LOCALIZED BLUR ESTIMATION ON PHOTOGRAPHY IMAGES AND APPLICATIONS

Dun-Yu Hsiao (蕭敦育), Soo-Chang Pei (貝蘇章), and Chiou-Shann Fuh (傅楸善)[+]

Department of Electrical Engineering, National Taiwan University,
Department of Computer Science and Information Engineering[+] ,
Taipei, Taiwan 10617, R.O.C.
Email: r92942028@ntu.edu.tw, pei@cc.ee.ntu.edu.tw, fuh@csie.ntu.edu.tw[+]

ABSTRACT

With the prevalence of digital camera, more and more people have the chance to enter the field of photography. Defocus blurring easily occurs while taking pictures, and in some of the cases a defocused image is considered as an image of defect. In this paper, we determine the blurred area automatically on a single image. There are plenty of former studies about how to perform image enhancement or blur removal on the defocused region to restore the content. However, most of these algorithms require iterative operations, complex blur models or many image sources to acquire a result. The idea proposed in this paper is beneficial to image restoration, whereas in most of the cases image enhancement usually restores the entire image but not just the blurred area only. If automatic blur detection could be done in prior, the operation of image restoration would be faster and the image quality could be better. In the view point of computer vision, the knowledge about blurred region may offer a useful clue for depth prediction or other usages. At the end of this paper, we proposed some possible applications where this blur estimation idea can be applied to.

1. INTRODUCTION

As recent advances in digital photography, coding and electronic optical technologies have made digital camera very popular; more and more people are encouraged to take pictures with their own digital cameras. It is very easy to ruin a picture if the photographer (or user) does not set the focus of camera correctly. Hence, there are lots of commercial software packages helping people to restore their images. Nevertheless, with these software tools, usually plenty of time must be taken to get one enhanced image.

Many approaches for blurring detection have been proposed in the literature [2] - [27]. However, most of them are pretty slow and complicated. If the speed of the image restoration process cannot catch the pace people take pictures, that process is not likely to be accepted by the masses. An efficient and real-time algorithm for automatic blur detection in digital photography will thus be very useful.

In this paper, we propose a new scheme for blur estimation. First, we apply a procedure similar to Marichal's [1], which uses some DCT domain knowledge to estimate a rough quality of the entire image by calculating and summarizing the histogram of DCT coefficients. The first phase provides us the prior knowledge about the image. Next, some normalization and thresholding operations are performed on each of the DCT blocks. Some metrics are then defined to distinguish blurred regions from sharp regions. The last phase is to estimate the degree of blurring by combining the prior knowledge and the distance information. After those operations, the blurred regions are thus detected.

2. PREVIOUS WORK

There are abundant studies in the field of blur estimation. Marichal et al. [1] tries to measure a global camera blur in a simple yet robust way. Based on the histogram of non-zero DCT coefficients, a 3-phase scheme is developed to accomplish the entire estimation process. The first phase is to take into account all the DCT coefficients of the entire image as a whole. This global information offers the idea about the degree of global sharpness/blurring. Since these non-vanishing DCT coefficients are actually image-dependent, the second phase is to look at the null coefficients only. These blurred images tend to have most of their high frequency DCT coefficients set to zero regardless their image contents. Then, in the last phase, the factor of image size is removed by normalizing the histogram information. The number of non-zero occurrences of an AC DCT coefficient is divided by the number of non-zero occurrences of the DC coefficient. After these three phases, a quality number ranging from 100% (no blur) to 0% (totally blurred) is generated. The pseudo codes of this process are provided in Fig. 1. The method proposed by Marichal et al. has the advantages in its simplicity, but it does not provide the blurring measure in a correct sense when applied to an extremely blurred image. That is, when we test this algorithm on an entirely out-of-focus image, the resulting quality values may indicate that the images are just partially blurred. For example, the quality number for Fig. 3 (b) is estimated to be only about 80% blurred. Empirically, we found that the totally blurred value is 0% only if the image contains a single monotone color. This is unlikely to be the case when we deal with a blurred image. As the experiments done above, the method proposed by [1] would provide non-linear result as the quality proportional to the percentage of blurred regions.

