반도체디스플레이기술학회지 제19권 제3호(2020년 9월)
Journal of the Semiconductor & Display Technology, Vol. 19, No. 3. September 2020.

Local Binary Pattern Based Defocus Blur Detection Using Adaptive Threshold

Muhammad Tariq Mahmood* and Young Kyu Choi*†

*† School of Computer Science and Engineering, Korea University of Technology and Education

ABSTRACT

Enormous methods have been proposed for the detection and segmentation of blur and non-blur regions of the images. Due to the limited available information about the blur type, scenario and the level of blurriness, detection and segmentation is a challenging task. Hence, the performance of the blur measure operators is an essential factor and needs improvement to attain perfection. In this paper, we propose an effective blur measure based on the local binary pattern (LBP) with the adaptive threshold for blur detection. The sharpness metric developed based on LBP uses a fixed threshold irrespective of the blur type and level which may not be suitable for images with large variations in imaging conditions and blur type and level. Contradictory, the proposed measure uses an adaptive threshold for each image based on the image and the blur properties to generate an improved sharpness metric. The adaptive threshold is computed based on the model learned through the support vector machine (SVM). The performance of the proposed method is evaluated using a well-known dataset and compared with five state-of-the-art methods. The comparative analysis reveals that the proposed method performs significantly better qualitatively and quantitatively against all the methods.

Key Words : Blur Measure, Blur Segmentation, Local Binary Pattern, Support Vector Machine, Adaptive Threshold

1. Introduction

Generally, images captured using optical imaging systems contain blurred and sharp regions. Blur can be categorized mainly into defocus blur and motion blur. Detection and classification of blurred and non-blurred regions is very vital in many computer vision applications including segmentation of images [1], object detection [2] and classification of scenes [3]. Blur detection with multiple images and blur detection with the single image are the two major classes of blur detection techniques. In the first class, multiple images of the same scene are taken with multiple focal setting and align to acquire a high-quality image. This method has restricted acceptability due to the limitation of scene alignment. Blur detection with single image uses a single image to compute the blur map and hence it is the most preferred method among the researchers. Recently, a number of techniques have been proposed to address this problem [4,5]. Generally, these techniques consist of two key steps. Firstly, a blur measure operator is used to differentiate the blurred pixels from the sharp pixels in the image. It provides an initial blur map. Then, a classification method is applied which produces the segmented or classified blur map.

In literature, many blur measures for blur detection are proposed [6-8]. Krotkov and Martin estimate the entropy and range of the histogram of the image [9], which is utilized to calculate the blur map using frequency of the pixels. Shi et al. discriminate the blur and non-blur region using several features. First, by calculating the average power spectrum in the frequency domain, second, computing the gradient distribution of the local patches then evaluating the kurtosis since kurtosis varies in blurred and sharp regions [6]. Golestaneh and Karam exploited the

†E-mail: ykchoi@koreatech.ac.kr

difference in frequency for blur and non-blur region of the image and computed the spatially varying blur by applying multiscale fusion of the high-frequency Discrete Cosine Transform (DCT) coefficients (HiFST) [10].

After calculating the blur map, the next step is to segment the blur and sharp regions of the map [11]. Shi et al. [6] used the graph-cut method of [12] and the value 0.9 and below 0.1 to S and T given in it. Yi and Eramian proposed an algorithm for segmenting blurred and sharp regions. Multi-scale sharpness map generation, alpha matting initialization, alpha map computation, and multi-scale sharpness inference are the four main steps of the algorithm [13]. Golestaneh and Karam used fixed threshold (0.98) to generate camera focus point map [10].

