Blur Kernel Estimation using the Radon Transform

Taeg Sang Cho[1] Sylvain Paris[2] Berthold K. P. Horn[1] William T. Freeman[1]

[1]Massachusetts Institute of Technology [2]Adobe Systems, Inc.

Abstract

Camera shake is a common source of degradation in photographs. Restoring blurred pictures is challenging because both the blur kernel and the sharp image are unknown, which makes this problem severely underconstrained. In this work, we estimate camera shake by analyzing edges in the image, effectively constructing the Radon transform *of the kernel. Building upon this result, we describe two algorithms for estimating spatially invariant blur kernels. In the first method, we directly invert the transform, which is computationally efficient since it is not necessary to also estimate the latent sharp image. This approach is well suited for scenes with a diversity of edges, such as man-made environments. In the second method, we incorporate the Radon transform within the MAP estimation framework to jointly estimate the kernel and the image. While more expensive, this algorithm performs well on a broader variety of scenes, even when fewer edges can be observed. Our experiments show that our algorithms achieve comparable results to the state of the art in general and produce superior outputs on man-made scenes and photos degraded by a small kernel.*

1. Introduction

Hand-held photos with a slow shutter speed are often blurry because the photographer moves during the exposure. Tripods solve this issue but are cumbersome to transport. The mechanical stabilization of the lens or of the sensor provided in modern cameras also greatly help reducing shake but longer exposures remain challenging. Recently, Fergus et al. [4] have proposed an algorithm to restore blurry photos. The difficulty is that many pairs of images and blur kernels can explain the captured blurry photos, and most of them are not acceptable because of noise and ringing artifacts. Nonetheless, Fergus's algorithm recovers a satisfying sharp image thanks to a variational inference scheme and *a priori* knowledge about natural images. Since then, several other approaches have been proposed. However, in these methods, the problem remains severely unconstrained. Although we require that the convolution of the recovered image and kernel together reproduce the captured photo and that the kernel is compact and non-negative, there are still numerous possible solutions to the problem. Because of that, these methods resort to complex optimization techniques that are slow and that do not always succeed.

In this paper, we show that edges in the scene reveal additional cues about the kernel and that this information helps constrain the deblurring problem and makes it easier to solve. Intuitively, different orientations are affected differently by blur, and the set of different edge profiles can be seen as a "signature" of the kernel. Formally, we show that we can recover the *Radon transform* of the kernel from the blurred edges. We complement this result with heuristics to detect edges in blurred photos. In scenes with numerous edges such as a man-made environment, this approach is sufficient to directly estimate the kernel. Further, this process does not involve repeated deconvolutions, which significantly speeds up the process. In scenarios with fewer edges, we integrate our cues in a maximum a priori formulation (MAP) and define a likelihood term that constrains the estimated kernel. Our tests show that this approach allows for high-quality deblurring on a broad variety of scenes. We also analyze the performance of several algorithms including ours on different categories of scenes and with various kernels. This experiment reveals that blur kernel estimation is a rich problem with different best solutions for different regions of the problem space. In particular, it shows that our methods perform well on photographs that are poorly restored by other existing techniques.

Contributions The main contribution of this work is the construction of the Radon transform of the blur kernel by analyzing the edges in the image. Using this result, we describe two practical algorithms for removing spatially invariant blur, one based on the inverse Radon transform and one based on MAP estimation. Finally, we show that our method is successful on images that are challenging for other techniques.

1.1. Related work

In this paper, we focus on removing spatially invariant blur. To resolve the inherent ambiguity in blind deconvoluton, different assumptions on blur kernels and natural images have been incorporated. Fergus *et al*. [4] exploit the knowledge that the gradient histogram of natural images exhibits a heavy-tailed profile and that blur kernels due to camera shake have a sparse support. Shan *et al*. [12] introduce a local prior to detect and smooth surfaces. Cai *et al*. [1] assume that a blur kernel is sparse in the curvelet domain and the sharp image is sparse in the framelet domain. These techniques solve a large system of equations to find the sharp image and/or the blur kernel that reproduce the observation while conforming to a prior knowledge about blur and natural images, something which is time consuming and do not always succeed.

A few methods explicitly exploit blurred edges to estimate blur. Rosenfeld and Kak introduced the idea in their seminal book [11]. Jia [6] estimates an alpha matte from user-selected edges, and subsequently recovers the blur kernel from the matte. Joshi *et al*. [7] assume that the blur kernel is unimodal to predict the location of sharp edges in a blurry photo and estimate the blur kernel accordingly. Cho and Lee [2] extend Joshi *et al*. [7] in a multi-scale manner to handle multimodal kernels. Cho and Lee [2] reduce the computation by working in the gradient domain.

The Radon transform has been used to estimate 1D kernels that explain motion blur, i.e. the camera is assumed to move along a straight line [5, 14].

In comparison to these methods, we introduce a new likelihood term that describes 2D blur kernels by their Radon transform. This term is valid for arbitrary kernels and does not require a multiscale approach. We build an efficient algorithm upon this result and show that it can restore images that are challenging for other methods.

2. Kernel estimation from edges

In this section, we introduce our image formation model and give a brief introduction to the Radon transform. Then, we build upon these concepts to estimate the blur kernel using the edges in the blurred image.