For the sake of simplicity, we didn't consider those algorithms which are highly complicated or require multiple images to determine blurred regions. For more details and other studies, readers are referred to the references.

```
MinDCTValue; /* Minimum DCT value to take
                into account, typically 8 */
MaxHistValue;/* Histogram relative frequency
                to reach, typically 0.1 */

/* Constants for measure weighting */
Weight[64]={8,7,6,5,4,3,2,1,7,8,7,6,5,4,3,2,
6,7,8,7,6,5,4,3,5,6,7,8,7,6,5,4,4,5,6,7,8,7,
6,5,3,4,5,6,7,8,7,6,2,3,4,5,6,7,8,7,1,2,3,4,
5,6,7,8};
TotalWeight = 344;

/* Variables for computation */
DCTnonzeroHist[64]; /* histogram */
blur;  /* blur measure */

/* Initialization of histogram to zero */
for (k = 0; k < 64; k++)
   DCTnonzeroHist[k] = 0;

/* Compute Histogram */
for (all macroblocks)
   {
    for (every luminance block)
       {
        for (every DCT component k)
               /* 0 <= k < 64 */
          {
           /* Add to histogram if coeffi-
              cient is big enough */
          if (abs(blockDCT[k]) > MinDCTValue)
            DCTnonzeroHist[k]++;
          }
       }
 }

/* Estimate blur via weighting matrix */
blur = 0;
for (every DCT component k) /* 0 <= k < 64 */
/* add the corresponding weight for all

   coefficients with a sufficient number of
   occurences */
if (DCTnonzeroHist[k] <
      MaxHistValue*DCTnonzeroHist[0])
   blur += Weight[k];

/* divide by the sum of all weights */
blur /= WeightTotal;

/* 1-blur is the quality percentage of the
   frame in terms of blur */
```

Figure 1. Pseudo codes proposed by [1].

3. DISCOVERY AND LOCALIZED BLUR ESTIMATION SCHEME

Our idea starts from the variation of DCT coefficients when a blur filter is processed over an image. We use a Gaussian blurring filter to approximate the true blurring kernel in real defocused images. We discover that the DC coefficient tends to become greater whereas those AC coefficients tend to become smaller when the blurring operator is applied. Hence if we could separate those regions with high DC and low ACs from those regions with an opposite distribution of DCT coefficients, it is possible to tell which parts of the images are blurred. Supported by experiment results, we will demonstrate the effectiveness of this idea in blur estimation. There are some more discoveries while examining the variations of DCT coefficients. Further; discussions will be given in Section 6.

The proposed scheme consists of 3 main phases. Details of each phase are to be presented as follows:

1. The first phase is set to adjust the threshold to distinguish blurred regions. A problem has been encountered when we looked at entire non-blurred images and totally-blurred images, see Fig. 4 (a), (b). Mean DCT coefficients could separate the difference between blur and sharp regions and comprehensible for human. But it does not provide the true boundaries about the blurred regions, thus a thresholding operation must be done here to obtain blurred regions. If the threshold is a constant, inevitably some regions in an entirely non-blurred image will be declared as blurred and vice versa. To aid this problem, an adaptive threshold is applied. We review on the method proposed by [1] and consider the quality measure might helpful in this case. After modifying a bit of its algorithm about the weighting matrix, we gain some improvement on the global blur estimation. The quality range of the estimation on overall sharp/blurred image is extended from 100%/80% to 100%/70%. Now the modified algorithm could provide a global blur estimation which could be taken as the prior knowledge about the target image. With this prior information, we could first detect the image is globally blurred or not then we adjust our threshold to meet the requirement about the condition of the target image in the final phase to obtain good estimation.
2. The middle phase is the blur estimation core of the whole process. A mean DCT block of coefficients of the target image is calculated in this phase. Then we process through every DCT block to normalize to this mean value based on the observation of the phenomena when we blur an image. It is the separation operation to distinguish blurred regions from sharp regions which originates from the idea described at the beginning of this section. Taking the mean DCT coefficients as a normalization threshold is generally good because the mean values are more content dependent thus the separation is more accurate and independent of what content is. After the normalization is done, we convert the image into spatial domain and do a smooth procedure to reduce the edginess of the resulting image. Till now we have an overview about the image of its blur condition. The brighter parts of the image indicate that those regions are

sharper. The extremely bright areas we call the areas are at focus. Therefore, the darker parts of the image are in the sense of blurred regions or smooth surfaces. Human could have the feeling about those differences, but more steps shall be done for a better threshold and for computer to realize the knowledge for blurred regions.

