Analyzing spatially-varying blur

Analyzing Spatially-varying Blur

Ayan Chakrabarti[1,2]
ayanc@eecs.harvard.edu
[1]SEAS, Harvard University
Cambridge, MA 02138, USA.

Todd Zickler[1]
zickler@seas.harvard.edu
[2] CTL, Adobe Systems Inc.,
Cambridge, MA 02142, USA.

William T. Freeman[2,3]
bilff@mit.edu
[3]CSAIL, Mass. Inst. of Tech.,
Cambridge, MA 02139, USA.

Abstract

Blur is caused by a pixel receiving light from multiple scene points, and in many cases, such as object motion, the induced blur varies spatially across the image plane. However, the seemingly straight-forward task of estimating spatially-varying blur from a single image has proved hard to accomplish reliably. This work considers such blur and makes two contributions: a local blur cue that measures the likelihood of a small neighborhood being blurred by a candidate blur kernel; and an algorithm that, given an image, simultaneously selects a motion blur kernel and segments the region that it affects. The methods are shown to perform well on a diversity of images.

1. Introduction

Blur refers to image degradation caused by a pixel recording light from multiple scene points, and its common causes include camera shake, defocus, and object motion. Blur is characterized by a "kernel", or point-spread function (PSF), and when the cause of blur is defocus or object motion, the kernel typically changes across the image plane. Such spatially-varying blur is much harder to analyze than its uniform counterpart, and as a result, most methods for estimating spatially-varying blur require multiple images, an initial segmentation, or something else to augment the seemingly meager information available in a single photo.

The limited success with spatially-varying blur lies in stark contrast to recent advances with uniform blur. In the latter case, one can now reliably estimate and reverse the effects of blur through a variety of methods, even when the (spatially-uniform) kernel is of a complex, non-parametric form [4, 11, 12, 14, 17, 20]. The difference in the spatially-varying case is that blur must be inferred locally, using many fewer observations than are available in a large, uniformly-blurred image. And this means that one must use stronger image models to be able to extract the required information from a reduced set of measurements.

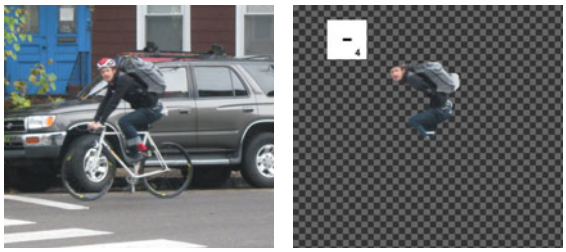


Figure 1. What's moving ? Given an image with motion blur, we seek to infer the blur kernel and the region it affects. The image on the right shows a segmentation of the blurred object in the image, and an estimate of the blur kernel generated by our algorithm.

This paper presents a novel cue for analyzing blur locally and reasoning about non-uniform cases. Starting with a standard sub-band decomposition (a "local Fourier transform"), we introduce a probability model for unblurred natural images that is simple enough to "play well" with the decomposition but powerful enough for accurate inference. This gives us a robust and efficient likelihood measure for a small image window being blurred by a given kernel.

This cue is then applied to the problem of segmenting motion-blurred regions from otherwise sharp images and simultaneously selecting the blur kernel acting on the affected region. We evaluate this approach using a diverse collection of images that each contain a single moving object. It yields satisfactory estimates of the blur kernel in most cases, and as suggested in Fig. 1, it is able to obtain useful segmentations when combined with color information.

2. Related work

Inferring the point spread function from a blurred image requires simultaneously reasoning about the PSF and the latent sharp image on which it operates. To address the problem, one must define the family of blur kernels to be considered as well as a model—often probabilistic in nature—for sharp images. Existing methods differ in terms of the types of blur they consider, the input they assume, and whether or not they consider spatial variation.

978-1-4244-6985-7/10/$26.00 ©2010 IEEE

When the blur is spatially-uniform, one has the good fortune of being able to accumulate evidence across the entire image plane, and as a result, one can afford to consider a very general class of blur kernels. In this context, a variety of recent techniques have shown that it is possible to recover completely non-parametric (*i.e.*, tabulated) blur kernels, such as those induced by camera shake, from as little as one image [4, 11, 12, 14, 17, 20]. One general insight that is drawn from this work is that instead of simultaneously estimating the blur kernel and a single sharp image that best explain a given input, it is often preferable to first estimate the blur kernel as that which is most likely under a *distribution* of sharp images. Levin et al. [17] refer to this process as "MAP$_k$ estimation", and we will use it here.