In this paper, we propose an effective blur measure based on the local binary pattern (LBP) with an adaptive threshold for blur detection. The LBP based sharpness metric of [13] uses a fixed threshold irrespective of the type and level of blur that may not be suitable for images with variations in blur amount and type. Contradictory, we generate sharpness metric of each image separately using different threshold which is determined empirically. Our proposed method uses an adaptive threshold for each image based on its blur properties, and generates improved sharpness metric. The adaptive threshold is computed based on the model learned through support vector machine (SVM). To develop the SVM based model, first, we prepare the training data which consist of a feature vector and target value for each image. The feature consists of various measures that capture the variations in the image. The performance of the proposed method is evaluated on a well-known dataset and is compared with five state-of-the-art methods. The comparative analysis reveals that the proposed method performs significantly better qualitatively and quantitatively against all of the comparative methods.

2. Proposed Method

In this study, a method is developed to find the best threshold for every image in the dataset to be used in LBP instead of giving a fixed threshold. The block diagram of the proposed method is shown in Fig.1. First, the feature vector $\mathbf{f}$ is generated from the input image. This feature vector is then used to develop an SVM model $\hat{\mathbf{g}}(\mathbf{f})$, which will predict the threshold $\mathbf{t_{var}}$ for each image. In the next step,

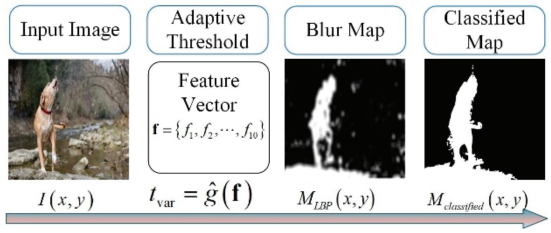

Fig. 1. Block diagram of the proposed method.

LBP sharpness metric $\mathbf{M_{LBP}}$ of each image is computed using its best threshold $\mathbf{t_{var}}$ and, lastly the binarization of retrieved sharpness map is done to achieve $\mathbf{M_{classified}}$. The proposed method is further explained in subsequent subsections.

2.1 Model for Adaptive Threshold

The main objective of this section is to develop an SVM based classifier $\hat{g}(f)$.

Data Preparation: In data preparation, first, we create a set of useful features. We have computed a feature vector with ten features named as $(f_1, f_2 \dots f_{10})$. The detail of the features is elaborate in Table 1. However, the model accuracy can vary by choosing different features for learning. We use images of dataset to generate the data. In this way, for each image 1×10 dimensional features vector $f = (f_1, f_2, \dots, f_{10})$ is computed. To train a model every feature vector f should have a target value, which is the best threshold for each image in our case. The best threshold t is computed as according to Algorithm 1. Mathematically, training data can be represented as:

$$D_T = \{f^{(i)}, t^{(i)}\} \qquad i = 1,2,\dots,N_1$$

Algorithm 1: Find best threshold t for each image
Input: I:graylevel image, GT:ground truth blur map
Output: t: best threshod

```
max ← −1
for i = T_min : T_max
    compute sharpness metric using Eq. 3
    compute accuracy and F-measure
    arr = accuracy (i) + F − measure (i)
    if arr > max then
        max = arr
        t = i
    end
end
```

where N_1 is the sample size of the training data. In our experiment total size of the training and testing data is $N = N_1 + N_2$, where N_2 is the sample size of the test data D_{test}.

Model Learning: A classifier is evolved to solve a multiclass problem using support vector machine (SVM) binary learners. Multiple binary classifiers can be used to construct a multiclass classifier by decomposing the prediction into multiple binary decisions [14].

To decompose the binary classifier decision into one, we have used *'onevsall'* coding type. Each class in the class set k is individually separable from all the other classes and for each binary learner, one class is taken as positive, and the rest is taken as negative. This design uses all the combinations of positive class for the binary learner. Non-linearity in the features is taken care of by kernel function by transforming nonlinear spaces into linear spaces. In our experiment, we have used 'Gaussian' as the kernel function. Training data D_T is used to train an SVM multiclass classifier $\hat{\mathbf{g}}(\mathbf{f})$ for multiclass classification. The evolved classifier $\hat{\mathbf{g}}(\mathbf{f})$ takes the value of feature vector $\mathbf{f}$ of an image as input and classify it into one of the nine classes i.e. $(0,1,2,\dots,8)$.

