

Image Partial Blur Detection and Classification*

Renting Liu Zhaorong Li Jiaya Jia
Department of Computer Science and Engineering
The Chinese University of Hong Kong
{rtlou, zrli, leojia}@cse.cuhk.edu.hk

Abstract

In this paper, we propose a partially-blurred-image classification and analysis framework for automatically detecting images containing blurred regions and recognizing the blur types for those regions without needing to perform blur kernel estimation and image deblurring. We develop several blur features modeled by image color, gradient, and spectrum information, and use feature parameter training to robustly classify blurred images. Our blur detection is based on image patches, making region-wise training and classification in one image efficient. Extensive experiments show that our method works satisfactorily on challenging image data, which establishes a technical foundation for solving several computer vision problems, such as motion analysis and image restoration, using the blur information.

1. Introduction

In this paper, we focus on detecting and analyzing partially blurred images and propose a novel method to automatically detect blurred images, extract possible blurred regions, and further classify them into two categories, i.e., near-isotropic blur and directional motion blur.

Our method attempts to tackle two major problems. One is blur detection with simultaneous extraction of blurred regions. The result in this step provides useful high-level regional information, facilitating a variety of region-based image applications, such as content-based image retrieval, object-based image compression, video object extraction, image enhancement, and image segmentation. It can also serve as one of the criteria of measuring the quality of captured images.

The second objective of our method is to automatically classify the detected blur regions into two types: near-isotropic blur (including out-of-focus blur) and directional



(a) (b)
Figure 1. Two image examples with (a) motion blurred regions and (b) out-of-focus blurred regions.

motion blur. We classify image blur into these two classes because they are most commonly studied in image restoration. The blur classified images also easily find applications in motion analysis and image restoration. Two partial-blur-image examples are illustrated in Figure 1.

Although the topics of image blur analysis have attracted much attention in recent years, most previous work focuses on solving the deblurring problem. General blur detection, on the contrary, is seldom explored and is still far from practical. Rugna *et al.* [5] introduced a learning method to classify blurry or non-blurry regions in one input image. This method is based on an observation that blurry regions are more invariant to low pass filtering. In our experiments, we find that only using this information is not sufficient for designing a reliable classifier. Different blur measures should be combined in order to achieve high-quality blur detection. This method also does not distinguish blur types.

Utilizing the statistics of gradient information along different directions, the method in [18] builds an energy function based on the inferred blur kernel in order to segment image into blur/nonblur layers. This method only discovers motion blurred regions by inferring directional blur kernels. Other blur estimation methods, such as the one proposed in [7], only provide a measure of blur extent, which cannot be directly used to discriminate blurry against non-blurry regions.

In this paper, we present a new blur detection and analysis method for automatically extracting blurry regions by combining specifically designed blur features represented

*The work described in this paper was fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. 412307).

by spectral, gradient, and color information, respectively. Then we use the direction information of local autocorrelation function, which evaluates how well a local window in a blur region matches a spatially-shifted version of itself, to further discriminate the type of blur. In our blur classification, no blind deconvolution, which may involve complex kernel estimation, is performed.

Our features are extracted in local regions using the color-structure information relative to that collected globally in each image. This process is based on a key thought: if we directly use the image color information for each local region, it is hard to define a threshold for classification due to the variation of natural image structures. We, thus, take account of the relative information between the patch and the image, which makes our feature analysis reliable in regard to the color-structure diversity in natural images. Extensive experiments show that our method works satisfactorily with challenging image data.

2. Related Work and Overview of Our System

We first review related work of blur detection, image deblurring, and low Depth of Field (DoF) image auto-segmentation, and then give an overview of our system.

2.1. Previous Work in Blur Analysis

For blur detection and estimation, previous approaches aim at measuring blur extent of edges and are based on the analysis of edge sharpness [20]. Chuang *et al.* [4] extended this idea by first fitting gradient magnitude along edge direction to a normal distribution. Then the standard deviation of this distribution, together with gradient magnitude, are regarded as the blur measure. Elder and Zucker [7] modeled focal blur by a Gaussian blur kernel and calculated the response using the first and second order derivative steerable Gaussian basis filters [11]. Therefore, focal blur is estimated by the thickness of object contours. Zhang and Bergholm [26] defined Gaussian Difference Signature, which functions similarly to the first-order derivative of Gaussian, in order to measure the diffuseness caused by out-of-focus objects. Note that all these methods assume that the Point Spread Function (PSF) is modeled by a Gaussian blur filter. They cannot be applied to detecting ubiquitous non-Gaussian blur.