3. The last phase is done by varying thresholds from the interval of mean minus four times of standard deviation to plus 4 times of standard deviation. The amount of threshold is varying with the quality factor acquired from the first phase. Blurred regions are shown as white regions in the detection results. For experiment results, please refer to the experiment result section.

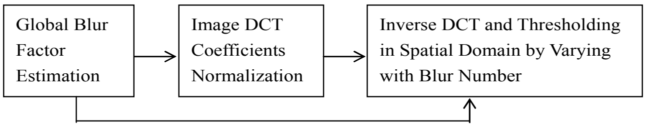


Figure 2. Block diagram of the blur estimation scheme proposed.

4. EXPERIMENT RESULTS

We have test on several kinds of images, the following results are grouped by the manner of test images.

In this group, results are the of normal photography image. We perform our test on an image which has partial sharp and blurred regions. As seen in Fig. 3, (a) is the input image, which has a quality number of 78.07%; this number is generated from the first phase, indicates the degree of global sharpness. Fig.3 (b) is the normalized image of Fig. 3 (a), which is the result of the second phase. It could be observed that sharper regions are brighter and blurrier regions are darker. After thresholding, final result is shown in Fig. 3 (c), black blobs indicate sharp regions and white blobs indicate blurred ones. This algorithm does indicate the sharp regions (keypad on the phone) and blurred ones (backgrounds). Blurred regions are locally estimated, blur percentage is 88.3848 %. The final results and are quite close to our expectation.

Since the algorithm has proven its effectiveness in localized blur detection on normal images, we furthermore verify our method on some other extreme sharp/blur cases. Fig. 4 represents the cases of overall sharp/blurred images. In this group of experiment, the essentiality of the first phase is unfolded. If the threshold is not taken adaptively, we will expect some blurred region detection in an overall sharp image or some sharp region detection in an overall blurred image. Our method could avoid those misdetections via the global sharp/blur degree estimation. The quality number is 100% in Fig. 4 (a), which means a tight threshold is applied to this image; hence the blur detection is 0% (Fig. 4 (e)). On the other hand, the image of Fig. 4 (b) has a quality number of 73.75%, which will result in a loose threshold on this image; the blur detection result is 100% blurred. In these cases, extreme sharp/blur images are detected correctly.

After those experiments on images which are blurred by defocus, we perform our test on those images which are blurred by Gaussian filtering. The test image Fig. 5 (a) is the texture from ordinary floor (shot is taken perpendicularly to the floor), and Fig. 5 (b) is the image after applying Gaussian blur of 4 pixels on (a). We want to determine whether we algorithm is working on other blurred images or not. The results of Fig. 5 (e), (f) are clear. The image that Gaussian filtering is not applied is having quality number of 100% and the blur percentage is 0.0156 %. The error might result from some minor lens distortion or defocus. For the image applied Gaussian filtering, the quality number is 47.84% and the blur percentage is 100%. Thus the results are very convincing.