2.2 LBP based Blur Measure

Conventional LBP [15] work by taking a window for each grayscale pixel i_c. For each central pixel i_c the 8-bit binary number is generated, when i_c is compared to each of its neighbors along the circle (8 neighbors for window of size 3×3). The LBP code is defined as:

$$LBP_{K,R}(i_c) = \sum_{k=0}^{K-1} S(i_k - i_c) \times 2^k \tag{1}$$

$$S(v) = \begin{cases} 0 & v < t_{fix} \\ 1 & v \geq t_{fix} \end{cases} \tag{2}$$

where i_c and i_k are the intensity of the central pixel and neighboring pixel located on the circular radius R at coordinate $(x_c, y_c)$and (x_k, y_k) respectively.

Yi and Eramian [13] proposed the LBP based segmentation of defocus blur in which LBP patterns (8-bit) was reduced into only 10 patterns, out of which 9 patterns (denoted by '0' - '8') are of uniform, and all the non-uniform patterns are put into a single bin denoted by '9'. They observed that the bin 6, 7, 8 and 9 in the blurred region is prominently less than in the sharp regions. Hence, the proposed sharpness metric is,

$$M_{LBP}(x,y) = \frac{1}{N}\sum_{i=6}^{9}\left(LBP_{8,1}^{riu2}\right)_n^{(i)}(x,y) \tag{3}$$

where $\left(LBP_{8,1}^{riu2}\right)_n^{(i)}$ is the number of 8-bit LBP pattern of type i and N is the total number of pixels in the local selected region and it normalizes the map.

2.3 Blur Classification

Once the classifier $\hat{\mathbf{g}}(\mathbf{f})$ is developed, threshold prediction becomes a simple and straightforward process. The feature vector of an image is provided to the classifier, and it returns an integer value for that image. The feature vector is classified into the range of '0-8' threshold value t_{var}. This t_{var} is then used as an input to the LBP process to generate the blur map M_{LBP}. The output of the LBP is the grayscale image. The blur map M_{LBP} is now converted into binary image $M_{classified}$ with threshold T_{seg} given by,

$$T_{seg} = \frac{1}{H \times W}\sum_{(x,y)} M_{LBP}(x,y) \tag{4}$$

where H and W are the height and width of the retrieved blur map M_{LBP}.

$$M_{classified}(x,y) = \begin{cases} 0 & M_{LBP}(x,y) < T_{seg} \\ 1 & otherwise \end{cases} \tag{5}$$

Once the binary image $M_{classified}$ is computed for all of the images, the performance of our method is evaluated.

3. Results and Discussion

3.1 Experimental Setup

In our experiment, we have used publicly available dataset [6] which consists of 704 defocus partially blurred images. This dataset contains a variety of images with different magnitude of defocus blur and resolution, covering numerous attributes and scenarios like nature, vehicles, mankind, other living and non-living beings. Each image of this dataset is provided with a hand-segmented ground truth image indicating the blurred and non-blurred regions. Three widely used criteria for the evaluation of a classifier are Accuracy, Precision, and F-measure. The performance of different methods is evaluated using these three criteria.

3.2 Effectiveness of Variable Threshold

The performance of LBP based method is comparatively poor on a number of the images. The reason is that it has used a fixed threshold t_{fix} for all the images as described in Eq. 2. One such example is shown in Fig. 2. Threshold prediction based on the type of image is the only and an important difference between the proposed method and the LBP. The qualitative comparison of proposed method (t_{var}) with LBP (t_{fix}). Fig. 2a shows the qualitative comparison while Fig. 2a shows quantitative comparison. It can be seen that the proposed method outperforms LBP in terms of visual quality of classified blur maps as well as all of the metrics: Accuracy, Precision, Recall, and F-measure. Hence there is a scope for the performance improvement of LBP if variable threshold is employed.

