smaller rectangles based on the information content inside each division [16]. Sampling pyramids have also been widely used. In these algorithms an image is recursively halved in size using a sub-sampling scheme and smoothed at each step to give a pyramid of images where each is half the size of level below. This approach is limited in the size of the steps at which its sampling size is reduced and thus objects at scales that exist between and levels of the pyramid are lost [14].

The scale-space representation was proposed to combat this problem. The scale-space is a representation that comprises a continuous scale parameter and preserves the same spatial sampling at all scales. It is shown in [14] that the only kernel that can achieve this is the Gaussian kernel. This approach takes an input image and blurs the image with a series of Gaussian kernels, each with a larger variance than the last. As the image becomes more and more blurred the finer scale information is averaged out and the larger scale structures are all that are left. In this way we can produce a series of images that each contain a different scale of structures but we do not introduce any quantization noise.

To take this representation a step further we can subtract each level of this multi-scale representation from the one below it to produce a Difference-of-Gaussian (DoG) representation of the image. This representation is essentially the second-order derivative of the images at each scale level. This multi-scale gradient information has been used in many feature detection, object detection and segmentation algorithms of which the most notable is probably the SIFT feature detector [17].

III. ALGORITHM DESCRIPTION

The algorithm described in this paper starts with an input image which we assume has been blurred with a Gaussian kernel as shown in the following equation.

$$I = F \otimes G(x, \sigma_1^2), \qquad (6)$$

Where I is the input image, F is the image without the blur and the function G is a Gaussian kernel with a standard deviation of σ_1. The goal of the algorithm is to identify the standard deviation of this blur with no *a priori* information about the conditions under which the image was captured.

The next step is to construct a scale space representation of the input image I. This is done by blurring the input image I with a range of Gaussian kernels with increasingly large standard deviations. The range of standard deviations is calculated in a similar fashion to [17]. We start at a standard deviation of 1 and we call each doubling of this initial value an octave of σ values. We choose how many levels to divide each octave into. The range is then constructed as described by the pseudo-code in the following figure. This code assumes we want to construct 5 octaves of σ values with 10 divisions in each octave.

```
octaveDivisions = 10
numOfOctaves = 5
scaleFactor = 2.0^(1.0/octaveDivisions)
numOfLevels = octaveDivision*numOfOctaves+1
sigma(1) = 1;

For s = 2 to numOfLevels
     Sigma(s) = sigma(s-1)*scaleFactor
end
```

Figure 4: Pseudo-code describing generation of σ values for the scale-space representation

To construct the scale-space representation D we then convolve the input image with a Gaussian kernel with each of the σ values in the generated range.

$$D(\sigma_2) = F \otimes G(x, \sigma_1^2) \otimes G(x, \sigma_2^2), \qquad (7)$$

Where σ_2 is the standard deviation from our generated range and σ_1 is the standard deviation of the Gaussian kernel we are trying to detect. The next step is to find the absolute error between the input frame and the images in the scale-space representation.

$$E(\sigma_2) = |F \otimes G(x, \sigma_1^2) - F \otimes G(x, \sigma_1^2) \otimes G(x, \sigma_2^2)|, \quad (8)$$

$$E(\sigma_2) = F \otimes |G(x, \sigma_1^2) - G(x, \sigma_1^2) \otimes G(x, \sigma_2^2)|, \qquad (9)$$

Due to the distributability of convolution it can be seen that the error E contains the equation 4 convolved with the unblurred image F. This implies that the same analysis of the error response of E can be applied to determine the value of σ_1.

Thus once we have the error response for all values of σ_2 in our scale-space we find the first derivative of E with respect to σ_2. We use the basic finite difference technique to estimate the derivative of the range of σ_2 values and E as following set of convolutions.

$$dE = E \otimes [-1 \ 1], \tag{10}$$

$$d\sigma = \sigma_2 \otimes [-1 \ 1], \tag{11}$$

Where the $[-1, 1]$ term is a discrete kernel with two elements. The final step of the algorithm is to find the maxima of $dE/d\sigma$ and the corresponding σ_2 value. This value is the detected standard deviation of the blur that the input image contained. This process of blurring an image with a series of Gaussians with increasing standard deviations is also used to produce the scale space representation of an image. Thus this algorithm can be cheaply performed in tandem with algorithms that make use of the scale space representation of an image.