2.1. Blur model

We model the formation of the blurred image B as a convolution of a blur kernel k and a sharp latent image I:

$$B = k \otimes I + \nu \tag{1}$$

where ν is the observation noise. Our goal is to reconstruct the sharp image I from the observed blurry photo B.

2.2. Background on the Radon transform

We briefly review the Radon transform and its inverse for 2D signals. For an in-depth presentation, we refer to [3, 15].

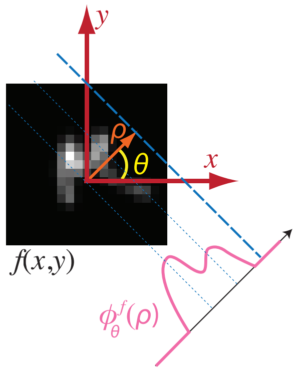

Figure 1: *The Radon transform $\phi_\theta^f(\rho)$ of a signal f is an integral of the signal along the line $\rho = x\cos(\theta) + y\sin(\theta)$ (i.e the dashed line).*

The Radon transform of a signal $f(x, y)$ is the collection of integrals of $f(x, y)$ along straight lines, where the straight lines can be conveniently parameterized using the offset ρ and the orientation θ as shown in Figure 1.

$$\phi_\theta^f(\rho) = \iint f(x, y)\, \delta\big(\rho - x\cos(\theta) - y\sin(\theta)\big)\, \mathrm{d}x\, \mathrm{d}y \tag{2}$$

The function ϕ_θ^f, for fixed θ, can be viewed as a projection of the signal f along the direction orthogonal to orientation θ. With enough projections of f along different orientations θ, the original signal f can be recovered. This is known as the *inverse Radon transform* and is computationally inexpensive. For instance, the commonly used *filtered backprojection* method consists of a "ramp filter" (1-D convolutions) applied to each projection, followed by backprojection. One way to implement this method is to use 1-D Fourier transforms to convert the 1-D convolutions into products, and then use a 2-D inverse Fourier transform [15, § 7.2]. We further discuss this issue in Section 2.4.

2.3. Radon transform of the blur kernel

In this section, we show how to compute a Radon projection from a color step edge. For now, we assume that we have detected edges. We explain how to do so in Section 2.5. The intuition behind our approach is that a line integral of the kernel can be formulated as the convolution of the kernel with the image of an ideal line. We implement this idea in three phases. First, we fit an ideal binary edge to a color edge. Then, we take the derivative in the direction orthogonal to the edge to obtain an estimate of the response to an ideal line. Finally, we show that the response equals a projection of the kernel. Or in other words, the collection of such responses for edges in different orientations yields samples of the Radon transform of the kernel. Without loss of generality, we will illustrate the approach using a vertical edge, that is the case $\theta = 0$. The situation for other orientations can be derived simply by rotating the axes. The steps are as follows:

(i) Remove the colors We consider a binary step edge $H(x,y)$ equal to 1 if $x \geq 0$ and 0 otherwise. Given two colors c_1 and c_2, we define an ideal color step edge as $E = Hc_1 + (1-H)c_2$. The blurred version B_E of this edge is: $k \otimes E = c_1(k \otimes H) + c_2(1 - k \otimes H)$. In this section, we assume that c_1 and c_2 are known, we will describe how to estimate them later, and seek to retrieve $B_H = k \otimes H$ that describes the blurred edge independent of its colors. At each pixel, we have three equations, one for each RGB channel, of the form $B_E = B_H c_1 + (1 - B_H)c_2$. Since B_H is the only unknown, this is an overconstrained linear system that can be solved with a least-squares formulation, which in practice is a simple average of the solutions from each RGB channel.

(ii) Convert an edge response into a line response We now have a scalar description of the blurred edge: $B_H = k \otimes H$ and we seek to derive the blurred line response from the blurred edge response. To do so, we remark that $H(x,y) = \int_{-\infty}^{x} \delta(t)\,\mathrm{d}t$ and that $\frac{\partial H}{\partial x}(x,y) = \delta(x)$, i.e. taking the derivative of the ideal step edge along the direction orthogonal to it produces a line. We compute the derivative of $k \otimes H$ in the x direction and obtain $B_L = \partial B_H / \partial x = k \otimes \delta_x$ which is the image of a blurred line (here $\delta_x(x,y) = \delta(x)$).

(iii) Sample the profile Finally, we show that sampling B_L horizontally produces a vertical Radon projection of the kernel. Expanding $k \otimes \delta_x$, we get $B_L(x,y) = \iint k(u,v)\delta(x-u)\,\mathrm{d}u\,\mathrm{d}v$. Since the formula does not depend on y, we can omit it. Comparing with Equation (2) in the case $\theta = 0$, one can recognize that $B_L(x)$ is equal to the vertical Radon projection $\phi_0^k(x)$ of k.

(iv) Align the projections To reconstruct an accurate blur kernel, the projections must be consistently aligned with each other. For this, we exploit the fact that the center of mass of the kernel projects onto the center of mass of each projection. In practice, we compute a first estimate of each projection and then shift it to align its center of mass on the origin of the coordinate system.

