[[1]](#footnote-1)

Blur Detection and Classification using Local Binary Pattern and Hough Line Detection

Term paper - Computer Vision 2020-1

Minjoo Park, ID: 1931036004, cathymjp@koreatech.ac.kr

*Abstract*— Not all images taken are blur-free. The reason for those blurs are diverse. Most of the times, blurs are thought to be a mistake of an image, but some blurs are intentionally caused to give an artistic look. Most of the recent methods researchers consider are performed using machine learning. Researchers come up with better machine learning architecture to detect blurs. In this term paper, however, traditional method of detecting blur is going to be used. The blur map is going to be achieved by calculating the Local Binary Pattern (LBP) sharpness of a single image. Then, Hough line detection is applied to the blurred image is going to be classified according to the type of its blur; motion blur, or lens-out-of-focus.

# INTRODUCTION

Blurred images are caused by object motion, lens out-of-focus, and camera shake. These are thought to be an error in a photo. However, some photos intentionally blur some areas to make the other object stand out, or to give an artistic look. In order to enhance the blurred image, it is necessary to know where the blur is in an image and whether the blur is intentional or not. Some of the fields blur detection plays role are image segmentation, depth estimation, blur magnification, image de-blurring, and image refocusing.

There are various algorithms for blur detection and segmentation. Methods suggested for blur detection can be divided into two broad categories. First approach is using traditional hand-crafted method. For hand-crafted methods, Xu et al [6], Pertuz et al [7], Xiao et al. [9], and Liu et al [1] suggested different methods for blur detection without the use of a network.

Second approach, which is currently arising and the main target of this field, is using machine learning to detect blurs. Huang et al. [8], Tang et al. [10], and Zhao et al. [11] uses networks to detect blurs in an image. As using network is more complex and detailed compared to traditional hand-crafted methods, the results are highly promising.

There are limited number of research papers released on blur classification. Liu et al, [1] uses local autocorrelation congruency, and Su et al. [2] uses certain alpha channel constraints to classify its blur type. Fan et al. [3], Yang et al [4], and Zhang et al [5] uses machine learning or dictionaries to classify the blur type. Although the results are promising, the results are not clear for traditional hand-crafted methods. Compared to the number of papers released on blur detection, blur classification is less studied and less focused when discussing blur detection. For this final term project, traditional hand-crafted method is going to be used.

The current methods for blur detection and classification relies on using machine learning. Therefore, these approaches are not appropriate for the final project as the aim for this term project is to use traditional hand-crafted method. The most challenging part of the blur detection using traditional hand-crafted method is setting up the initial state. A few methods will be combined to set a more accurate foundation for blur detection. Also, most of the papers currently available for blur detection does not classify the blur type. It focuses more on restoring the blurred image instead of detecting the blur and segmenting it at once. In this project, a simple yet promising approach using Hough line detection is going to be used to classify the image into its blur category.

The method proposed in this final term projectuses Local Binary Pattern (LBP) discussed in Yi el at. [12] as the base for blur detection and generating the blur map. LBP is used in fields such as texture segmentation, face recognition, background subtraction, and 3D surface recognition. This is one of the traditional hand-crafted method proposed. The sharpness metric presented in the paper calculates the local binary patterns since areas with blurs have fewer local binary patterns. 8-bit LBP is sued. This is for rotation of the image in 8 directions and 1 in the center. Then all of the frequency of patterns in these regions are considered to see if the pattern frequency is less or not. With these difference in max and min sharpness values, sharpness map is calculated and are finally concatenated to produce the final blur response map.

For blur classification, Hough line detection on the inverse of threshold image is applied. Motion blurs have multiple lines in the direction of its movement. Just applying the Hough line detection on inverse of threshold will give all of the lines detected. These unnecessary lines should not be counted when detecting the motion blur. Therefore, minimum number of detected edges and maximum line gap is set to consider multiple lines at the same direction. As the motion blurs can appear on multiple degree of angle, the angle information from the Hough line detection is used to get multiple lines in an equivalent angle.