We found that the algorithm was fairly sensitive to additive white noise and as such we introduced an iterative median filtering pre-processing stage to the algorithm to aid in suppressing noise. This stage consisted of applying two iterations of a 3x3 median filter to the input image before the above described process is performed.

IV. EXPERIMENTS

To examine the performance of the algorithm at detecting unknown Gaussian blurs in natural images we performed the following experiments. Four test photographs were chosen and are shown as figures 5 through 8 below.

Each image was degraded with a Gaussian blur with standard deviations ranging from 1 to 20. After each blur was applied an additive white Gaussian noise was applied resulting in a signal-to-noise (SNR) of 30 dB (strong noise) and 40 dB (milder noise). The algorithm was then used to measure the amount of blur in the image. The results of these experiments are displayed in figures 9 through 12.

Figure 5: Aircraft test image
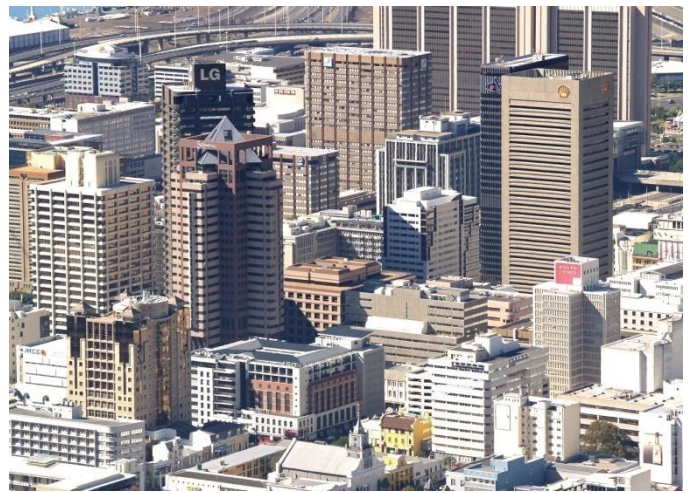
Figure 6: City test image

Figure 7: Bridge test image

Figure 8: Forest test image

As can be seen the algorithm successfully identifies the strength of the Gaussian blur applied to the images quite accurately in a range of standard deviations from 1 to 20

which is a far wider range of sigma values than algorithms currently in the literature. The presence of noise does decrease the accuracy of the identification especially in the Aircraft image which has large areas of uniform colour where noise becomes very apparent but the iterative median filtering does make the algorithm fairly resistant to noise.

It is interesting to note that in the Aircraft test image the strength of the identified blur does get over estimated. This is due to the large uniform coloured areas which have very little high frequency information content. This lack of high frequency content makes the images appear to be more blurred than they really are. In contrast the blur in the City test image is consistently underestimated due to the large amount of high frequency information present in the image. This over abundance of high frequency information makes the image appear to be less blurred than it is.
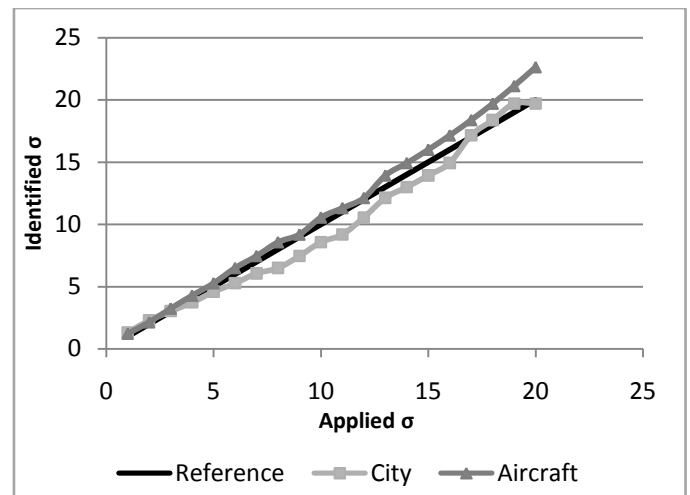