Practical construction of a Radon projection

In practice, we are given a blurry image B. We first run heuristics to detect color step edges. For each edge, we consider a local region B_E and estimate the c_1 and c_2 colors. Then, we sample the pixels along the direction orthogonal to the edge. For each of these pixels, we compute $B_H = \frac{1}{3}\sum_{\{r,g,b\}} \frac{B_E - c_1}{c_2 - c_1}$. This produces a 1D set of pixels with values in $[0,1]$. The first derivative of this signal is the Radon projection $\phi_\theta^k(x)$ of the kernel along the direction θ orthogonal to the edge. We compute its center of mass and shift it accordingly. The next section describes how to use these projections to recover the kernel.

2.4. Recovering the blur kernel from its projections

In this section, we seek to recover the kernel k from a set of Radon projections $\{\widetilde{\phi}_i\}$ along the directions $\{\theta_i\}$ estimated using edges in the input blurry image (§ 2.3). A standard approach is to use filtered backprojection that convolves the 1-D projections with a ramp filter and then backproject them in 2-D to build k (see [15, § 7.2] for details). Although the produced estimate (Fig. 2b) captures the overall structure of the kernel, it is too "thick" and it exhibits "streaks" that would degrade the deblurred image. We address this issue by adding *a priori* knowledge about blur kernels. Since they correspond to the camera trajectory during the exposure, they are sparse and smooth. We use a Bayesian formulation to express this assumption, that is, we seek to maximize the posterior probability $p(k|B)$ of the blur kernel k given the observed image B. We use a classical decomposition into a likelihood term and a prior: $p(k|B) \propto p(B|k)\,p(k)$. To model the likelihood term $p(B|k)$, we define the linear operator R_θ that computes a line integral in the direction θ and seek to satisfy the constraint that the actual projections $R_{\theta_i}k$ of the blur kernel should match our measured projections $\widetilde{\phi}_i$:

$$p(B|k) \propto \exp\left(-\frac{1}{2\eta^2_{\text{obs}}} \sum_{i=1}^{N} \|\widetilde{\phi}_i - R_{\theta_i}k\|^2\right) \qquad (3)$$

where N is the number of Radon projection $\widetilde{\phi}$ that we have extracted from the image and η^2_{obs} is the variance of observation noise that comes from the imaging noise with variance η^2_{img} and from the inaccuracies in the estimation of the orientation θ. Since we estimate the projections $\widetilde{\phi}_i$ with finite differences for the first derivative (§ 2.3), the impact of the imaging noise is doubled. The orientation noise adds a fac-

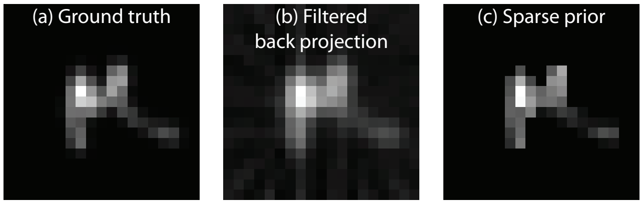

Figure 2: *We compare the performance of different inverse Radon transform algorithms on a synthetic example. We blur the test pattern (shown in the supplemental material) with the kernel (a), and estimate 120 projections in 12 regularly spaced orientations. The kernel recovered using filtered backprojection (b) is too big and exhibits faint streaks (this may be better seen on the electronic version). In comparison, our approach based on the sparse prior (Eq. 4) yields a more accurate result (c).*

tor α. This gives $\eta_{\text{obs}}^2 = (2+\alpha)\,\eta_{\text{img}}^2$. We set α to 1 using cross-validation.

For the kernel prior, we incorporate the knowledge that intensity profiles of blur kernels, as well as gradient profiles of blur kernels, are sparse:

$$p(k) \propto \exp\left(-\lambda_1\|k\|^{\gamma_1} - \lambda_2\|\nabla k\|^{\gamma_2}\right) \tag{4}$$

We use the same parameters for all experiments, determined through cross-validation: $\lambda_1 = 1.5$, $\gamma_1 = 0.9$, $\lambda_2 = 0.1$, and $\gamma_2 = 0.5$. In practice, we minimize the negative log-posterior $-\log\big(p(B|k)\,p(k)\big)$ with an iterative reweighted least-squares method [8, 13].

Robustness to noise We found that our approach behaves well with a level of Gaussian noise up to $\sigma \approx 1\%$. Our experiments show similar results for other methods, e.g. [2, 4]. Noise may also affect the orientation θ since we use image gradients to estimate it. Figure 3 shows that our method behaves well up for a Gaussian noise on θ up to $\sigma \approx 5^\circ$, which is largely sufficient in practice.

