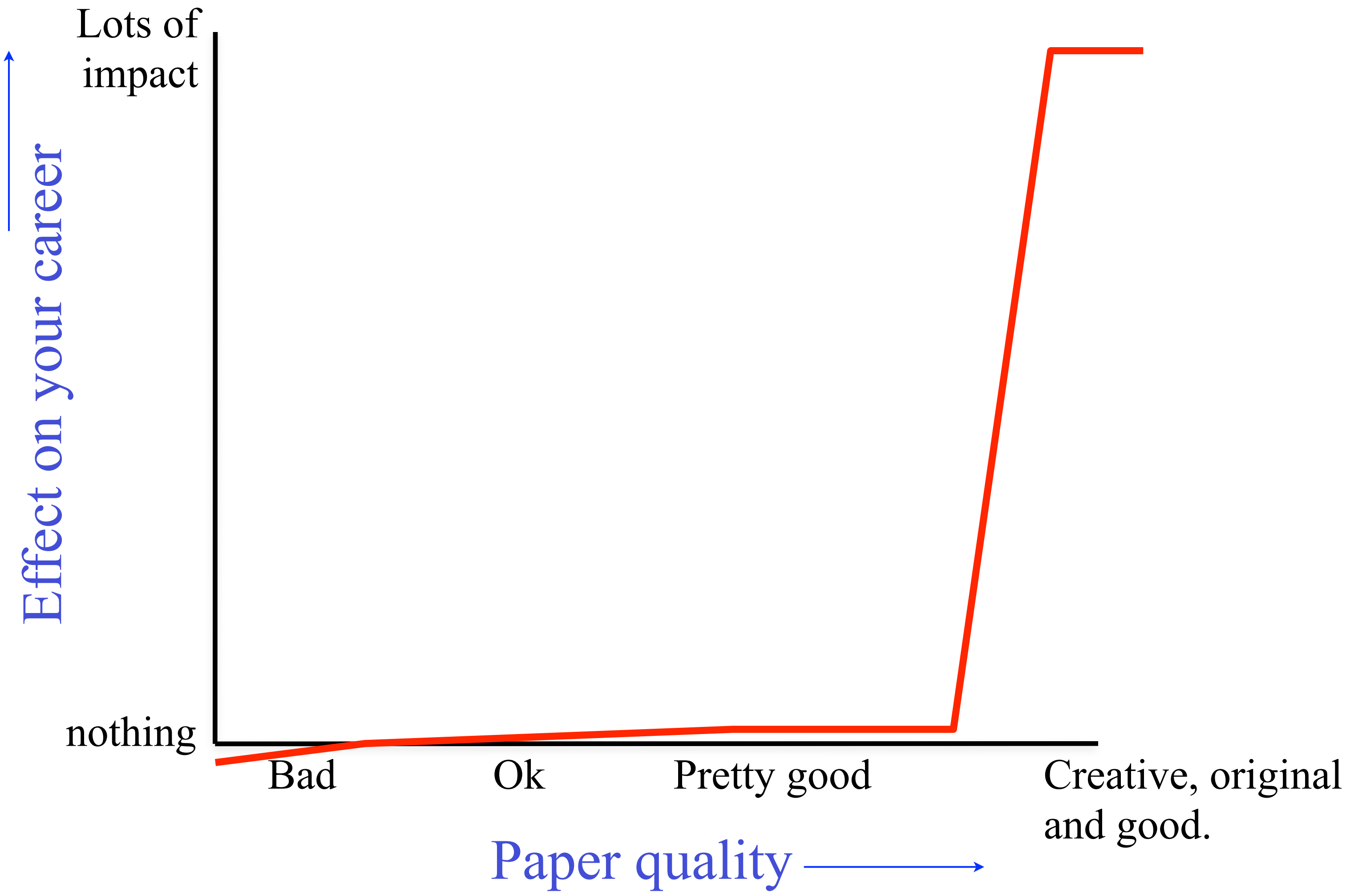
How to write a good research paper

Bill Freeman
MIT CSAIL

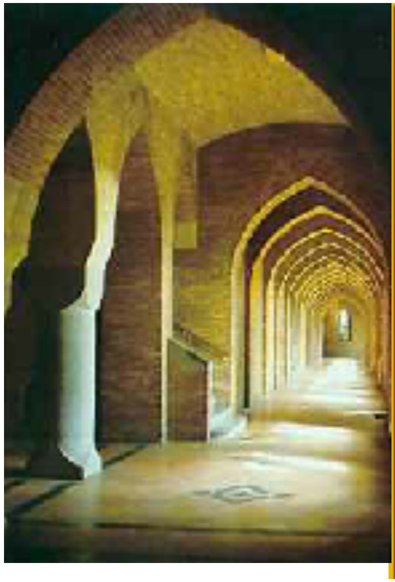
Nov. 29, 2018

A paper's impact on your career



Our image of the research community

- Scholars, plenty of time on their hands, pouring over your manuscript.