# Related work

First of all, second part of the blur detection method is done using machine learning. Since a novel algorithm is used for the network, the results are promising. Huang et al. [8] uses deep 6-layer CNN model. Multiscale blur maps are fused to get better blur detection result. For this method, it tends to work well when the blur is placed as the background instead of the object in the front. Tang et al. [10] proposed DeFusionNET. It effectively suppresses the background cluster by fusing and refining multi-scale deep features. Feature fusing and refining module is also proposed which refine features of different layers cross-layer manner. It is effective at capturing the complementary information from both the shallow features and deep semantic features. Zhao et al. [11] made the first attempt to develop an end-to-end deep network for defocus blur detection. It introduced BTBNet for integrating low-level cues and high-level semantic information. By handing images with different sizes, it looks at magnified regions which could be handling images with low resolution from the beginning.

Traditional hand-crafted methods are less accurate but still give promising results. Xu et al. [6] is effective for getting a defocus map with a single image. It does not require well separated image edges, and in focus regions. With ranks of local paths and defocus amount, edge point defocus amount can be estimated. The defocus blur in this algorithm is estimated only at the edge locations which could lead to less accurate result. Pertuz el al. [7], uses only the implicit information about the scene geometry. Parameters of the lens are not required for the focusing process as well. Finding the difference between the out-of-focus pixel and originally flatted regions or smoothed edges are a challenge. Xiao et al. [9] uses Point Spread Functions (PSF). It does not require any use of blur kernel, degree, or camera parameters. It gives the blur degree at each location in an image. Multiscale singular value concept is used to make up for the information ungiven. Some of the thresholds are over-estimated and are not perfectly accurate. Lastly, Liu et al. [1] region-wise training and classification is based on image patches, making the process in one image more efficient. One drawback from this algorithm is that if an image has complex feature, it fails to work efficiently.

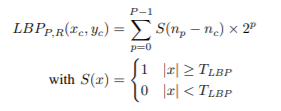
Yi el al. [12] proposed sharpness metric calculation. The paper achieved defocus blur segmentation using LBP patterns in blurred and unblurred image regions. Blurry regions have significantly less local binary patterns compared to the regions with sharp details. As stated above, rotation invariant is done by performing the circular bitwise right shift of which it minimizes the value of LBP when interpreted as a binary. The method proposed for the final term project builds on this algorithm and improve the initial setting of the image.

# Proposed Method

The paper referenced for this final term paper is *LBP-Based Segmentation on Defocus Blur [12].* Most of the algorithms are referenced from this research paper and mathematical methods are also from this algorithm.

Initial processing of the image before calculating the LBP involves applying Gaussian filter and Laplace function. Noise is removed by blurring the image with Gaussian filter. (3,3) kernel is applied and after blurring the image, convert the image the grayscale. Then, Laplacian filter is applied with depth of CV\_16S, and kernel size of 3. This processing is to reduce the noise and to amplify the difference of the blurred and non-blurred region.

Difference of the sharpness map is calculated by applying the LBP. The equivalent equation (1) proposed in [12] is as the following.

 (1)

nc is the intensity of the central pixel (xc, yc), and np is the intensities of the P neiboring pixels of the radius R of a circle cnetered at nc. TLBP indicates the positive threshold to achieve robustness for flat image regions. P in this case was chosen as 8 and R is 1. The locations of 8 Ps are right, left, above, below, aboveRight, aboveLeft, belowRight, and belowLeft. Figure 1 displayed below shows the locations of the eight neighboring pixels np for P = 8 and R = 1.

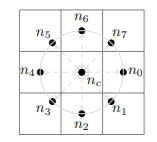


Fig. 1 Location of 8-bit LBP

Bilinear interpolation is applied to obtain the intensity of np as the it does not fall in the center of image pixels. All eight interpolated values are calculated for pixel (i, j) of the original value. Next step is to calculate the uniformity and correct pattern id by adding all eight values. This value should be uniform but if the value is not uniform, value of 9 is replaced for bit count to indicate that it is a non-uniform pattern.

Once the LBP sharpness are calculated, sharpness map is calculated with patch size of 21 x 21 pixels and threshold TLBP of 0.016. The equation for calculating the sharpness map is in equation (2) where sharpnessMapm is minimum of the sharpness map calculated and sharpnessMapM is the maximum value of the sharpness map calculated.

sharpness Map = (2)

With the equation, multi-scale sharpness map generation is complete. The sharpness metric is computed at each image pixel and are constructed at total of three times.