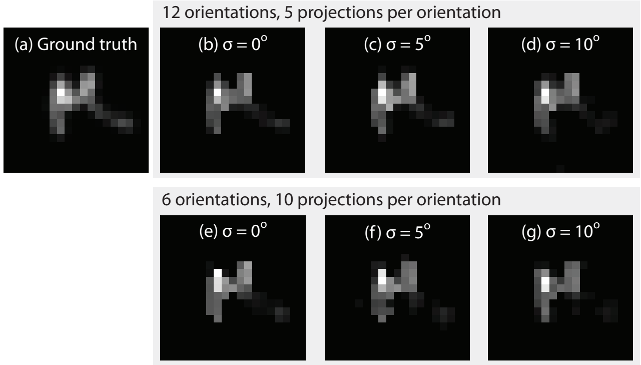

Figure 3: *We analyze the robustness of our inverse Radon transform algorithm with respect to edge orientation estimation error. We use the same test pattern as in Figure 2 and add Gaussian noise of variance σ^2 to the orientation θ. We show the reconstructed kernels in each case.*

Speeding up the computation When the number N of projections is large, the optimization becomes slow. We observe, however, that having many projections in similar orientations is beneficial mostly in terms of reducing noise of the projection along that orientation. We exploit this by averaging out the noise first and then solving the optimization problem. We do so by binning projections with similar orientations. We define a small set of n orientations $\widehat{\theta}_j$ with $n \ll N$ and group the measured values θ_i according to their nearest $\widehat{\theta}_j$. This forms sets $\mathcal{P}_j$ of i indexes. For each non-empty set, we compute the average projection $\widehat{\phi}_j = \frac{1}{w_j}\sum_{i\in\mathcal{P}_j}\widetilde{\phi}_i$ where w_j is the size of $\mathcal{P}_j$. We then reformulate the likelihood as:

$$\exp\left(-\frac{1}{2\eta_{\text{obs}}^2}\;\sum_{j=1}^{n} w_j\|\widehat{\phi}_j - R_{\widehat{\theta}_j}k\|^2\right) \tag{5}$$

In practice, we define a $\widehat{\theta}_j$ value every degree and less than half of them are non-empty.

2.5. Detecting step edges in blurry images

To build the Radon projections ϕ_i, we need to find isolated step edges. We introduce a set of image analysis heuristics that detect edges in blurry images (Fig. 4). We initiate the process with a map E of candidate edges by running a simple edge detector that thresholds the gradients to keep the top 2.5% strongest edges. Then, we select edges using simple rules. First, we discard edges with a low contrast because the ratio between the edge profile and noise would be too low. We measure the edge colors c_1 and c_2 (§ 2.3) and if $\|c_1 - c_2\| < 0.03$ in the RGB space, we discard the edge. Then, we select isolated edges where we can observe a single edge at a time. Assuming that we know the maximum size s of the kernel, in practice given by the user, we discard edges that have one or more other edges at a distance less than s in their orthogonal direction. Third, we check that the colors along the edge are a mixture of c_1 and c_2. If one the edge sample is farther than 0.03 of the $[c_1, c_2]$ segment in RGB space, we discard the edge. Last, we select straight edges. We consider the orientations $\theta_\ell \in [0, \pi[$ of the candidate edges within a disc $\mathcal{D}$ of radius s. We apply the criterion proposed by Watson [16] to estimate their alignment. We compute $\frac{1}{L}\sqrt{\big(\sum_\ell \cos(2\theta_\ell)\big)^2 + \big(\sum_\ell \sin(2\theta_\ell)\big)^2}$ where L is the number of candidate edges in $\mathcal{D}$. We keep only the edge for which this quantity is larger than 0.97. The edge detection works sufficiently well in practice but its accuracy degrades with large blur kernels. We further discuss this issue in the result section.

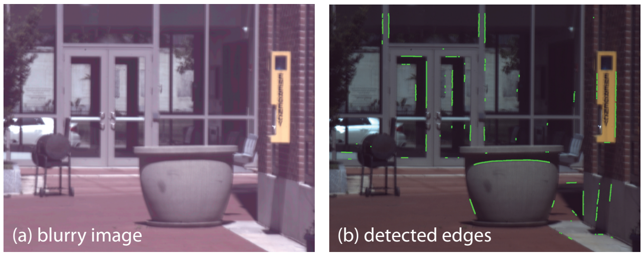

Figure 4: *Example of edges detected in a blurry image using our heuristics.*

2.6. Experimental results

We compare our algorithm's performance to three other methods: Fergus *et al.* [4], Shan *et al.* [12], and Cho and Lee [2]. We use the same deconvolution algorithm [8] for everyone so that only the kernel estimation algorithm differentiates the results. Our tests are based on images in raw format (14 bits per pixel) to ensure that the data are linear and that no degradation is introduced by JPEG compression. Figure 5 shows results on photos of man-made environments on which our method typically performs well. In these cases, our approach produces better results than previous work, e.g. with less ringing and/or sharper details. It is also efficient with running times on the order of a minute with our unoptimized Matlab implementation. Our algorithm does not perform as well when there are not enough edges in different orientations (Fig. 6a) or when we do not detect enough edges (Fig. 6b). In these cases, we are not able to retrieve Radon projections with enough precision and/or with a sufficient diversity of orientations. Improving our heuristics to detect edges must help in these cases. In the following section, we propose a second algorithm in which we incorporate the Radon projections into a MAP estimation process. In the result section (§ 4), we compare these two approaches and further analyze their performance.