Blur caused by motion or defocus often varies spatially in an image, and in these cases, one must infer the blur kernels using local evidence alone. To succeed at this task, most methods consider a reduced family of blur kernels (*e.g.*, one-dimensional box filters), and they incorporate more input than a single image. When two or more images are available, one can exploit the differences between blur and sensor noise [22] or the required consistency between blur and apparent motion [1, 2, 5, 6] or blur and depth [9, 10]. As an alternative to using two or more images, one can use a single image but assume that a solution to the foreground/background matting problem is given as input [7, 8, 13]. Finally, one may be able to use a single image, but with special capture conditions to exaggerate the effect of the blur, as is done in [16] with a coded aperture (and additional user input) for estimating defocus blur that varies spatially with scene depth.

More related to our work is the pioneering effort of Levin [15], who also considers single- image input without pre-matting. The idea is to segment an image into blurred and non-blurred regions and to estimate the PSFs by exploit differences in the distribution of intensity gradients within the two types of regions. These relatively simple image statistics allow compelling results in some cases, but they fail dramatically in others. One of the primary motivations of our work is to understand these failures and to develop stronger computational tools to eliminate them.

3. Modeling Blur

We use $y[n]$ to represent an observed image, with $n \in \mathbb{R}^2$ indicating the location of a pixel. Let $x[n]$ be the corresponding latent sharp image, that is, the image that would have been captured in the absence of any blur or noise. The images x and y are assumed to be related by

$$y[n] = (k_n * x)[n] + z[n], \tag{1}$$

where z is sensor noise and k_n is the spatially-varying blur kernel acting at pixel location n. We assume z to be white Gaussian noise:

$$z[n] \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_z^2). \tag{2}$$

Our goal is to estimate the blur kernels k_n. This problem is under-determined even when the kernel does not change from point to point, and allowing spatial variation adds considerable complexity. We need to make some assumptions to proceed, and one assumption we will make is that the kernel is constant in any local neighborhood. This turns out to be a reasonable assumption in most cases, but as we will see, it limits our ability to obtain localized boundaries based on blur information alone.

3.1. Sub-band Decomposition

Blur is naturally analyzed in the frequency domain, and when the blur is uniform over an image ($k_n = k$), the convolution theorem lets us simplify the analysis by "diagonalizing" (1) with the Fourier transform:

$$Y(\omega) = X(\omega)K(\omega) + Z(\omega), \tag{3}$$

where $\omega \in [-\pi, \pi]^2$ and $Y(\omega)$, $X(\omega)$, $K(\omega)$ and $Z(\omega)$ are Fourier transforms of x, y, k and z respectively. However, when the blur kernel varies spatially, the signals $x[n]$ and $y[n]$ are no longer related by convolution, and a global Fourier transform is of limited utility.

Instead, we require a localized frequency representation. Let $w[n] \in \{0, 1\}$ be a symmetric window function with limited spatial support, and let $\{f_i\}$ be a set of filters:

$$f_i[n] = w[n] \times \exp\left(-j\langle \omega_i, n\rangle\right), \tag{4}$$

with frequencies $\omega_i \in \mathbb{R}^2$. The choice of $\{\omega_i\}_i$ will depend on the window size, and it can be made to keep the filters orthogonal. Thus, we assume $\langle f_i, f_j\rangle = 0$ if $i \neq j$.

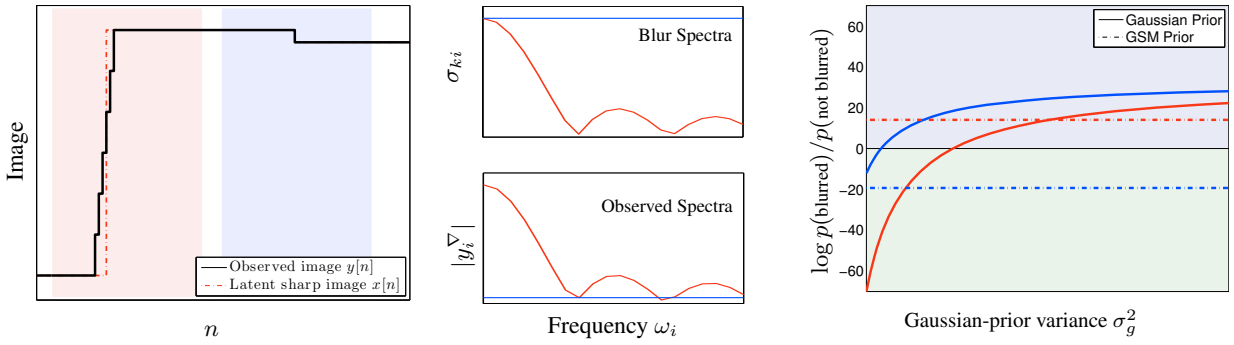
Applying these filters to an image x yields the corresponding responses $x_i[n] = (x * f_i)[n]$. For any location n, the set of values $\{x_i[n]\}_i$ can be imagined as the Fourier decomposition of a window centered at that location.