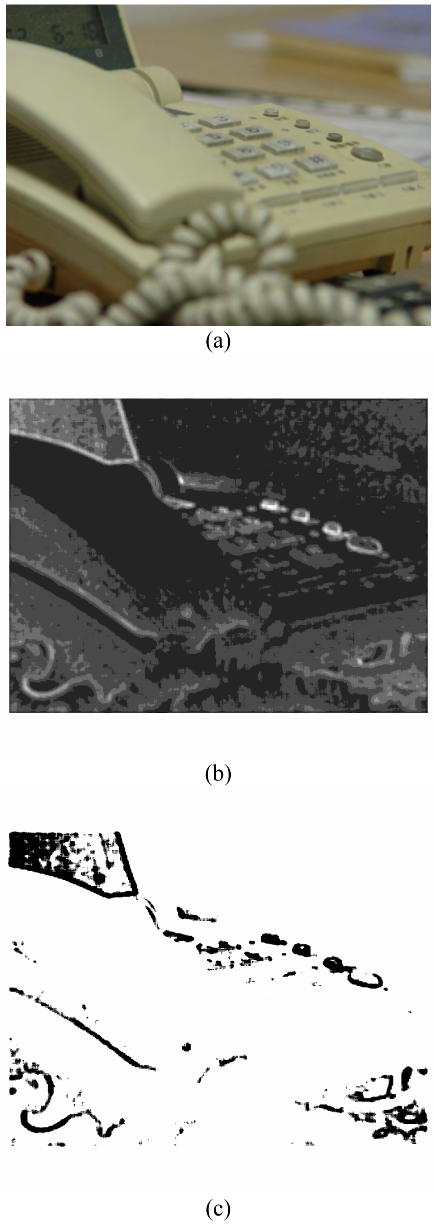
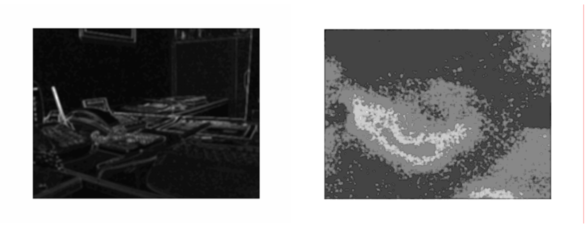


Figure 3. Local blur estimation on a partially defocus blurred image, (a) a partially defocus blurred image, , (b) the blur estimation result of (a) after the second phase, with brighter blobs indicating sharper regions while

darker blobs indicating blurrier regions , (c) the final blur estimation result of (a), with white blobs indicating blurred regions.

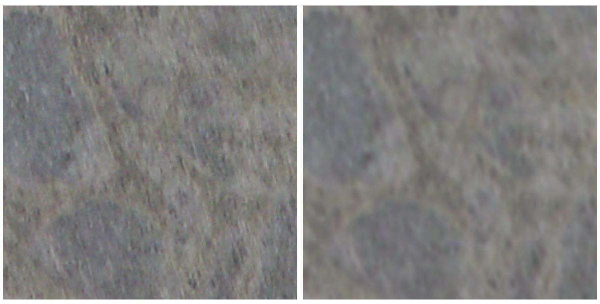
(a) (b)



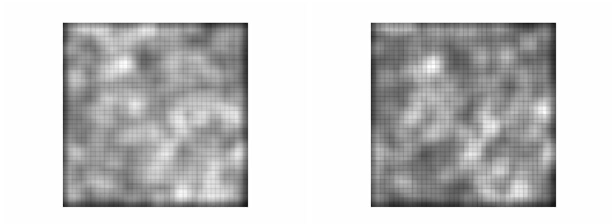
(c) (d)

(e) (f)

Figure 4. Comparison of local blur estimation between an overly sharp image and an overly blurred image, (a) an overly sharp image, (b) an overly blurred image, (c) the blur estimation result of (a) after the second phase, with brighter blobs indicating sharper regions while darker blobs indicating blurrier regions , (d) the blur estimation result of (b) after the second phase, (e) the final blur estimation result of (a), with white blobs indicating blurred regions, (f) the final blur estimation result of (b).



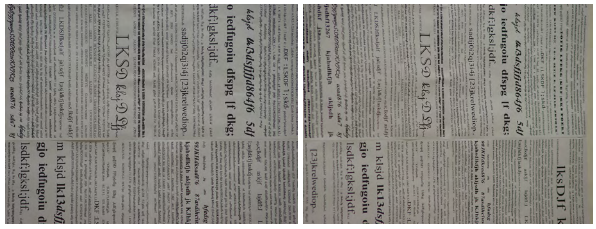
(a) (b)



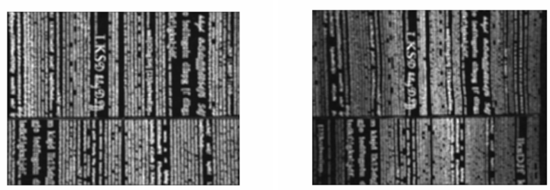
(c) (d)

(e) (f)