3. The RadonMAP algorithm

To handle images with fewer edges, we develop a method that incorporates kernel projection constraints within a MAP estimation. We maximize the posterior probability with respect to the blur k and the image I jointly [2, 12]:

$$\arg\max_{k,I}\ p(k, I|B) \tag{6}$$

which we decompose into likelihood and prior terms: $p(k, I|B) \propto p(B|k, I)\, p(k)\, p(I)$. We use an image prior $p(I) \propto \exp\left(-\lambda\|\nabla I\|^{\gamma}\right)$ that favors a piecewise-smooth latent image and the same kernel prior $p(k)$ as earlier (Eq. 4).

One could model the likelihood $p(B|k, I)$ using the observation model (Eq. 1) only:

$$p(B|k, I) \propto \exp\left(-\tfrac{1}{2\eta_{\text{img}}^2}\|B - k \otimes I\|^2\right) \tag{7}$$

However, with such a likelihood and priors, Levin *et al.* [9] show that the joint estimation of the kernel k and the sharp image I does not produce a satisfying result because the joint probability (Eq. 6) is maximized when k is an impulse and I the input blurry image B. To resolve this issue, we augment the likelihood term $p(B|k, I)$ with the Radon projections (Eq. 5):

$$\exp\left(-\frac{\|B - k \otimes I\|^2}{2\eta_{\text{img}}^2} - \frac{\sum_{j=1}^{n} w_j \|\widehat{\phi}_j - R_{\widehat{\theta}_j} k\|^2}{2\eta_{\text{obs}}^2}\right) \tag{8}$$

The Radon transform term relies on a strong assumption that natural images consist of step edges and that every detected edge should be an ideal step edge. It essentially penalizes the no-blur solution, and steers the joint distribution $p(k, I|B)$ to favor the correct solution.

Practical algorithm Algorithm 1 shows the pseudocode for the joint estimation algorithm which we name *RadonMAP*. We resort to an alternating optimization to find k and I (Eq. 6): we first maximize the joint distribution $p(k, I|B)$ with respect to the kernel k while keeping the image I fixed, then we maximize it with respect to I while holding k fixed. We iterate the two steps until convergence (typically 5 times). Inspired by Cho and Lee [2], we filter the latent image estimate I_{est} using a bilateral filter [10] before re-estimating the kernel. This step significantly improves the kernel estimation by removing ringing and noise artifacts due to the inaccuracy of the initial guess of the kernel.

Algorithm 1 The RadonMAP blur estimation algorithm

```
% Initial kernel estimation
$k_{\text{out}} \leftarrow \arg\min_k$ Eq. 5
for $\ell$ = 1 to $maxIteration$ do
    $I_{\text{est}} \leftarrow \arg\max_I p(k_{\text{out}}, I|B)$ % Estimate latent image
    $I_{\text{bf}} \leftarrow \text{bilateralFilter}(I_{\text{est}})$
    $k_{\text{out}} \leftarrow \arg\max_k\ p(k, I_{\text{bf}}|B)$ % Estimate kernel
end for
$I_{\text{out}} \leftarrow \arg\max_I\ p(k_{\text{out}}, I|B)$
```

Experimental results Figure 6(a,b) illustrates how the RadonMAP algorithm improves the results of the photos that are challenging for the direct inverse transform. In these cases, the RadonMAP outputs are crisper and have less ringing. In general, RadonMAP yields cleaner kernels without the disconnected components that may sometimes appear with the direct inverse transform when only a few edges are available. Our unoptimized Matlab implementation of RadonMAP runs in about 30 minutes.

4. Quantitative evaluation

We compare the performance of blur estimation algorithms quantitatively using the error metric introduced by Levin et al. [9]. This metric assumes that the ground-truth sharp image I_{gt} and kernel k_{gt} are known. It computes the accuracy of a deblurred image I_{out} while accounting for possible inaccuracies due to deconvolution by comparing it to the accuracy of an image $I_{k_{\text{gt}}}$ deblurred with the ground-truth kernel. The error ratio is defined as $\|I_{\text{out}} - I_{\text{gt}}\|^2/\|I_{k_{\text{gt}}} - I_{\text{gt}}\|^2$. An error ratio of 1 is ideal, that is, the estimated kernel yields a result as good as the ground-truth kernel. Lower quality results correspond to higher ratios. We use this metric as suggested by Levin *et al.*: we plot the cumulative error ratio (CER). The higher the curve, the better. We use the 8 kernels extracted from real-world blurry photos by Levin *et al.* [9]. The same authors also provide a set of 255×255 gray-scale test images but we believe that such low-resolution black-and-white images are not representative of real-world photos. Further, image analysis such as edge detection is nearly impossible on these small images, which makes it difficult to evaluate methods like Cho and Lee's [2] and ours. We address this point with a set of 6 one-megapixel color images