When k_n is constant in neighborhood of size equal or greater than the support of w, (1) and (4) can be combined:

$$y_i[n] = \left(x * (k_n * f_i)\right)[n] + (z * f_i)[n]. \tag{5}$$

This is similar to the spatially-uniform case (3), but different in a critical way: In this spatially-varying case, there is no means of expressing the transform coefficients $\{y_i[n]\}_i$ in terms of the corresponding $\{x_i[n]\}_i$ alone, because pixels in any window of y have values determined by pixels outside of that window in x. In short, we do not obtain the same "diagonalization".

However, we can derive a useful relationship between the *statistics* of the coefficients $\{x_i[n]\}_i$ and $\{y_i[n]\}_i$. Specifically, let $y = x * k$ hold in a neighborhood of n that is equal to the support of w, and let all elements of $x[n]$ in



(a) (b) (c)

Figure 2. Evaluation of the Gaussian and GSM priors with a toy example. (a) A 1-D image with a high contrast blurred edge (the dotted red line corresponds to the latent sharp edge) and a low contrast blurred edge. (b) A comparison of the spectra $\{\sigma_{ki}\}_i$ (top) corresponding to the blur kernel (red line) and no blur (blue line), with the actual observed coefficient magnitudes for two windows containing the blurred and sharp edges (red and blue lines respectively, corresponding to the windows shown in (a)). We see that while the observed magnitudes match the "shapes" of their corresponding expected spectra, they are scaled differently based on the contrast of the edges. (c) An evaluation of the Gaussian and GSM prior-based likelihoods in (9) and (11) to classify both windows correctly as being blurred or not-blurred. A window is classified as blurred if the log of the ratio is above 0. We can see that while the GSM-based likelihood yields the correct classification for both edges, there is no single value of σ_g^2 for which both windows are classified correctly, *i.e.* for which the blue and red curves are below and above 0 respectively.

a slightly larger neighborhood—equal to the support of w plus that of k—be independent and identically distributed as $\mathcal{N}(0, \sigma^2)$. Then it is easy to show that the transform coefficients $\{x_i[n]\}_i$ and $\{y_i[n]\}_i$ are also distributed as zero-mean Gaussians with variances given by

$$\begin{aligned} \mathbb{E}|x_i[n]|^2 &= \sigma^2 \langle f_i, f_i \rangle, \\ \mathbb{E}|y_i[n]|^2 &= \sigma^2 \sum_n |(k * f_i)[n]|^2 \triangleq \sigma^2 \sigma_{ki}^2, \\ \mathbb{E}x_i[n]x_j^*[n] &= \mathbb{E}y_i[n]y_j^*[n] = 0, \qquad \forall i \neq j. \end{aligned} \quad (6)$$

Therefore, as an analogue to the diagonalization in (3), convolution with k independently scales the variance of each coefficient $y_i[n]$ so that it can be related directly to the corresponding coefficient $x_i[n]$ of the latent sharp image. Because of this analogy, we will refer to the set of transform coefficients of the blur kernel, $\{\sigma_{ki}^2\}_i = \{\sum_n |(k * f_i)[n]|^2\}_i$, as the *blur spectrum* in the remainder of the document. (A 1-D example is shown in top of Fig. 2(b).)

4. Prior on Sharp Images

Since the problem of estimating the blur kernels k_n from the input image y is under-determined, we need a statistical prior model for the latent sharp image x to constrain the problem. We will then use this prior in conjunction with a MAP_k estimation approach [17] to derive expressions for $p(\{y_i[n]\}_i|k)$: the likelihood that the blur kernel at n is k.

4.1. Gaussian

A natural choice for an image prior might be a Gaussian distribution on image gradients. Specifically, if ∇ is a derivative filter in a particular direction,

$$x^\nabla[n] = (\nabla * x)[n] \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_g^2). \quad (7)$$

Since (5) is linear in x we can write

$$y_i^\nabla[n] = f_i * y_i^\nabla = (k_n * x^\nabla) * f_i + (\nabla * z) * f_i. \quad (8)$$

Combining (7) and (8) with the fact that the noise is white Gaussian (2), and using our expressions for variance (6), we can derive the following expression for the likelihood of the observed coefficients $\{y_i^\nabla[n]\}_i$ when $k_n = k$:

$$p(\{y_i^\nabla[n]\}_i|k) = \prod_i \mathcal{N}\left(y_i^\nabla[n]|0, \sigma_g^2\sigma_{ki}^2 + \sigma_{zi}^2\right), \quad (9)$$

where σ_{ki}^2 is the blur spectrum as per (6) and σ_{zi}^2 are the filtered noise coefficients. These coefficients can be written as simple scaled versions of the noise variance, $\sigma_{zi}^2 = \sigma_z^2\sigma_{\nabla i}^2$ with $\sigma_{\nabla i}^2 \triangleq \sum_n |\nabla * f_i[n]|^2$, using reasoning similar to (6).