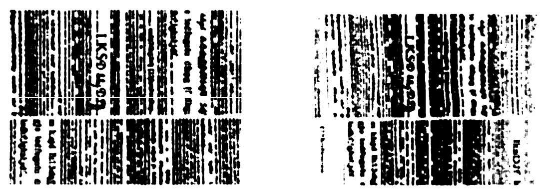
Figure 5. Comparison of local blur estimation between an user generated overly sharp and an user generated overly blurred image, (a) an overly sharp image, (b) an entire Gaussian blurred image of (a), (c) the blur estimation result of (a) after the second phase, with brighter blobs indicating sharper regions while darker blobs indicating blurrier regions , (d) the blur estimation result of (b) after the second phase, (e) the final blur estimation result of (a), with white blobs indicating blurred regions, (f) the final blur estimation result of (b).



(a) (b)



(c) (d)



(e) (f)

Figure 6. Local blur estimation on images defocus from different shot angle of a word poster, (a) an overly sharp image which is taken perpendicularly (90 degrees) to the poster, (b) an partially defocus blurred image which is taken from 40 degrees to the poster, (c) the blur

estimation result of (a) after the second phase, with brighter blobs indicating sharper regions while darker blobs indicating blurrier regions , (d) the blur estimation result of (b) after the second phase, (e) the final blur estimation result of (a), with white blobs indicating blurred regions, (f) the final blur estimation result of (b).

Next we experiment on images defocused via different aspect angle. For a plane, if the shot is taken perpendicularly (90 degrees to the plane), without taking account of lens distortion, the shot image shall be purely sharp. While varying the shot aspect angle from 90 degrees to 0, there would be more and more blurred regions caused from defocusing. In Fig. 6, we look into the case of continuous defocus cases. The blur percentage of Fig. 6 (a) and (b) are 46.9242 % and 54.5535 % respectively. This poster is generated from random word sequences with arbitrary font and size. The misdetection on blurry might result from the empty background, which is the paper. This might be considered as a special case.

The last group of results is performed on a real plane with texture. The test plane is a large wood plate; several shots have been taken from different aspect angles. Here is the comparison of the shots taken in the angle of 90 degrees and 40 degrees to the plane. From Fig. 7, we could see that the misdetection does not exist in this case. We have quality number of 100% and blur percentage of 0% in the image shot at 90 degrees, and quality number of 87.38% and blur percentage of 18.0129% in the image taken at 40 degrees. From the outcomes, we might say that this algorithm is somewhat content-dependent; but in most ordinary cases with textures, this method offers reliable results.

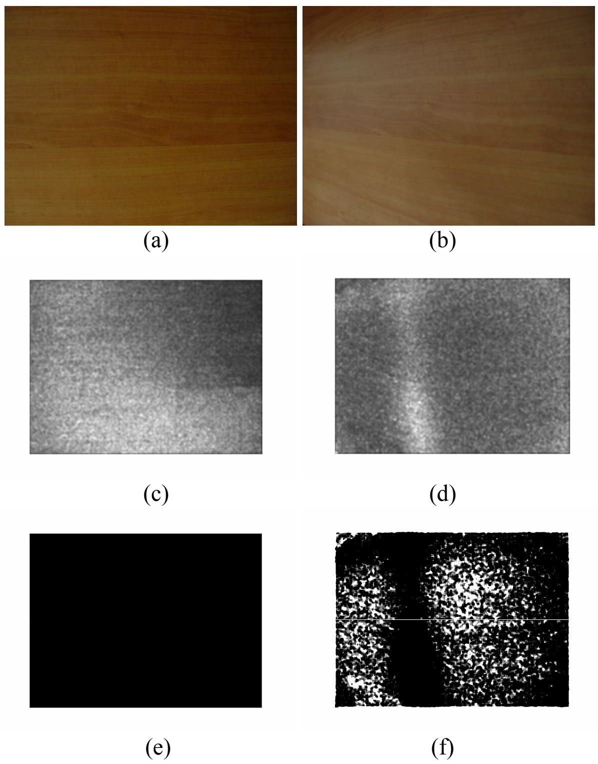


Figure 7. Local blur estimation on images defocus from different shot angle of a wood plate, (a) an overly sharp image which is taken perpendicularly (90 degrees) to the plane, (b) an partially defocus blurred image which is taken from 40 degrees to the plane, (c) the blur estimation result of (a) after the second phase, with brighter blobs indicating sharper regions while darker blobs indicating blurrier regions , (d) the blur estimation result of (b) after the second phase, (e) the final blur estimation result of (a), with white blobs indicating blurred regions, (f) the final blur estimation result of (b).