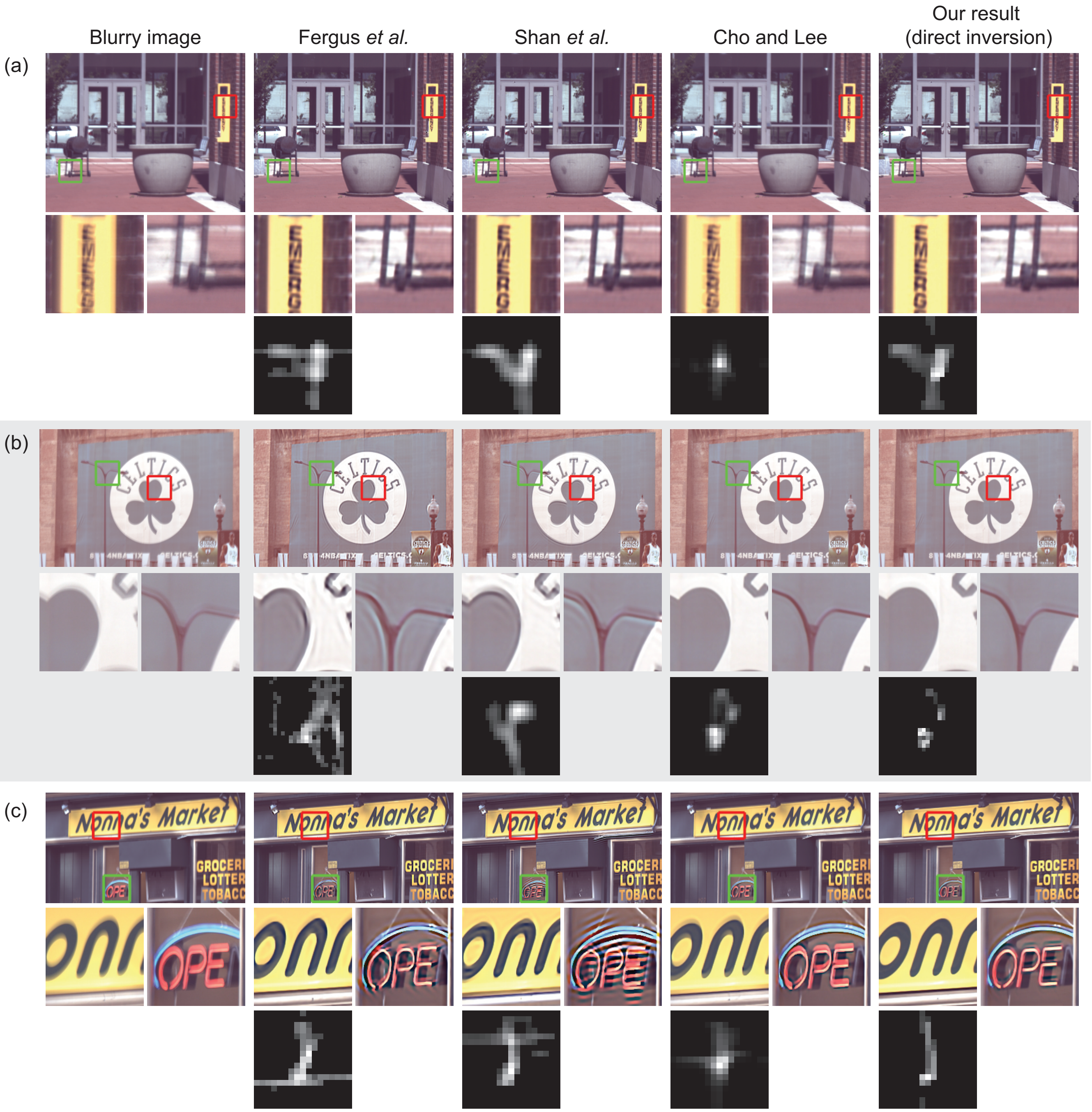

Figure 5: *We compare our algorithm to the methods of Fergus* et al. *[4], Shan* et al. *[12], and Cho and Lee [2]. On this man-made environments, our algorithm compares favorably to prior art. These results may be better seen on the electronic version.*

that are closer to a practical scenario. Half of the images are rich in edges while the other half has fewer of them so that we can test our methods in a favorable case and on more challenging data. We combine the 8 kernels and the 6 images and add Gaussian noise with $\sigma = 0.5\%$ to generate 48 test images.

Figure 7 compares the algorithms of Fergus *et al*. [4], Shan *et al*. [12], Cho and Lee [2], and our two approaches: the direct inverse transform (§ 2.4) and RadonMAP (§ 3). From a global point of view, Cho and Lee's method and RadonMAP perform better than others (Fig. 7a), with a slight advantage for Cho and Lee's algorithm. We further investigated the results by observing subsets of our 48 test images. First, we looked at the influence of the image content. As expected,