Blind image deconvolution [8, 14] aims to estimate the blur filter and latent unblurred images. It is a severely ill-posed problem. Although recently many methods have been proposed in image deblurring, most of them only tackle spatially-invariant blur, i.e., all pixels in the input image are blurred by the same PSF. Some methods [1, 18, 22] were proposed to tackle the partial blur problem with the help of user interaction or blur kernel assumption. For all these methods, if the PSF can be correctly reconstructed, the type

of blur is also known using the structure of PSF. However, in practice, blind deconvolution usually performs unsatisfactorily even by making restrictive assumptions on image and kernel structures. It does not handle well our partially-blurred images. Besides, a visually plausible deconvolution result does not imply that the PSF is correctly estimated. These factors make blind deconvolution not a good choice for general blur detection in terms of efficiency and accuracy, especially for handling images in a large database.

Another type of blur analysis is Low Depth of Field (DoF) image auto-segmentation. Low DoF is a photography technique which abstracts the photographer’s intention by giving a clear focus only on an Object of Interest (OOI). Previous methods for automatic OOI extraction [15, 17, 24] are not suitable for our blur detection because they only work on low DoF input images containing out-of-focus background. In [6], Low DoF images are detected by calculating a *low DoF indicator*, defined by the ratio of wavelet coefficients in high-frequency of the central regions of the whole image. This method simply assumes that low DoF images contain focused object near the center and surrounding pixels are out of focus. This method also does not suit our general-purpose blur detection.

2.2. System Overview

Due to the diversity of natural images, in this paper, we propose a learning framework with a training process to detect partially blurred images. These images are further classified into two blur types.

In our system, blur detection and blur type classification are achieved in two steps. First, detection of blurred images is performed. In this step, we propose to use a combination of three features, namely, *Local Power Spectrum Slope*, *Gradient Histogram Span*, and *Maximum Saturation*, to model the blur characteristics in different ways. Second, directional motion blurred regions are distinguished from out-of-focus blurred regions by using another feature, i.e., *Local Autocorrelation Congruency*.

Note that automatic blur detection without really estimating the blur kernels is not straightforward. In our system, besides introducing spectral and visual clues, we also include analysis of the characteristics of the two blurs and design corresponding features.

3. Blur Features

There are four different features developed and combined in our system. These features are derived by analyzing the visual and spectral clues from images.

- *Local Power Spectrum Slope*. Due to the low-pass-filtering characteristic of a blurred region, some high frequency components are lost. So the amplitude spec-

trum slope of a blurred region tends to be steeper than that of an unblurred region.

- *Gradient Histogram Span.* The distribution of gradient magnitude serves as an important visual clue in blur detection. Blurred regions rarely contain sharp edges, which results in small gradient magnitude. Accordingly, the distributions of the log gradient magnitude for blurred regions should have shorter tails than that for other regions.
- *Maximum Saturation.* Unblurred regions are likely to have more vivid colors than blur regions. The maximum value of saturation in blurred regions is correspondingly expected to be smaller than that in unblurred regions.
- *Local Autocorrelation Congruency.* If a region is blurred by relative motion between an object and the background in a certain direction, all edges of the object will be blurred, except those sharing the same direction with the motion. This is regarded as another important visual clue in our blur analysis.

In order to handle partial blur, our approach adopts a region-based feature extraction. Specifically, we partition the input image into blocks and analyze features in each of them. We show in the following subsections that this local representation provides reliable blur measures.

In our blur description, flat regions, i.e., the blocks containing absolutely no edge structures, are ambiguous since they can be interpreted as either blur or non-blur. So, in our system, these ambiguous regions are automatically removed beforehand. This makes our blur detection reliable for many natural images containing, for example, cloudless sky or textureless walls.

3.1. Local Power Spectrum Slope

We first compute the power spectrum of an image I with size $N \times N$ by taking the squared magnitude after Discrete Fourier transform (DFT)

$$S(u, v) = \frac{1}{N^2} |I(u, v)|^2, \quad (1)$$

where $I(u, v)$ denotes the Fourier transformed image. We then represent the two-dimensional frequency in polar coordinates, i.e., $u = f \cos \theta$ and $v = f \sin \theta$, and construct $S(f, \theta)$. According to [3, 9], by summing the power spectra S over all directions θ , $S(f)$, using polar coordinates, can be approximated by

$$S(f) = \sum_{\theta} S(f, \theta) \simeq A/f^{-\alpha}, \quad (2)$$

where A is an amplitude scaling factor for each orientation and α is the frequency exponent, called *slope of power spectrum*.