A Gaussian prior has the virtue of simplicity and the likelihood in (9) can be computed easily in closed form. Moreover, despite the fact that empirical distributions of gradients in natural images have significantly heavier tails than a Gaussian distribution, the model has been used with some amount of success for blur estimation in the absence of spatial variation [17].

Unfortunately, it is far less useful when dealing with spatially-varying blur. This is illustrated in Fig. 2 using a toy one-dimensional "image" that contains a low-contrast edge that is not blurred, and a high-contrast edge that is blurred by a box filter k. We assume that the kernel k is

known, and our task is to decide, by looking at the spectrum in each local window of the input image, whether that window was blurred by k or was not blurred at all (or equivalently, was blurred by an impulse). The top of part (b) in the figure shows the spectra of the two blur kernels (k and the impulse).

Figure 2(c) shows the likelihood ratio for different values of the sharp image model parameter σ_g, and it shows that the Gaussian-based likelihood model (9) is never able to classify both windows correctly. That is, we can choose the model parameter to correctly classify one or the other, but not both simultaneously. Intuitively, this results from the fact even if we can reasonably expect the mean square gradient values of an entire sharp image to be close to σ_g^2, the same is not true within different small windows of that sharp image whose statistics can change quite dramatically.

To gain further insight, we can look at the local spectra of the input image around the two edges, as depicted in the bottom of Fig. 2(b). We see that while the magnitudes of the spectra match the *shapes* of the two blur kernel spectra, their relative scales are very different. The classification fails, then, because the simple Gaussian-based likelihood model (9) involves *absolute* variance values, at a scale that is fixed and determined by the choice of our image prior model parameter σ_g^2. Viewed another way, the likelihood term is adversely affected by the difference in contrast between the two edges, and this prevents it from being able to make a decision based purely on how sharp they are.

In the next section, we introduce an alternate image prior model that specifically aims at capturing edge sharpness distinct from edge contrast. It is a heavy-tailed distribution that has the advantage of being reasonably tractable computationally within the sub-band representation.

4.2. Gaussian Scale Mixture

Instead of choosing a single Gaussian with variance σ_g^2, we model gradients in an image using an infinite mixture of zero-mean Gaussian distributions having a continuous range of variances. Additionally, we 'tie together' the scale parameters within each spatial neighborhood η. Formally,

$$p(\{x^{\nabla}[n]\}_{n\in\eta}) = \int p_s(s) \prod_{n\in\eta} \mathcal{N}\left(x^{\nabla}[n]|0,s\right)\, ds, \quad (10)$$

where s is the scale parameter and $p_s(s)$ is a probability distribution on s which we choose to be uniform in the set $\mathcal{S} = (0, s_{\max})$. Note that conditioned on s, the image gradients in neighborhood η are i.i.d, and the marginal distribution of any single gradient $x^{\nabla}[n]$ has heavier tails than a Gaussian distribution.

This is a slight variation of a Gaussian Scale Mixture (GSM), and GSM-based priors have been used previously to model natural images[21] for denoising [18] and deconvolution [23]. If we choose the size of the neighborhood η to be equal to or greater than the sum of the supports of the window w and blur kernel k, we can derive an expression for the likelihood analogous to the simple Gaussian case of the previous section:

$$p(\{y_i^{\nabla}[n]\}_i|k) \propto \int_{\mathcal{S}} \prod_i \mathcal{N}\left(y_i^{\nabla}[n]|0, s\sigma_{ki}^2 + \sigma_{zi}^2\right) ds. \quad (11)$$

Comparing this to (9), we see that the likelihood term integrates over all possible values of the variance (or scale) s in a window, as opposed to relying on a single choice σ_g^2. Returning to the toy example of Fig. 2, this has the effect of making the likelihood more agnostic to the absolute magnitudes of the local spectra, and allows it to latch on to the *shape* of the spectrum instead. As shown in Fig. 2(c), the GSM prior leads to a correct classification of the two edges.