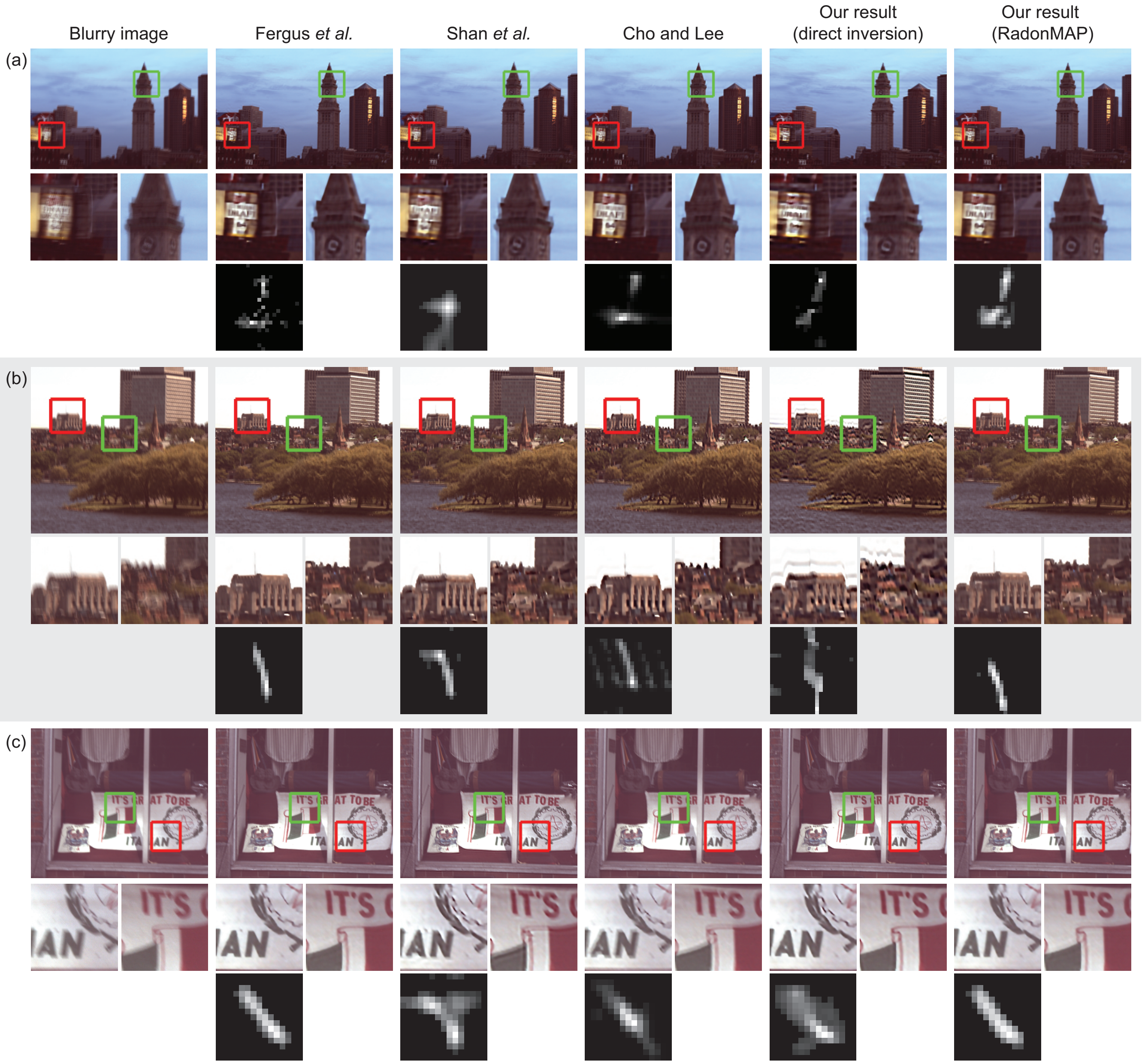

Figure 6: *By integrating kernel projection constraints to a MAP process, we improve the kernel estimation performance. In particular, we can restore blurry images, even when there are few edges detected (a-b). These results may be better seen on the electronic version.*

our methods yield more accurate results when a lot of edges are available (Fig. 7b,c). More surprisingly, the performance of Cho and Lee's technique decreases in the many-edges case, which may be due to the lower amount of texture in the images. We also analyzed the influence of the kernel size and found that our approach performs significantly better with small kernels than with larger ones (Fig. 7d,e). This comes from the higher performance of our edge detection routine when the images are less blurred. This result suggests that the heuristics to find image edges is currently the limiting factor of our method. From a more general point of view, these plots unveil cases that are challenging for existing techniques: images with many edges and low texture, and small blur kernels. The plots also show that our approach based on Radon projections brings a satisfying solution to these hard cases.

Acknowledgments This research is partially funded by NGA NEGI-1582- 04-0004, Shell Research, ONR-MURI Grant N00014-06-1-0734, and by gift from Microsoft and Adobe. The first author is partially supported by Samsung