For blur classificaiton, Hough line detection was performed on the inverse threhold image. As motion blur images contain multiple lines in an image in a single direction, applying the Hough line detection is able to collect those lines. Inverse threhold was applied to the image since line detection is performed on black lines and regular threshold colors blurred areas as black. Just applying the Hough line transform is not enough to detect motion blur. Motion blurs have multiple lines going on one direction. Therefore, minimum length of the detected line and minimum number of the lines in a single direction is also calculated. using the Hough line transform, the angle of the line can be achieved. Stack the information of all the lines in an array and check the angle and length of each lines.

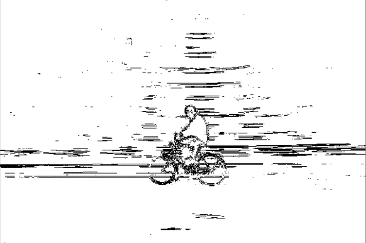
 

Fig. 2: Motion blur

Figure 2 above shows the result of Hough line detection. As the picture shows, there are multiple lines going in one direction. By calculating the angle and length of each line detected, the image is classified to motion blur. If there are not enough line or lines of different angles, it is clasffied as lens out-of-focus blur type.

# Results and Discussion

The final term project was done using python. Total of ten images were tested. Out of ten images, three images were motion blur and seven images were lens out-of-focus.

For LBP calculation, P was set to 8 and R was set to 1 to get the 8-bit location. The values used for sharpness map are 21 x 21 pixel for patch size and 0.016 for threshold. Total of three sharpness metric was performed. For preprocessing stage, 3 x 3 kernel size was used for Gaussian blur and depth of CV\_16S and kernel size of 3 for Laplacian function.





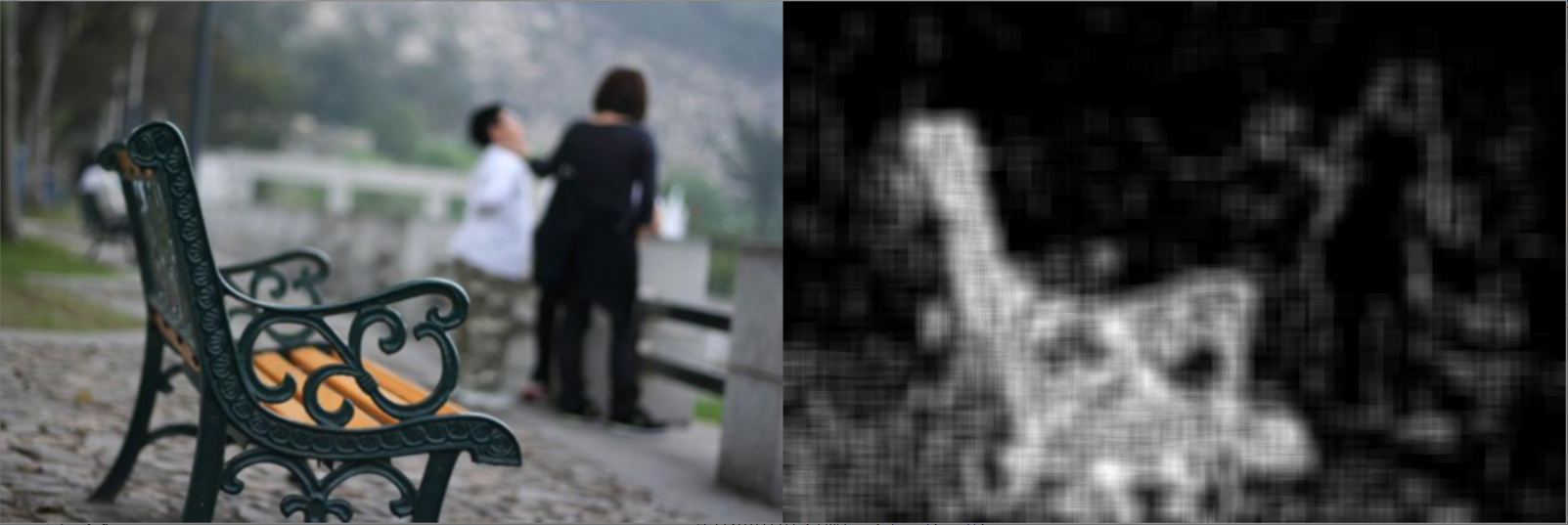




Fig 3: Blur Map

Figure 4 is the comparision between the original result for [12] and result applying the Gasussian filter and Laplacian filter. The one below has some more extra lines but is more clear for areas that have focus.