A number of studies [3, 9, 23] have demonstrated that, using image-wise computation, the power spectra of most natural images make $\alpha \approx 2$. A blurred image usually has a large α [2, 10] while an unblurred image, contrarily, corresponds to a small α . However, a natural image may contain objects with different boundary sharpness. Our experiments show that it is not reliable to simply set a threshold to α for blur estimation since the value of α varies in different images in a pretty wide range.

In [12], α discrimination experiments showed that human has sufficient ability to discriminate changes in α even using very small image patches. This indicates that the blur is not determined by the absolute value of α over the whole image, but the relative local-to-global α differences.

Based on the above observation, we introduce the local and relative blur estimation using the slope of power spectrum α . We first compute α_o , the global measure of the slope of power spectrum for the whole image. Then we compare α_p computed in each local block p with α_o . If α_p is much larger than α_o , it is quite possible that this block is blurred. Our metric q_1 , accordingly, is given by

$$q_1 = \frac{\alpha_p - \alpha_o}{\alpha_o}. \quad (3)$$

We show in Fig. 2 the computed local α_p with patch size 17×17 . Pixels with warmer color indicate higher blur possibility. The α value distribution coincides with human visual perception.

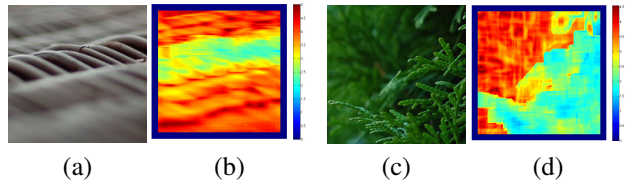


Figure 2. Local α_p illustration. α_p is calculated in patches with size 17×17 in these examples. (a) and (c) show two natural images. (b) and (d) illustrate the corresponding local α_p maps where warmer color pixels correspond to pixels with larger α_p values.

3.2. Gradient Histogram Span

Recent research in natural image modeling [21] has shown that the gradient magnitudes of natural images usually follow a heavy-tailed distribution, which means natural images are with primarily small or zero gradient magnitudes. However, in a blurred image, since blurred regions usually do not contain too many sharp edges, the gradient magnitude distribution should have much of its mass with small or zero values.

For the two input images in Fig. 3(a) and (b), the gradient magnitude distributions for two blurred patches shown in Fig. 3(c) and (d) exhibit no apparent tail while those for two unblurred regions contain heavy tails as shown in Fig. 3(e) and (f). We choose to fit the local gradient magnitude distribution by a mixture of two-component Gaussian model using Expectation-Maximization:

$$\pi_0 G(x; u_0, \sigma_0) + \pi_1 G(x; u_1, \sigma_1) \quad (4)$$

with mean $u_0 = u_1 = 0$ and variance $\sigma_1 > \sigma_0$. The components of the fitted Gaussian mixture models are illustrated as red curves in Fig. 3. The Gaussian component with larger variance σ_1 is mainly responsible for causing the heavy tail in the original distribution.

However, the value of σ_1 alone is not suitable to be used directly for blur detection. Different images containing unblurred objects may also have different levels of edge sharpness, making σ_1 variant. So we also adopt relative measurement and combine σ_1 with the structure contrast inside each patch. Specifically, we give large blur confidence to patches with significant structure contrast but relatively small gradient magnitude, which are most likely to be blurred. The local contrast is calculated by

$$C = \frac{L_{max} - L_{min}}{L_{max} + L_{min}}, \quad (5)$$

where L_{max} and L_{min} denote the maximum and minimum intensity values of all pixels in a region. We set the region size as 10×10 in our experiments, smaller than the size of a patch. Our metric q_2 for patch p is defined as

$$q_2 = \frac{\tau \sigma_1}{C_p + \epsilon^2}, \quad (6)$$

where C_p is calculated as the maximum local contrast C over all structural regions inside patch p , ϵ is a small value to prevent dividing by zero. Using our definition, q_2 will have small response when a patch has large C_p but small σ_1 . Parameter τ is set to 25 in experiments to balance the scales of σ_1 and C_p .

3.3. Maximum Saturation

We also take the color information into blur detection. It is observed that blurred pixels tend to have less vivid colors than unblurred pixels because of the smoothing effect of the blurring process. So we first compute pixel saturation by

$$S_p = 1 - \frac{3}{(R + G + B)} [\min(R, G, B)]. \quad (7)$$

Then, within each patch p , we compute saturation S_p for each pixel and find the maximum value $\max(S_p)$. It will be compared with $\max(S_o)$, the maximum saturation value of the whole image. Our metric q_3 is defined as