Figure 9: Blur identification results for City and Aircraft test images with 30 dB of additive white noise
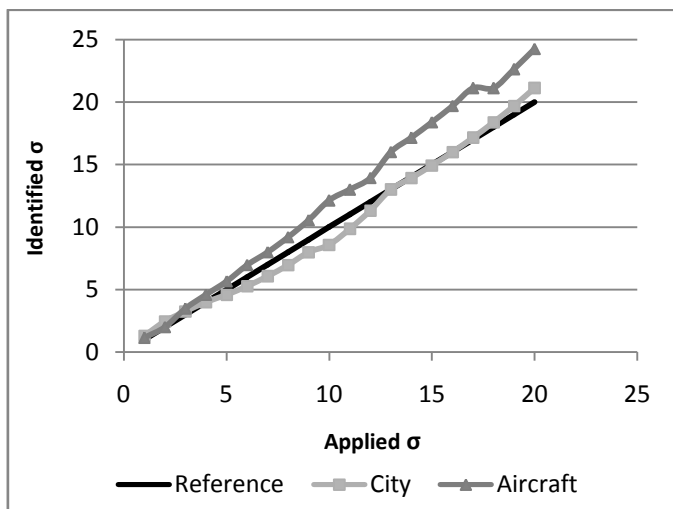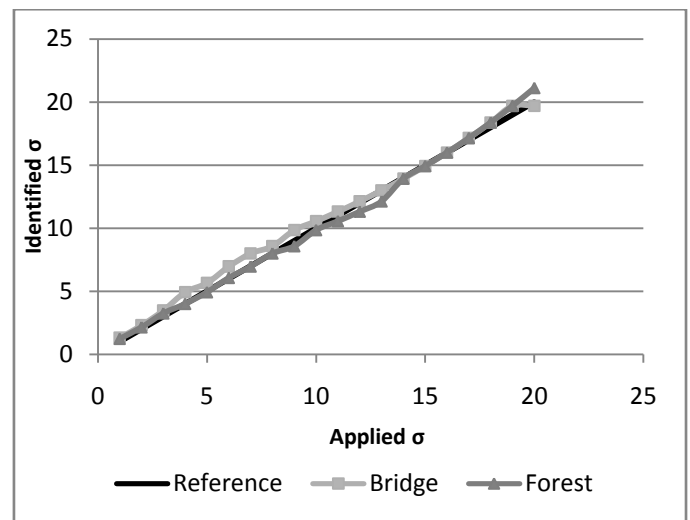

Figure 10: Blur identification results for Bridge and Forest test images with 30 dB of additive white noise

Figure 11: Blur identification results for City and Aircraft test images with 40 dB of additive white noise
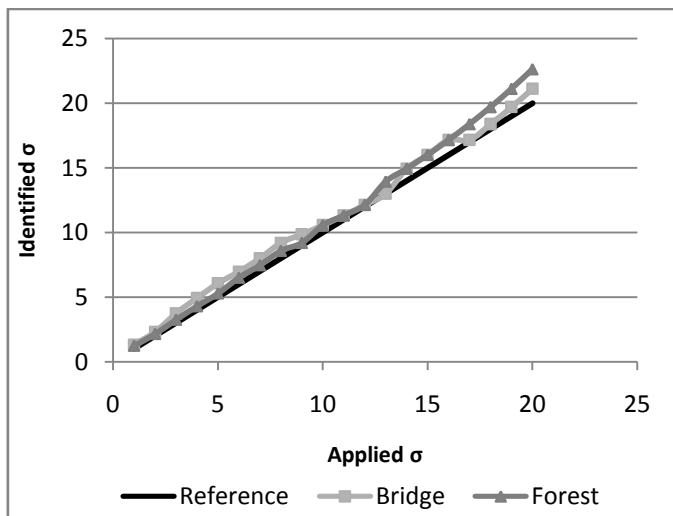

Figure 12: Blur identification results for Bridge and Forest test images with 40 dB of additive white noise

A final test was performed where an image that contains natural blur due to atmospheric turbulence is used as an input image. The strength of the blur is identified and the image is deconvolved using a plain Wiener filter using the identified Point Spread Function (PSF) [5]. The results of this experiment can be seen in figure 13. It is apparent that the identified blur strength is correct and the deconvolution deblurs the image without introducing ringing artifacts associated with an incorrectly identified PSF.