From above results, it could be shown that our proposed scheme is effective in localized blur detection and most outcomes it offers are reliable. In Fig. 6 and Fig. 7 given above, it implies that if we could tablet all the blur percentage results of an image related to its shot angle of a certain plane, we might be able to identify the aspect angle of that plane to the camera next time an image of the plane is taken. And this might be useful in robotic vision applications.

	Fig. 3 (a)	Fig. 4 (a)	Fig. 4 (b)	Fig. 5 (a)	Fig. 5 (b)	Fig. 7 (a)	Fig. 7 (b)
Quality number	78.07%	100%	73.75%	100%	47.84%	100%	87.38%
Blur percentage	88.38%	0%	100%	0.02%	100%	0%	18.01%

Table 1. Results of experiment groups, the poster (Fig. 6) data is excluded because the content of the image is not likely to occur in ordinary

5. ERROR DISCUSSIONS AND BOTTLENECKS

In most of the time our method is reliable, but it is under some assumptions. As the scheme we proposed is a three stage algorithm, the error could be resulting from either one of the stages. Stage 1 which is the prior knowledge obtaining stage has huge effect on the threshold that the next stage chooses. In some special cases it would result in inaccuracy of blur estimation, if some learning effort could be spent on this stage, the prediction results shall be very robust. Since our method is plainly based on the DCT domain knowledge, inevitably we would encounter the problem that how to distinguish a plain, stainless surface (such as a plain wall) from a blurred scene. Thus when the image contains a great amount of smooth, single-color region, our estimator might have a chance of misdetection. But this is the ill defined problem in computer vision, in some extreme cases, even human can’t tell a blurred region from a smooth one. In [4], [11] also have mentioned about this problem, to relief from it, they define the scene must not be entirely empty nor smoothed. Hence we set up some criteria such as our target scene must be textured and is taken from camera to prevent some misdetection and increase our accuracy.

6. FUTURE APPLICATIONS

As we mentioned before in the earlier sections, this estimation method could be widely applicable. Taking the effect of defocus, we could see that human can interpret the blur effect into the sense of the aspect angle of the surface he is looking at or to tell from near or far part of a plane. By obtaining few known aspect angle images of the target plane, and combining with the blur estimation method, we could identify the surface aspect angle via the distribution variation and the amount percentage of the blurred regions. Thus it is helpful in some robot vision problems and provides another way of passive detection.

7. CONCLUSIONS

The new scheme in blur estimation has been described in this paper. Tests of this method are operated on a computer environment of 1.3G CPU with 640MB RAM. For each run of estimation, the calculation interval depends on the size of the target image and varies from couples of seconds (5 mega pixels image) to instant. Experiment results show that this scheme has its power in blurred regions detection and the advantages of simplicity. By handling the problem criteria carefully, this scheme is useful in other applications.

8. REFERENCES

[1]. Marichal, X.; Ma W. Y.; Zhang H. J.; "Blur determination in the compressed domain using DCT information" in *Proceedings of International Conference on Image Processing*, vol. 2, pp. 386 - 390, 1999

[2]. Chiang M. C.; Boult, T.E.; "Local blur estimation and super-resolution" in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 821 – 826, 1997

[3]. Chardon, S.; Vozel, B.; Chehdi, K.; "A comparative study between parametric blur estimation methods" in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 6, pp. 3233 – 3236, 1999

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

[5]. Kim S. K.; Park S. R.; Paik J. K.; "Simultaneous out-of-focus blur estimation and restoration for digital auto-focusing system" in *IEEE Transactions on Consumer Electronics*, vol. 44, No. 3, pp. 1071 – 1075, 1998

[6]. Kubota, A.; Kodama, K.; Aizawa, K.; "Registration and blur estimation methods for multiple differently focused images" in *Proceedings of International Conference on Image Processing*, vol. 2, pp. 447 – 451, 1999