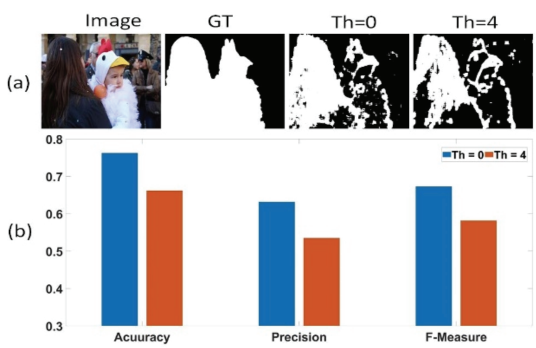

Fig. 2. Classified blur maps using a fixed $\boldsymbol{t = 4}$ and variable threshold $\boldsymbol{t_{var} = 0}$.

3.3 Comparative Analysis

In this section, our proposed method has been compared qualitatively and quantitatively with five state-of-the-art comparative methods. Those methods are *i)* LBP based segmentation of defocus blur [13], *ii)* High frequency Discrete Cosine Transform (DCT) coefficients (HiFST) [10], *iii)* Histogram Entropy (HE) [9], *iv)*Discriminative Blur Detection Features using Local Power Spectrum (LPS) [16], *v)* Discriminative Blur Detection Features using Kurtosis (LK) [6].

For the proposed method, blur map M_{LBP} is computed with a variable threshold $\boldsymbol{t_{var}}$ for each image. The sharpness map is scaled with the local window size of 15×15 pixels. Qualitative performance is also evaluated on the randomly picked images having different degree of blur. All the methods are compared with the hand-labeled ground

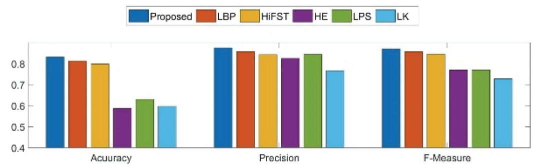

Fig. 3. Comparison of proposed method with comparative methods.

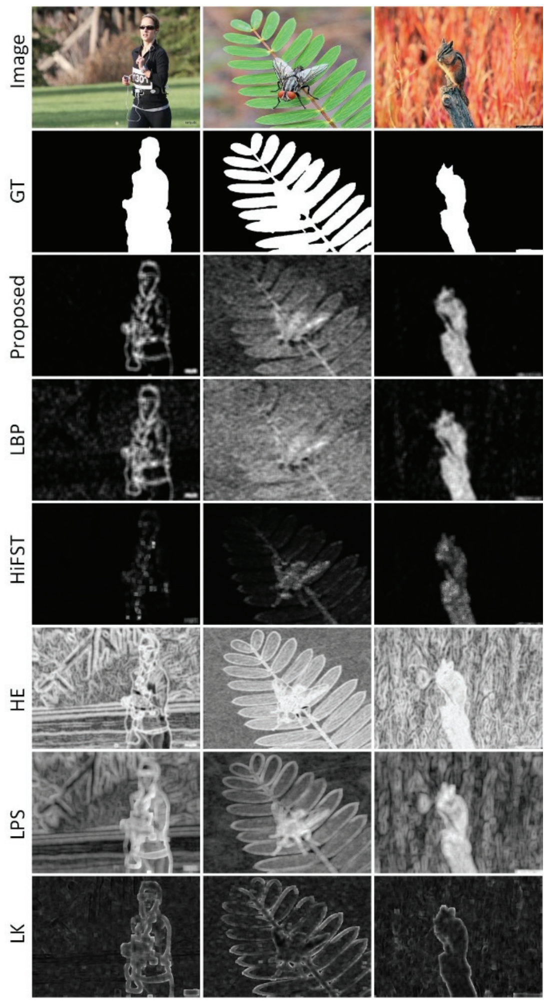

Fig. 4. Blur maps obtained by different methods.

truth. Fig. 4. shows the random images from dataset in which blur map and classified map for all the methods are computed. This figure clearly indicate that our proposed method can segment the blurred and unblurred regions with high accuracy regardless of the blur type and scenario. Hence the performance of our method is better as compared to the LBP with a fixed threshold and other comparative methods.