$$q_3 = \frac{\max(S_p) - \max(S_o)}{\max(S_o)}. \quad (8)$$

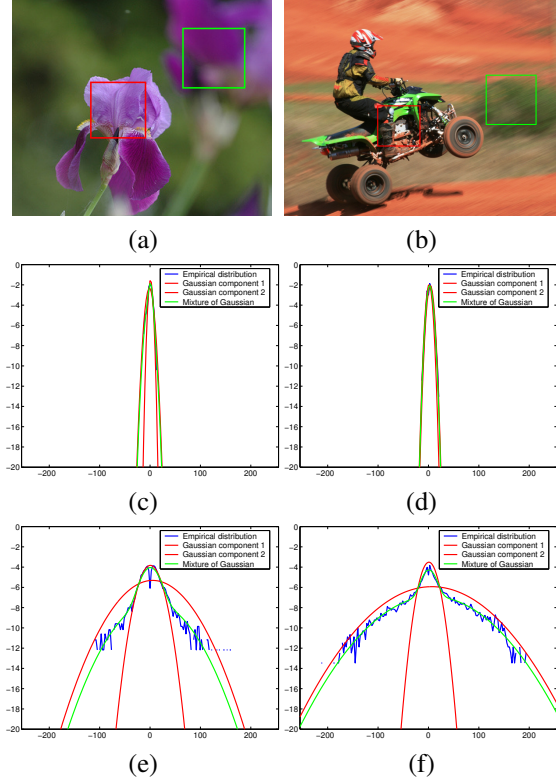


Figure 3. The distributions of gradient magnitudes and their approximations by mixture of Gaussians in patches. (a) One focal blurred image with two patches containing blurred and unblurred pixels respectively. (b) One motion blurred image with two patches selected similarly. (c) and (e) show the respective distributions of gradient magnitudes of the two patches in (a). The original distributions are shown in blue whereas the two Gaussian components are illustrated using red curves. (d) and (f) show the respective distributions of gradient magnitudes of the two regions in (b).

3.4. Local Autocorrelation Congruency

The local autocorrelation function [25] is a measure of how well a signal matches a time-shifted version of itself. As for an image, it can be interpreted as a measure of how well a local window in a blurred region matches a spatially-shifted version of itself. In our method, we use the directional information of local autocorrelation function to further discriminate the type of blurs inside those detected blur regions.

The image power spectrum is related to the autocorrelation function through the Wiener-Khinchin theorem, which states that the auto-correlation function and the power spectrum form a Fourier transform pair [25]. So besides the analysis of local power spectrum slope in defining feature q_1 , we further analyze the local autocorrelation characteristics of blurred regions.

The local autocorrelation function at a point (x, y) under

a shift $(\Delta x, \Delta y)$ in image space can be calculated by

$$f(x, y) = \sum_{(x_k, y_k) \in W} [I(x_k, y_k) - I(x_k + \Delta x, y_k + \Delta y)]^2,$$

where W is a local window centered at point (x, y) . Using first-order Taylor expansion to approximate $I(x_k + \Delta x, y_k + \Delta y)$, $f(x, y)$ can be represented using a local autocorrelation matrix M :

$$f(x, y) \simeq \sum_{(x_k, y_k) \in W} \left\{ [I_x(x_k, y_k), I_y(x_k, y_k)] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right\}^2$$

$$= [\Delta x, \Delta y] M \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}, \text{ where}$$

$$M = \sum_{(x_k, y_k) \in W} \begin{bmatrix} I_x^2(x_k, y_k) & I_x(x_k, y_k)I_y(x_k, y_k) \\ I_x(x_k, y_k)I_y(x_k, y_k) & I_y^2(x_k, y_k) \end{bmatrix}.$$

When taking a close look at the blurring process, one can notice that the color of one pixel is spread to its neighborhood after blurring, increasing its color similarity to its neighboring pixels. Besides, the color spreading strategy of each pixel is defined by the PSF. So the local autocorrelation function in blurred images, which depends on $(\Delta x, \Delta y)$, should also be strongly related to the shape and value of PSF.

In fact, the Harris corner descriptor [13] is closely related to local autocorrelation function by adding a window function when calculating matrix M . If the window function is binary and rectangular, the two representations are actually identical. So we adopt the idea of Harris corner detector and calculate matrix M with a Gaussian window function to suppress noise.