Fig 4: Comparison between [12] and proposed method

Figure 5 below is the inverse threhold of an image lens out-of-focus on the left and motion blur on the right. The results shows that for lens out-of-focus blur, there are not visible lines in a single direction as compared to the image on the right for motion blur.

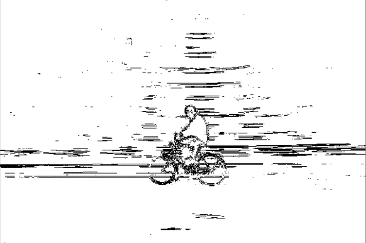
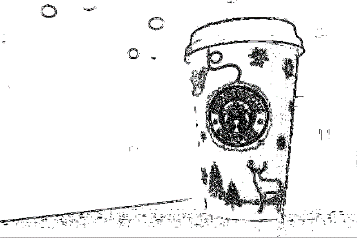


Fig 5: Inverse threshold

# Conclusion

For this final term project, I presented an approach for detecting blur and classifying it by detecting its edges using traditional hand-crafted method. Although I was struggled at coming up with a new method for blur detection and complete result for blur classification, I was able to present a novel approach for blur detection and classification which could be used in future. By combining different methods presented in different research papers, I was able to improve the accuracy. With the experience I learned about blur detection and classification for this final project, I will be able to enlarge my vision in the field of computer vision.

References

1. Renting Liu, Zhaorong Li and Jiaya Jia, "Image partial blur detection and classification," 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, 2008, pp. 1-8, doi: 10.1109/CVPR.2008.4587465.
2. B. Su, S. Lu, and C. L. Tan. Blurred image region detection

and classification. In *19th ACM international conference on Multimedia*, pages 1397–1400, 2011.

1. Fan, Mingyuan & Huang, Rui & Feng, Wei & Sun, Jizhou. (2017). Image Blur Classification and Blur Usefulness Assessment. 10.1109/ICMEW.2017.8026291..
2. Dong Yang and Shiyin Qin. 2016. Restoration of partial blurred image based on blur detection and classification. JECE 2016, Article 1 (January 2016), 1 pages. DOI:https://doi.org/10.1155/2016/2374926
3. Zhang, Shanghang et al. “Learning to Understand Image Blur.” 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (2018): 6586-6595.
4. G. Xu, Y. Quan and H. Ji, "Estimating Defocus Blur via Rank of Local Patches," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, 2017, pp. 5381-5389
5. Said Pertuz, Miguel Angel Garcia, Domenec Puig, Focus-aided scene segmentation, Computer Vision and Image Understanding, Volume 133, 2015, Pages 66-75,
6. Rui Huang, Wei Feng, Mingyuan Fan, Liang Wan, Jizhou Sun, Multiscale blur detection by learning discriminative deep features, Neurocomputing, Volume 285, 2018, Pages 154-166,
7. Huimei Xiao, Wei Lu, Ruipeng Li, Nan Zhong, Yuileong Yeung, Junjia Chen, Fei Xue, Wei Sun, Defocus blur detection based on multiscale SVD fusion in gradient domain, Journal of Visual Communication and Image Representation, Volume 59, 2019, Pages 52-61,
8. C. Tang, X. Zhu, X. Liu, L. Wang and A. Zomaya, "DeFusionNET: Defocus Blur Detection via Recurrently Fusing and Refining Multi-Scale Deep Features," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 2695-2704.
9. W. Zhao, F. Zhao, D. Wang and H. Lu, "Defocus Blur Detection via Multi-stream Bottom-Top-Bottom Fully Convolutional Network," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018, pp. 3080-3088.
10. X. Yi and M. Eramian, "LBP-Based Segmentation of Defocus Blur," in IEEE Transactions on Image Processing, vol. 25, no. 4, pp. 1626-1638, April 2016, doi: 10.1109/TIP.2016.2528042.

1. [↑](#footnote-ref-1)