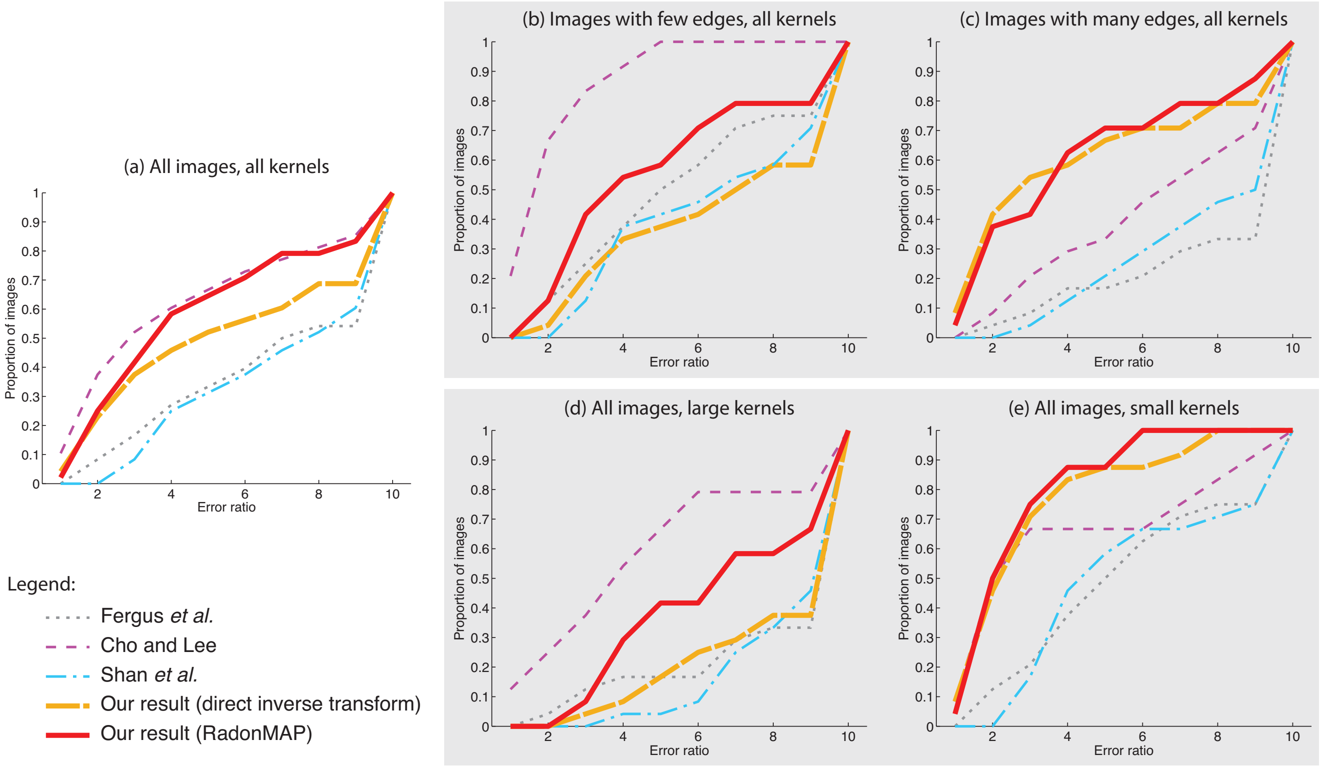

Figure 7: *Cumulative error ratios for the following algorithms: Fergus* et al. *[4], Shan* et al. *[12], Cho and Lee [2], our method based on directly inverting the Radon transform (§ 2.4), and our RadonMAP algorithm (§ 3). On average (a), Cho and Lee's algorithm performs best, closely followed by our RadonMAP method. A fine-grain analysis reveals that Cho and Lee's technique yields good results on images with few edges (b) and when the blur kernel is large (d). In comparison, our methods perform best on images with many edges (c) and with small kernels (e), thereby producing good results on cases that were poorly addressed so far.*

Scholarship Foundation.

References

[1] J.-F. Cai, H. Ji, C. Liu, and Z. Shen. Blind motion deblurring from a single image using sparse approximation. In *Proc. of the conf. on Computer Vision and Pattern Recognition*, 2009.

[2] S. Cho and S. Lee. Fast motion deblurring. *ACM Transactions on Graphics*, 28(5), 2009.

[3] S. R. Deans. *The Radon Transform and some of its applications*. Krieger Publishing Company, 1992.

[4] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single image. *ACM Trans. on Graphics*, 25(3), 2006.

[5] H. Ji and C. Liu. Motion blur identification from image gradients. In *Comp. Vision and Pattern Recognition*, 2008.

[6] J. Jia. Single image motion deblurring using transparency. In *Proc. of Comp. Vision and Pattern Recognition*, 2007.

[7] N. Joshi, R. Szeliski, and D. J. Kriegman. PSF estimation using sharp edge prediction. In *Proc. of the conf. on Computer Vision and Pattern Recognition*, 2008.

[8] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM Trans. on Graphics*, 26(3), 2007.

[9] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *Proc. of Computer Vision and Pattern Recognition*, 2009.

[10] S. Paris, P. Kornprobst, J. Tumblin, and F. Durand. Bilateral filtering: Theory and applications. Foundations and Trends in Computer Graphics and Vision, 2009.

[11] A. Rosenfeld and A. C. Kak. *Digital picture processing*. New York: Academic Press, 1982.

[12] Q. Shan, L. J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. *ACM Trans. on Graphics*, 27(3), 2008.

[13] C. V. Stewart. Robust parameter estimation in computer vision. *SIAM Reviews*, 41(3), 1999.

[14] H. Sun, M. Desvignes, Y. Yan, and W. Liu. Motion blur parameters identification from radon transform image gradients. In *Proc. of the conf. on Industrial Electronics*, 2009.

[15] P. Toft. *The Radon Transform - Theory and Implementation*. PhD thesis, Technical University of Denmark, 1996.

[16] G. S. Watson. *Statistics on spheres*. John Wiley and Sons, 1983.