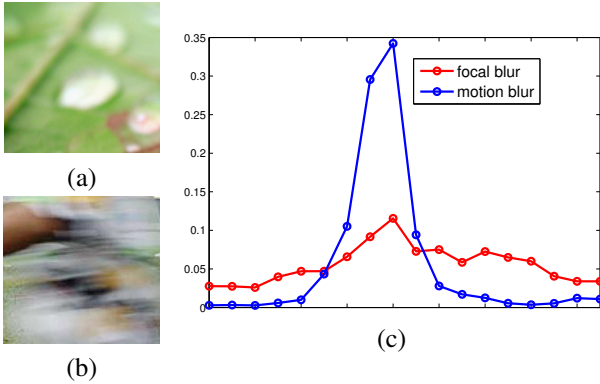


Figure 4. The histograms of weighted eigenvector directions for two sample patches. (a) A focal blurred patch. (b) A motion blurred patch. (c) Histograms of weighted eigenvector directions of (a) and (b). The blue histogram is for the focal blurred patch and the red histogram is for the motion blurred patch. Directional motion blur produces distinctive peak value.

We compute the eigenvalues $\lambda_1(x, y)$ and $\lambda_2(x, y)$ of M for each pixel (x, y) inside an image patch p , where

$\lambda_1(x, y) > \lambda_2(x, y)$, and denote their corresponding eigenvectors as $V_1(x, y)$ and $V_2(x, y)$. The values of the two eigenvalues are proportional to the principal curvatures of the local autocorrelation function and form a rotationally invariant description of Matrix M . Two classes can be produced by our blur type analysis:

1. If image patch p is mostly directional-motion blurred, the edges with gradient perpendicular to the blur direction will not be smoothed, leaving strong gradient only along one direction.

2. If the image patch is approximately focal blurred, the gradient magnitudes along all directions are attenuated.

To be more specific, the eigenvector $V_2(x, y)$, which corresponds to the smaller eigenvalue $\lambda_2(x, y)$, represents the major axis direction of the ellipse-shaped contour of a local autocorrelation function at point (x, y) . We thus construct a directional response histogram $hist(\theta)$ for patch p , where each bin represents one direction θ and the value of each bin is the number of pixels with eigenvector V_2 along direction θ in patch p , weighted by their ellipse axis ratio $\sqrt{\lambda_1/\lambda_2}$. One illustration is shown in Figure 4. In order to use the information of the histogram, we also normalize the bin values by dividing the total pixel number in one patch. Then our Local Autocorrelation Congruency measure is to check whether the histogram in this region has a distinctive peak. In our method, we compute the variance of the normalized bin values as our feature of Local Autocorrelation Congruency, i.e.

$$q_4 = \text{Var}\{hist(\theta)\}. \quad (9)$$

4. Classification

Using the defined blur features, our system consists of two steps. In the first step, we train a blur/nonblur classifier to discriminate different regions. This classifier is based on features proposed in Sections 3.1, 3.2, and 3.3. Then, in the second step, the detected blurry regions are measured by Local Autocorrelation Congruency in order to recognize the blur types.

Given the list of features for discriminating unblurred and blurred regions, we apply the Bayes classifier:

$$\eta_a = \frac{P(Blur | q_1, q_2, q_3)}{P(Sharp | q_1, q_2, q_3)}$$

$$= \frac{P(q_1, q_2, q_3 | Blur) P(Blur)}{P(q_1, q_2, q_3 | Sharp) P(Sharp)}, \quad (10)$$

where $P(Sharp | q_1, q_2, q_3)$, by the Bayes' rule, denotes the probability of labeling a region as "nonblur" given the defined features. Assuming the independence of different features, we simplify (10) to

$$\eta_a = \frac{P(q_1 | Blur) \dots P(q_3 | Blur)}{P(q_1 | Sharp) \dots P(q_3 | Sharp)}. \quad (11)$$

Then each conditional probability above is trained using the training image set.

In the second step, we further classify blurred regions into “motion blur” and “focal blur”, using our feature defined in Section 3.4. Conditional probabilities $P(q_4 | focal\ blur)$ and $P(q_4 | motion\ blur)$ are trained, and the blur type classification is achieved by computing $\eta_b = P(q_4 | focal\ blur) / P(q_4 | motion\ blur)$.

5. Experiments and Results

We describe our experiments in this section. In the first part, we construct training and testing datasets with manually labeled patches, and use them to test the accuracy of our patch classification algorithm. We also apply our algorithm to ranking the confidence of blur for an image. In the second part, we experiment with partially blurred image segmentation.

5.1. Blur Patch Detection

We collect totally 100 partially blurred images and 100 unblurred images from photo sharing websites, such as Flickr.com and PBase.com, to form our dataset. In each category, half of the images are used for training and the other half are for testing. All the images are manually segmented into square patches. The size of each patch ranges from 50×50 to 500×500 pixels, which occupies 5% ~ 20% of the size of the original images. Examples of images and patches in our datasets are shown in Fig. 6. Then we label each patch as one of the following three types: “sharp”, “motion blur”, or “focal blur”. In total, we generated 290 focal blur patches, 217 motion blur patches, and 516 unblurred patches from training set and 223, 139, and 271 patches, respectively, from testing set.