[7]. Nayar, S.K.; Watanabe, M.; Noguchi, M.; "Real-time focus range sensor" in *Proceedings of Fifth International Conference on Computer Vision*, pp. 995 – 1001, 1995

[8]. Nayar, S.K.; Watanabe, M.; Noguchi, M.; "Real-time focus range sensor" in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, No. 12, pp. 1186 – 1198, 1996

[9]. Kim S. K.; Paik J. K.; "Out-of-focus blur estimation and restoration for digital auto-focusing system" in *Electronics Letters*, vol. 34, No. 12, pp. 1217 – 1219, 1998

[10]. Farsiu, S.; Robinson, D.; Elad, M.; Milanfar, P.; "Fast and robust super-resolution" in *Proceedings of International Conference on Image Processing*, vol. 2, pp. II - 291-4 vol.3, 2003

[11]. Elder, J.H.; Zucker, S.W.; "Scale space localization, blur, and contour-based image coding" in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 27 – 34, 1996

[12]. Rooms, F.; Philips, W.; Van Oostveldt, P.; "Integrated approach for estimation and restoration of photon-limited images based on steerable pyramids" in *4th EURASIP Conference focused on Video/Image Processing and Multimedia Communications*, vol. 1, pp. 131 – 136, 2003

[13]. Price, J.R.; Gee, T.F.; Tobin, K.W., Jr.; "Blur estimation in limited-control environments" in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 1669 – 1672, 2001

[14]. Marziliano, P.; Dufaux, F.; Winkler, S.; Ebrahimi, T.; "A no-reference perceptual blur metric" in *Proceedings of International Conference on Image Processing*, vol. 3, pp. III-57 - III-60, 2002

[15]. Yifeng L.; Lampropoulos, G. A.; Qian S. E.; "New technologies for improving spatial resolution of satellite images" in *IEEE International Geoscience and Remote Sensing Symposium*, vol. 6, pp. 3326 – 3328, 2002

[16]. Yap K. H.; Guan L.; "An attractor space approach to blind image deconvolution" in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 1833 – 1836, 2001

[17]. Kim S. K.; Park S. R.; Paik J. K.; "Simultaneous Out-of-focus Blur Estimation And Restoration For A Digital Auto-focusing System" in *International Conference on Consumer Electronics*, pp. 430 – 432, 1998

[18]. Choi J. W.; Kang M. G.; Park K. T.; "An algorithm to extract camera-shaking degree and noise variance in the peak-trace domain" in *IEEE Transactions on Consumer Electronics*, vol. 44, No. 3, pp. 1159 – 1168, 1998

[19]. Zhang, J.; "The mean field theory in EM procedures for blind Markov random field image restoration" in *IEEE Transactions on Image Processing*, vol. 2, No. 1, pp. 27 – 40, 1993

[20]. Vozel, B.; Chehdi, K.; Carton-Vandecandelaere, M.-P.; "A blind restoration system of blurred and noisy numerical images" in *Proceedings of International Conference on Image Processing*, vol. 1, pp. 105 – 108, 2000

[21]. Deschenes, F.; Ziou, D.; "Homotopy-based computation of defocus blur and affine transform" in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. I-398 - I-404, 2003

[22]. Jalobeanu, A.; Blanc-Feraud, L.; Zerubia, J.; "Estimation of blur and noise parameters in remote sensing" in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, pp. IV-3580 - IV-3583, 2002

[23]. Rooms, F.; "Estimating image blur in the wavelet domain" in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, pp. IV-4190, 2002

[24]. Chitale, S.; Padgett, W.T.; "Blur identification and correction for a given imaging system" in *Proceedings of IEEE Southeastcon*, pp. 268 – 273, 1999

[25]. Zion, D.; Wang, S.; Vaillancourt, J.; "Depth from defocus using the hermite transform" in *Proceedings of International Conference on Image Processing*, vol. 2, pp. 958 - 962, 1998

[26]. Gokstorp, M.; "Computing depth from out-of-focus blur using a local frequency representation" in *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, vol. 1, pp. 153 – 158, 1994

[27]. Trussell, H.J.; Vora, P.L.; "Bounds on restoration quality using a priori information" in *International Conference on Acoustics, Speech, and Signal Processing*, vol.3, pp. 1758 – 1761, 1988