One drawback of (11) is that it is significantly more expensive to compute due to the integration over $\mathcal{S}$. However, we find that the integrand—the likelihood of coefficients conditioned on any particular s— is a product of exponentials and typically behaves as an impulse on s. That is, the conditional likelihood typically has a sharp peak at an "optimal" choice of s and is negligible for all other values. So we can approximate it using

$$p(\{y_i^{\nabla}[n]\}_i|k) \propto \max_{s\in\mathcal{S}} \prod_i \mathcal{N}\left(y_i^{\nabla}[n]|0, s\sigma_{ki}^2 + \sigma_{zi}^2\right). \quad (12)$$

The optimal value $\hat{s}$ that maximizes the conditional likelihood in the above equation must satisfy

$$\hat{s} = \left(\sum_i r_i(\hat{s})\right)^{-1} \sum_i r_i(\hat{s}) \frac{\left|y_i^{\nabla}[n]\right|^2 - \sigma_{zi}^2}{\sigma_{ki}^2},$$
$$r_i(\hat{s}) = \left(1 + \frac{\sigma_{zi}^2}{\hat{s}\sigma_{ki}^2}\right)^{-2}, \quad (13)$$

and can be found using fixed point iterations. Convergence is typically quick for low values of σ_z^2. If the solution $\hat{s}$ lies outside the set $\mathcal{S}$, it can be appropriately clipped and then used to compute (12).

5. Detecting Motion

In the previous sections, we introduced a local blur cue that allows the efficient computation of a robust likelihood measure (12) for a given window in the image being blurred by a candidate kernel. We now describe an algorithm that uses this cue to simultaneously segment a motion-blurred region of an image and determine the blur kernel that affects this region. This is useful because the blur kernel contains direct information about the orientation and speed (relative to exposure time) of moving objects [1, 2, 5, 6]. In the following, we assume that all of the motion-blurred regions of the input image are corrupted by the same kernel.

For a given image $y[n]$, every pixel is either part of the stationary sharp "background" or the motion-blurred "foreground". Therefore, $k_n \in \{k_0, k_m\}$ where k_0, chosen to either be an impulse (*i.e.* no blur at all) or a mild defocus blur, corresponds to the blur kernel acting on the stationary regions of the image , and k_m is the motion-blur kernel. Our task is to estimate k_m and to assign a label $I[n]$ to every location, where $I[n] = 0$ if $k_n = k_0$ and 1 otherwise.

The motion blur kernel k_m is in turn assumed to be one of a discrete set of possible candidates, corresponding to horizontal or vertical box filters of certain lengths. (These correspond roughly to horizontal or vertical object motion with fixed velocity.) Formally,

$$k_m \in \{b_{hl_1}, \ldots b_{hl_L}, b_{vl_1}, \ldots b_{vl_L}\}, \tag{14}$$

where b_{hl} is a horizontal "box" filter of length (corresponding to number of pixels the object moved during exposure) l, *i.e.*,

$$b_{hl}[n] = \begin{cases} 1/l & \text{if } n_y = 0, 0 \le n_x < l, \\ 0 & \text{otherwise,} \end{cases}, \tag{15}$$

b_{vl} is a similarly defined vertical box filter, and $\{l_1, \ldots l_L\}$ is the set of lengths we wish to select from.

5.1. Selecting the Kernel

To handle both horizontal and vertical candidate blurs $\{b_{hl}\}$ and $\{b_{vl}\}$, we need to use two sets of coefficients $\{y_i^h[n]\}_i$ and $\{y_i^v[n]\}_i$ defined as,

$$y_i^h[n] = (f_{ih} * \nabla_h * y)[n], \quad y_i^v[n] = (f_{iv} * \nabla_v * y)[n], \tag{16}$$

where ∇_h, ∇_v are horizontal and vertical gradient filters, and $\{f_{ih}\}_i, \{f_{iv}\}_i$ correspond to sub-band decompositions based on horizontal and vertical 1-D windows w_h and w_v.

The first step is to select the motion blur kernel, k_m, and in order to do so, we need to derive an expression for the likelihood $p_m(b) \doteq p(k_m = b)$. If b is horizontal, then the coefficients $y_i^h[n]$ should be explained either by b or k_0, and the coefficients in the orthogonal direction $y_i^v[n]$ should all be explained by k_0 (*i.e.* they are not affected by the motion blur). The converse is true if b is vertical, and this reasoning leads us to define

$$\begin{aligned} p_m(b_{lh}) &= \left[\prod_n \max_{k \in \{k_0, b_{lh}\}} p\left(\{y_i^h[n]\}_i | k\right)\right] \\ &\quad \times \left[\prod_n p\left(\{y_i^v[n]\}_i | k_0\right)\right], \\ p_m(b_{lv}) &= \left[\prod_n \max_{k \in \{k_0, b_{lv}\}} p\left(\{y_i^v[n]\}_i | k\right)\right] \\ &\quad \times \left[\prod_n p\left(\{y_i^h[n]\}_i | k_0\right)\right]. \end{aligned} \tag{17}$$

The blur kernel k_m can then be chosen amongst the candidate set in (14) to maximize this likelihood $p_m(\cdot)$.