We evaluate the performance of our classifier using the Precision-Recall curve. Let N be the number of patches to be classified, f_i be the label for patch i , and a_i be the ground truth label for patch i , we define the measurements as

$$\begin{aligned}
 Recall &= \frac{|\{i; f_i = a_i \& a_i = true\}|}{|\{i; f_i = true\}|}, \\
 Precision &= \frac{|\{i; f_i = a_i \& a_i = true\}|}{|\{i; a_i = true\}|}, \\
 Accuracy &= \frac{|\{i; f_i = a_i\}|}{N}.
 \end{aligned}$$

In the first step of our algorithm, patches labeled as “blur” are considered as *true* instances in evaluating blur/nonblur classifier whereas, in the second step, patches labeled as “motion” are considered as *true* in evaluating motion/focal blur classifier.

For evaluating individual features in blur detection, we plot precision-recall curves to show the discriminatory power. Similar to the definition of η_a , we set $\eta_i = P(q_i | Blur) / P(q_i | Sharp)$, where $i = \{1, 2, 3\}$ and show in Fig. 5(a) the precision-recall curves for each blur metric η_i and the combined metric η_a in classifying

blur/nonblur regions. Our classifier η_a , in general, is with the best performance and achieves over 95% precision for low recall. Among our blur metrics, the feature of gradient magnitude span η_2 shown in blue is the most discriminative one. This is not surprising since blurred patches rarely contain steep changes of intensities. Table 1 shows the maximum accuracy rate, as well as the overall accuracy rate calculated by averaging over all recall levels. The maximum accuracy rate 76.98% for blur/nonblur classification is achieved when $\eta_a = 0.4$. This threshold is used later for our application of blur region recognition and segmentation.

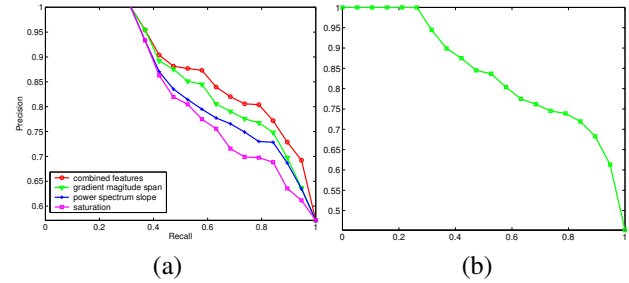


Figure 5. Precision-Recall curves of our classification. (a) Blur/nonblur classification results. The curve corresponding to the classifier using η_a is shown in red, whereas the curves for classifiers using η_1 , η_2 , and η_3 , are shown in green, blue, and purple, respectively. (b) Precision-Recall curve for motion/focal blur classification.

Classification Tasks	Overall Accuracy Rate	Maximum Accuracy Rate
Blur / Nonblur	63.78%	76.98%
Motion / Focal blur	65.45%	78.84%

Table 1. Accuracy Rate on the testing dataset.

In the second step, we test motion/focal blur classification using η_b . Fig. 5(b) shows the precision-recall curve. As listed in Table 1, the maximum accuracy rate is 78.84% when $\eta_b = 1.3$. This threshold is also used later for blur segmentation.

Because of the similarity of the blurred and low-contrast regions in natural images, our classification results inevitably contain errors. We examined incorrectly classified patches and found that the latent ambiguous texture or structure in patches is the main cause of errors. For example, Fig. 7(a) and (b) show patches wrongly classified as blurred regions due to the shadow or low contrast pixels. The patch in Fig. 7(c) is mistakenly classified as motion blur because of the strong directional structures.

The patch-based blur detection and classification can serve as a foundation for ranking the degree of image blur. In experiments, we collect a set of flower pictures searched from Google and Flickr. Each image is segmented into

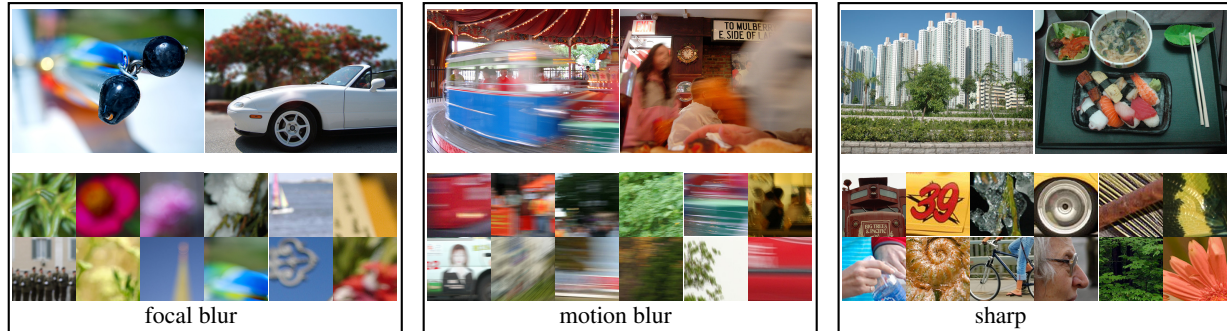


Figure 6. Selected examples of images(first row) and manually labeled patches(second row) from our datasets.