4. Conclusion

In this article, we have proposed an adaptive threshold-based method to improve the performance of the LBP method in blur detection. First, we trained a model using SVM which can predict the threshold based on the image features, and then respective thresholds are used to acquire the sharpness map of the images using LBP. We have evaluated the performance of the proposed method in terms of accuracy, precision and F-measure using a well-known dataset. The results show the effectiveness of the method to achieve good performance over a wide range of images, and it outperforms the state-of-the-art defocus segmentation methods.

Acknowledgement

This work was supported by Korea University of Technology and Education (KOREATECH) Education and Research Promotion Program (2020).

References

1. K. Bahrami, A. C. Kot, and J. Fan, “A novel approach for partial blur detection and segmentation,” in *Multimedia and Expo (ICME), 2013 IEEE International Conference on*, 2013, pp. 1-6.
2. P. Jiang, H. Ling, J. Yu, and J. Peng, “Salient region detection by ufo: Uniqueness, focusness and objectness,” in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 1976-1983.
3. K. G. Derpanis, M. Lecce, K. Daniilidis, and R. P. Wildes, “Dynamic scene understanding: The role of orientation features in space and time in scene classification,” in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 2012, pp. 1306-1313.
4. J. Shi, L. Xu, and J. Jia, “Just noticeable defocus blur detection and estimation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 657-665.
5. M. T. Mahmood, S. A. Siddiqui and Y. K. Choi, “Blur Detection through Multinomial Logistic Regression based Adaptive Threshold,” Journal of the Semiconductor & Display Technology, vol. 18, no. 4, pp. 1-6, 2019.
6. J. Shi, L. Xu, and J. Jia, “Discriminative blur detection features,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2965-2972.
7. S. Zhuo and T. Sim, “Defocus map estimation from a single image,” *Pattern Recognition,* vol. 44, pp. 1852-1858, 2011.
8. U. Ali and M. T. Mahmood, “Analysis of blur measure operators for single image blur segmentation,” *Applied Sciences,* vol. 8, p. 807, 2018.
9. E. Krotkov and J.-P. Martin, “Range from focus,” in *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, 1986, pp. 1093-1098.
10. S. A. Golestaneh and L. J. Karam, “Spatially-Varying Blur Detection Based on Multiscale Fused and Sorted Transform Coefficients of Gradient Magnitudes,” in *CVPR*, 2017, pp. 596-605.
11. M. T. Mahmood, U. Ali, and Y. K. Choi, “Single image defocus blur segmentation using Local Ternary Pattern,” *ICT Express,* 2019.
12. C. Rother, V. Kolmogorov, and A. Blake, “Grabcut: Interactive foreground extraction using iterated graph cuts,” in *ACM transactions on graphics (TOG)*, 2004, pp. 309-314.
13. X. Yi and M. Eramian, “LBP-based segmentation of defocus blur,” *IEEE transactions on image processing,* vol. 25, pp. 1626-1638, 2016.
14. E. L. Allwein, R. E. Schapire, and Y. Singer, “Reducing multiclass to binary: A unifying approach for margin classifiers,” *Journal of machine learning research,* vol. 1, pp. 113-141, 2000.
15. T. Ojala, i. Pietik\"a, Matti, and D. Harwood, “A comparative study of texture measures with classification based on featured distributions,” *Pattern recognition,* vol. 29, pp. 51-59, 1996.
16. R. Liu, Z. Li, and J. Jia, “Image partial blur detection and classification,” in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 2008, pp. 1-8.

접수일: 2020년 7월 7일, 심사일: 2020년 8월 18일,
게재확정일: 2020년 8월 18일