5.2. Segmentation

Once the kernel k_m has been chosen, the next step is to do a segmentation by assigning the labels $I[n]$ for each location. While the relative likelihoods of the sub-band coefficients under k_0 and k_m could be used to do the segmentation, best results are obtained when these are combined with statistical distributions for object and background color into an MRF model as used in traditional segmentation.

Similar to the approach in [19], we choose the labels $l[n]$ as the solution to an energy minimization problem. We define a "Gibbs" energy function to capture the blur likelihoods, color distributions, and a spatial smoothness constraint on $I[n]$ as

$$\begin{aligned} E(I[n], \theta_f, \theta_b) &= \sum_n B_n(I[n]) + \alpha \sum_n C_n(I[n], \theta_f, \theta_b) \\ &\quad + \sum_{(n,n') \in \mathcal{P}} V_{n,n'}(I[n], I[n']), \end{aligned} \tag{18}$$

where θ_f, θ_b are the sets of parameters for the color distributions of the moving object and stationary background, C_n and B_n encode the likelihoods of $I[n]$ from blur and color information respectively, α is a scalar parameter that controls the relative contribution of the two, and $V_{n,n'}(\cdot,\cdot)$ enforces smoothness by favoring the same label assignments at all pairs $\mathcal{P}$ of neighboring locations.

$B_n(I[n])$ is defined in terms of the likelihoods from our blur model:

$$B_n(I[n]) = \begin{cases} -\log p(\{y_i^\nabla[n]\}_i | k_o) & \text{if } I[n] = 0, \\ -\log p(\{y_i^\nabla[n]\}_i | k_m) & \text{if } I[n] = 1. \end{cases}, \tag{19}$$

where $y_i^\nabla[n]$ are the sub-band coefficients for the same direction as the selected k_m, i.e. $y_i^\nabla[n] = y_i^h[n]$ if a horizontal box blur was chosen as k_m, and the vertical coefficients $y_i^v[n]$ otherwise.

The color of pixels in the moving object and background are each modeled as a (discrete) mixture of Gaussians, parameterized by θ_f and θ_b respectively. C_n is then defined as

$$C_n(I[n], \theta_f, \theta_g) = -\log p(y[n] \,|I[n], \theta_f, \theta_b\,). \tag{20}$$

Fig. 3 shows a map of B_n, C_n and of their weighted combination for a typical image. Finally, we choose $V_{n,n'} = \gamma[I[n] \neq I[n']]$ where γ is a scalar that corresponds to the penalty for making different label assignments to neighboring pixels.

The labeling $I[n]$ can be initialized based on B_n alone, and then we employ an iterative scheme to minimize the energy E in (18). In each iteration, we estimate the optimal parameters θ_f and θ_b using expectation-maximization (EM), and then assign $I[n]$ using graph cuts [3].

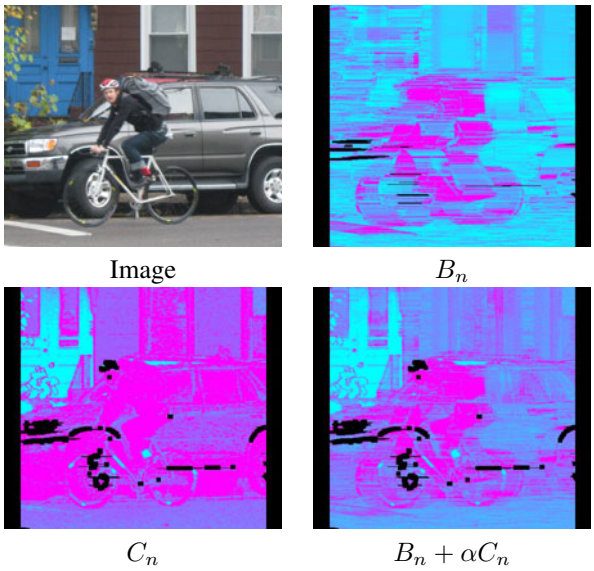


Figure 3. Contributions of blur and color models to segmentation. While the blur model suffers from localization problems because of defined window sizes, the color model by itself is unable to distinguish between blurred and stationary objects that look similar. The two, however, compliment each other and when combined in an MRF model with smoothness constraints, give good quality segmentations (see Fig. 5).