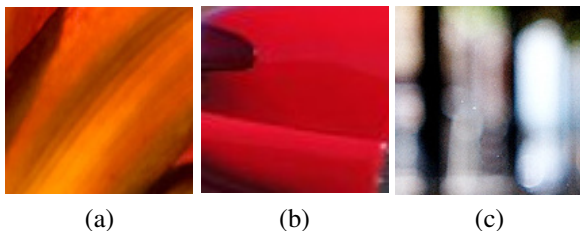


Figure 7. Examples of wrongly classified patches.

patches with size 20×20 . We rank each image by a blur confidence value defined proportional to the number of blurred regions in each image. Fig. 9 shows the blur recognition results by our method where the blur confidence of each image increases from left to right, coincident with human visual perception.

5.2. Blur Region Segmentation

Our method can also be used in partially blurred image segmentation. Because the blur classification is performed on patches, we detect the blur information of one pixel by using a local patch centered at it. The spatial similarity of blur types for neighboring pixels is also considered by introducing a smoothness term similar to that defined in [19]. Combining the blur detection and the pair-wise blur type similarity, we construct a Markov Random Field for each image. The corresponding Gibbs energy is minimized by applying the Graph Cuts method [16].

We show two blur segmentation examples in Fig. 8, where (a) and (b) give a comparison of the segmentation results produced by the method described in [18] and our approach. With comparable segmentation results, our method do not need to infer the blur kernel and make the directional motion assumption. Our features are computed within patches of size 50×50 centered at each pixel.

In (c) and (d) of Fig. 8, we show that our method is able to partition an image into several segments with respect to different blur types. In this example, we apply our blur/nonblur classifier to first segment out the unblurred regions (shown in red in (d)) and then extract mo-

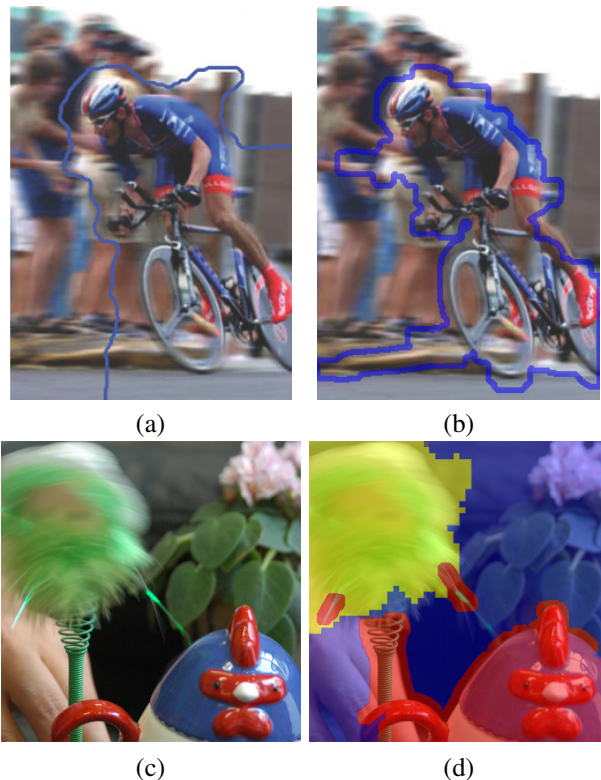


Figure 8. Blur segmentation results for partially blurred images. (a) Blur segmentation result presented in [18]. (b) Our result on the same example without needing to infer the blur kernels. (c) shows a challenging image example containing unblurred, motion blurred, and focal blurred regions. (d) Our segmentation result with unblurred regions in red, motion blurred region in yellow, and focal blurred regions in blue.

tion blurred regions (shown in yellow in (d)). Differently blurred regions are faithfully segmented, indicating that our automatic method can serve as a reliable basis for many region-based image applications, such as motion estimation, content-based image retrieval, and object extraction.



Figure 9. Partial blur recognition for flower images. Images are shown in an ascending order in term of the size of blurry regions.