V. CONCLUSION

In this work is was shown that it is possible to detect the standard deviation of a Gaussian blur that has been applied to an image with no *a priori* information about the conditions under which the image was captured. The method uses an interesting property of the Gaussian function. When a series of Gaussians with increasing standard deviations are convolved with the Gaussian to be identified an error is produced. The error response this process produces has an inflection point

where the standard deviations of the Gaussians coincide and allows us to identify the standard deviation of the Gaussian being analyzed. This process is shown to work with a Gaussian blur applied to natural images. This method of blurring an image with a series of Gaussians is also used to produce the scale space representation of an image and can be performed in parallel with any algorithm that uses a scale space representation of an image.

The experiments show that in natural images with the presence of noise it is possible to identify Gaussian blurs with standard deviations that span a wide range without using any sort of statistical methods that require extensive training. It is also shown how this method can be used to identify the blur present in an image blurred naturally by atmospheric turbulence and allows one to deconvolve that image successfully using a basic Wiener filter.

Figure 13: The standard deviation of the blur present in a real image blurred by atmospheric turbulence is identified and used to deconvolve the image using a basic Wiener filter.

REFERENCES

[1] S. Winkler, *Digital video quality*, Wiley, 2005.

[2] H.R. Wu, K.R. Rao, *Digital Video Image Quality and Perceptual Coding*, CRC Press, 2005.

[3] S.J. Reeves, R.M. Mersereau, "Blur Identification by the method of Generalized Cross-Validation", *IEEE Transactions of Image Processing*, Vol. 1, No. 3, 1992.

[4] R.C Puetter, T.R. Gosnell, A. Yahil, "Digital Image Reconstruction: Deblurring and Denoising", *Annual Review of Astronomy and Astrophysics*, Vol. 43, pp 139-194, 2005.

[5] R.L. Lagendijk, J. Biemond. "Basic Methods for Image Restoration and Identification" in *Handbook of Image and Video Processing*. A. Bovik, San Diego: Academic Press, 2000, pp 125-139.

[6] A. K. Katsaggelos, *Digital Image Restoration*. Springer-Verlag, 1991.

[7] N. Joshi, C.L. Zitnick, R. Szeliski, D.J. Griegman, "Image deblurring and denoising using color priors", *Proceedings of CVPR. 2009*, pp. 1550-1557, 2009.

[8] A. Raj and R. Zabih. A graph cut algorithm for generalized image deconvolution. In ICCV '05, pages 1048–1054, 2005.

[9] Y. Harikumar, G.; Bresler. Exact image deconvolution from multiple fir blurs. IEEE TIP, 8(6):846–862, 1999.

[10] R. L. Lagendijk, A. M. Tekalp, and J. Biemond, "Maximum likelihood image and blur identification: a unifylng approach," J. Opt. Eng. 29,422435 (1990).

[11] H. Zheng, O. Hellwich, "Double regularized Bayesian estimation for blur identification in video sequences", *P.J. Narayanan et al. (Eds.) ACCV*, Vol. 3852, pp. 943–952. Springer, 2006.

[12] N. Joshi, R. Szeliski, D. Kriegman. "PSF estimation using sharp edge prediction". *In CVPR '08*, pages 1–8, 2008.

[13] D. Li, R.M. Mersereau, S. Simske, "Blur Identification based on kurtosis minimization", *ICIP 2005*, pp. 905-908, 2005.

[14] T. Lindeberg, "Scale-space theory: A basic tool for analysing structures at different scales", *Journal of Applied Statistics*, vol 21, no. 2, pp 225-270, 1994.

[15] J.H. Elder and S.W. Zucker, "Local scale control for edge detection and blur estimation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 7, pp 699- 716, 1998.

[16] S. Hanan, R. Webber, "Storing a collection of polygons using Quadtrees", ACM Transactions on Graphics, Vol. 60, No. 2, pp 182 - 222, 1985.

[17] D.G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91-110, 2004.