6. Experiments

In this section, we test the algorithm with a variety of images, each captured with a point-and-shoot or cell-phone camera. Each image contains a single motion-blurred object, and we are interested in evaluating our method's ability to choose the correct orientation and length for the blur kernel and segment the blurred object.

6.1. Implementation Details

Both the horizontal and vertical sub-band transforms were defined in terms of windows of length $W = 61$. The filters $\{f_{ih}\}$ and $\{f_{iv}\}$ were then generated with ω_i equal to the corresponding 1-D set of frequencies $2\pi u/W$, $u \in \{1, \ldots 15\}$. The gradient filters ∇ were defined simply as $[-1, \ \ 1]$ (in the appropriate direction), *i.e.* the gradient values corresponded to the finite differences between neighboring pixels.

For the blur length, we searched over the discrete set of integers from 4 to 15 pixels. Larger lengths can be handled by downsampling the image to an appropriate resolution. For segmentation, the object and foreground RGB color vectors were modeled as mixtures of 4 and 6 Gaussians respectively. All choices of parameters (including σ_z^2, α and γ) were kept fixed for all images.

To evaluate the real-world applicability of the algorithm, a diverse set of images were captured using various devices, including a mid-level digital SLR, a consumer point-shoot camera and a cell phone cameras. Since the algorithm was run on the JPEG images, a tone scale correction was done, assuming a standard gamma of 2.2, to recover the linear intensities for computing the blur likelihoods. All images were scaled down to be roughly 550 pixels wide.

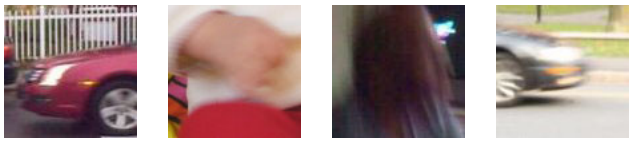


Figure 4. Crops of the blurred images from a few images with the corresponding detected blur length and orientation. The detected orientation matches the apparent motion direction, and images that are qualitatively more blurred have larger detected lengths.

These images and a MATLAB implementation of the algorithm can be downloaded from the project page at **http://www.eecs.harvard.edu/~ayanc/svblur/**.

6.2. Results

We first qualitatively evaluate the ability of the algorithm to choose the correct orientation and length for the motion blur kernel k_m. In Fig. 4, we show close-ups of moving objects (see Fig. 5 for full images) and the corresponding selected blur kernels. The algorithm correctly determines whether the motion is horizontal or vertical, and for objects that appear to have more significant blur it selects larger kernel lengths.

Figure 5 shows the full segmentation results for a set of images, as well as the relative likelihoods from the blur model alone. We find that the algorithm does well on this diverse set of real world images, despite approximate assumptions about noise variance, gamma correction, etc. It is also able to detect the blurred regions even in cases of relatively minor motion (see images 2 and 5) and is fairly robust even in cluttered backgrounds (see image 4). In some cases however, some regions are incorrectly marked blurred because of color similarities to the blurred object (image 2), or erroneous estimates from B_n (image 4, where the blur model "sees" soft shadows as blurred edges). Supplemental material on the project website linked above contains more detailed results, including results on other images.

Our MATLAB implementation of the algorithm takes roughly 4 minutes to compute the sub-band coefficients of an image, compute the likelihoods under all candidate blur kernels, choose k_m, and evaluate the blur energy B_n. The full segmentation, which includes iterations to estimate the color distributions using EM and label assignments using graph cuts, takes an additional 12 minutes. These execution times were recorded for a 600×450 image using a machine with a 2.8GHz Pentium processor and 1GB of RAM.

Figure 5. Results of motion blur estimation and segmentation on 7 images. (a) The input images. Image 1 was taken with a digital SLR, 2, 3 and 5 were taken with a cell phone camera, and 5, 6 and 7 were taken with a consumer point-shoot camera. (b) Relative probability maps of blur vs no-blur using the blur prior alone (*i.e.* B_n). (c,d) Segmentation results of the full algorithm, with (c) showing regions labeled as not blurred, and (d) showing regions labeled as blurred along with the estimated blur kernel.

7. Conclusion

We propose a new local cue for analyzing spatially-varying blur. It involves a sub-band decomposition to isolate the "local frequency" components of an image, and a prior image model based on a Gaussian Scale Mixture to handle the variation in gradient statistics among local windows of a single image. Combined, these allow one to efficiently compute the likelihood of a particular window in an image being blurred by a candidate kernel, and this likelihood conveniently incorporates a preference for sharp edges without being adversely affected by the natural variation in edge contrast that occurs in a typical image.