6. Conclusion and Future Work

In this paper, we have proposed a partial-blur image detection and analysis framework for automatically classifying whether one image contains blurred regions and what types of blur occur without needing to performing image deblurring. Several blur features, measuring image color, gradient, and spectrum information, are utilized in a parameter training process in order to robustly classify blurred images. Extensive experiments show that our method works satisfactorily with challenging image data and can be applied to partial-blur image detection and blur segmentation. Our method, in principal, provides a foundation for solving many blur-oriented and region-based computer vision problems, such as content-based image retrieval, image enhancement, high-level image segmentation, and object extraction.

References

- [1] L. Bar, B. Berkels, M. Rumpf, and G. Sapiro. A variational framework for simultaneous motion estimation and restoration of motion-blurred video. In *ICCV*, 2007. 2
- [2] P. Bex and W. Makous. Spatial frequency, phase, and the contrast of natural images. *Journal of the Optical Society of America A*, 19(6):1096–1106, 2002. 3
- [3] G. Burton and I. Moorhead. Color and spatial structure in natural scenes. *Applied Optics*, 26(1):157–160, 1987. 3
- [4] Y. Chung, J. Wang, R. Bailey, S. Chen, and S. Chang. A non-parametric blur measure based on edge analysis for image processing applications. *IEEE Conference on Cybernetics and Intelligent Systems*, 1, 2004. 2
- [5] J. Da Rugna and H. Konik. Automatic blur detection for metadata extraction in content-based retrieval context. In *SPIE*, volume 5304, pages 285–294, 2003. 1
- [6] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Studying aesthetics in photographic images using a computational approach. In *ECCV (3)*, pages 288–301, 2006. 2
- [7] J. H. Elder and S. W. Zucker. Local scale control for edge detection and blur estimation. *PAMI*, 20(7):699–716, 1998. 1, 2
- [8] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ACM Trans. Graph.*, 25(3):787–794, 2006. 2
- [9] D. Field. Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America A*, 4(12):2379–2394, 1987. 3
- [10] D. Field and N. Brady. Visual sensitivity, blur and the sources of variability in the amplitude spectra of natural scenes. *Vision Research*, 37(23):3367–3383, 1997. 3
- [11] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *PAMI*, 13(9):891–906, 1991. 2
- [12] B. Hansen and R. Hess. Discrimination of amplitude spectrum slope in the fovea and parafovea and the local amplitude distributions of natural scene imagery. *Journal of Vision*, 6(7):696–711, 2006. 3
- [13] C. Harris and M. Stephens. A combined corner and edge detector. *Alvey Vision Conference*, 15, 1988. 5
- [14] J. Jia. Single image motion deblurring using transparency. In *CVPR*, 2007. 2
- [15] C. Kim. Segmenting a low-depth-of-field image using morphological filters and region merging. *IEEE Transactions on Image Processing*, 14(10):1503–1511, 2005. 2
- [16] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *PAMI*, 26(2):147–159, 2004. 7
- [17] L. Kovács and T. Szirányi. Focus area extraction by blind deconvolution for defining regions of interest. *PAMI*, 29(6):1080–1085, 2007. 2
- [18] A. Levin. Blind motion deblurring using image statistics. In *NIPS*, pages 841–848, 2006. 1, 2, 7
- [19] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum. Lazy snapping. *ACM Trans. Graph.*, 23(3):303–308, 2004. 7
- [20] P. Marziliano, F. Dufaux, S. Winkler, and T. Ebrahimi. A no-reference perceptual blur metric. In *ICIP (3)*, pages 57–60, 2002. 2
- [21] S. Roth and M. J. Black. Fields of experts: A framework for learning image priors. In *CVPR (2)*, pages 860–867, 2005. 3
- [22] Q. Shan, W. Xiong, and J. Jia. Rotational Motion Deblurring of a Rigid Object from a Single Image. In *ICCV*, 2007. 2
- [23] D. Tolhurst, Y. Tadmor, and T. Chao. Amplitude spectra of natural images. *Ophthalmic Physiol Opt*, 12(2):229–32, 1992. 3
- [24] J. Z. Wang, J. Li, R. M. Gray, and G. Wiederhold. Unsupervised multiresolution segmentation for images with low depth of field. *PAMI*, 23(1):85–90, 2001. 2
- [25] E. W. Weisstein. Autocorrelation. MathWorld. <http://mathworld.wolfram.com/Autocorrelation.html>. 4
- [26] W. Zhang and F. Bergholm. Multi-Scale Blur Estimation and Edge Type Classification for Scene Analysis. *International Journal of Computer Vision*, 24(3):219–250, 1997. 2