We use the proposed likelihood measure to determine the orientation and degree of motion in images with moving objects, and by combining them with color information, we show that one can obtain reasonable segmentations of the objects. Our segmentations are hard, and an immediate direction to explore is computing soft mattes instead. The boundaries of moving objects are linear combinations of a blurred foreground and a sharp background, and it is likely that the proposed blur machinery can be used to enhance reasoning about such mattes that would otherwise be obtained using color information alone.

More broadly, while the proposed tools have been used to analyze motion blur, they are general enough to handle any instance of spatially-varying blur. For example, for defocus blur where the kernel changes with scene depth, one can imagine using the framework in [16] but doing away with the requirement for special capture conditions by using the proposed likelihood measure in (12) instead of cues from a coded aperture.

Acknowledgments

The authors thank Yair Weiss, Anat Levin, Gregg Wilensky, Scott Cohen and Stephen Schiller for discussions. AC and TZ were supported by the US Army Research Laboratory and the US Army Research Office under contract/grant number 54262-CI, by Adobe Systems Inc., and by a fellowship from the Alfred P. Sloan Foundation. WF was supported by NGA NEGI-1582-04-0004, MURI Grant N00014-06-1-0734, and gifts from Microsoft, Google, and Adobe.

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] B. Bascle, A. Blake, and A. Zisserman. Motion deblurring and super-resolution from an image sequence. In *ECCV*, 1996.

[3] Y. Boykov and G. Funka-Lea. Graph cuts and efficient N-D image segmentation. *IJCV*, 70(2):109–131, 2006.

[4] J. Cai, H. Ji, C. Liu, and Z. Shen. Blind motion deblurring from a single image using sparse approximation. In *CVPR*, 2009.

[5] W. Chen, N. Nandhakumar, and W. Martin. Image motion estimation from motion smear-a new computational model. *IEEE Trans. on PAMI*, 18(4):412–425, 1996.

[6] S. Cho, Y. Matsushita, and S. Lee. Removing non-uniform motion blur from images. In *ICCV*, 2007.

[7] S. Dai and Y. Wu. Estimating space-variant motion blur without deblurring. In *ICIP*, 2008.

[8] S. Dai and Y. Wu. Motion from blur. In *CVPR*, 2008.

[9] P. Favaro, M. Burger, and S. Soatto. Scene and motion reconstruction from defocused and motion-blurred images via anisotropic diffusion. In *ECCV*, 2004.

[10] P. Favaro and S. Soatto. A variational approach to scene reconstruction and image segmentation from motion-blur cues. In *CVPR*, 2004.

[11] R. Fergus, B. Singh, A. Hertzmann, S. Roweis, and W. Freeman. Removing camera shake from a single photograph. In *ACM SIGGRAPH*, 2006.

[12] H. Ji and C. Liu. Motion blur identification from image gradients. In *CVPR*, 2008.

[13] J. Jia. Single image motion deblurring using transparency. In *CVPR*, 2007.

[14] N. Joshi, R. Szeliski, and D. Kriegman. PSF estimation using sharp edge prediction. In *CVPR*, 2008.

[15] A. Levin. Blind Motion Deblurring using Image Statistics. In *NIPS*, 2006.

[16] A. Levin, R. Fergus, F. Durand, and W. Freeman. Image and depth from a conventional camera with a coded aperture. In *ACM SIGGRAPH*, 2007.

[17] A. Levin, Y. Weiss, F. Durand, and W. Freeman. Understanding and evaluating blind deconvolution algorithms. In *CVPR*, 2009.

[18] J. Portilla, V. Strela, M. Wainwright, and E. Simoncelli. Image denoising using Gaussian scale mixtures in the wavelet domain. *IEEE Trans. on Image Proc.*, 12(11):1338–1351, 2003.

[19] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM SIGGRAPH*, 2004.

[20] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. In *ACM SIGGRAPH*, 2008.

[21] M. Wainwright and E. Simoncelli. Scale mixtures of Gaussians and the statistics of natural images. In *NIPS*, 2000.

[22] L. Yuan, J. Sun, L. Quan, and H. Shum. Image deblurring with blurred/noisy image pairs. In *ACM SIGGRAPH*, 2007.

[23] Y. Zhang and N. Kingsbury. Image deconvolution using a gaussian scale mixtures model to approximate the wavelet sparseness constraint. In *ICASSP*, 2009.