The reality:
more like a large, crowded marketplace

http://ducksflytogether.wordpress.com/2008/08/02/looking-back-khan-el-khalili/

Ted Adelson on how to write a good paper

(1) Start by stating which problem you are addressing, keeping the audience in mind. They must care about it, which means that sometimes you must tell them why they should care about the problem.

(2) Then state briefly what the other solutions are to the problem, and why they aren't satisfactory. If they were satisfactory, you wouldn't need to do the work.

(3) Then explain your own solution, compare it with other solutions, and say why it's better.

(4) At the end, talk about related work where similar techniques and experiments have been used, but applied to a different problem.

Since I developed this formula, it seems that all the papers I've written have been accepted. (told informally, in conversation, 1990).

Example paper organization: removing camera shake from a single photograph

1 Introduction

2 Related work

3 Image model

4 Algorithm

  Estimating the blur kernel

    Multi-scale approach

    User supervision

  Image reconstruction

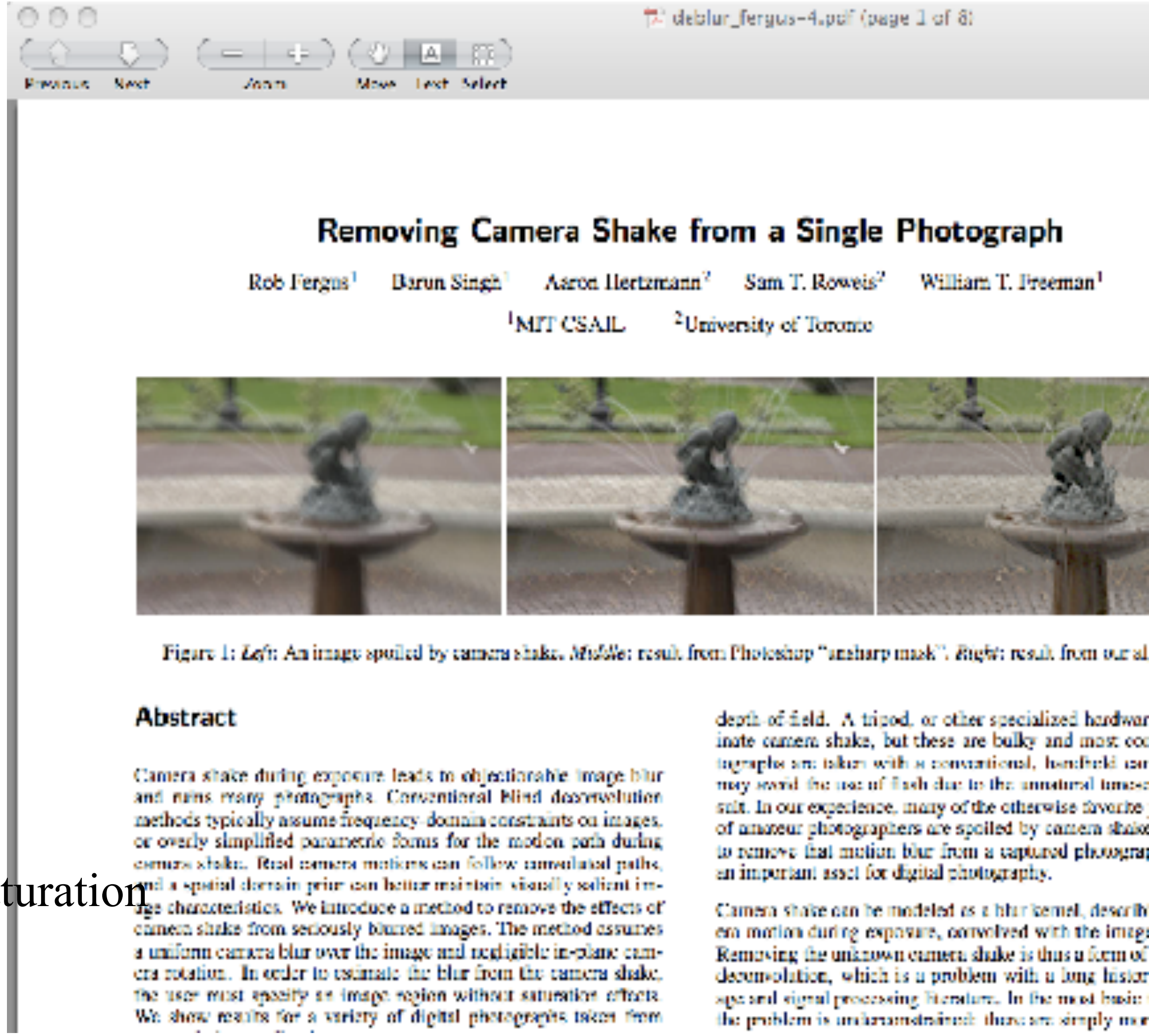
5 Experiments

  Small blurs

  Large blurs

  Images with significant saturation

6 Discussion



Removing Camera Shake from a Single Photograph

Rob Fergus[1] Barun Singh[1] Aaron Hertzmann[2] Sam T. Roweis[2] William T. Freeman[1]

[1]MIT CSAIL [2]University of Toronto

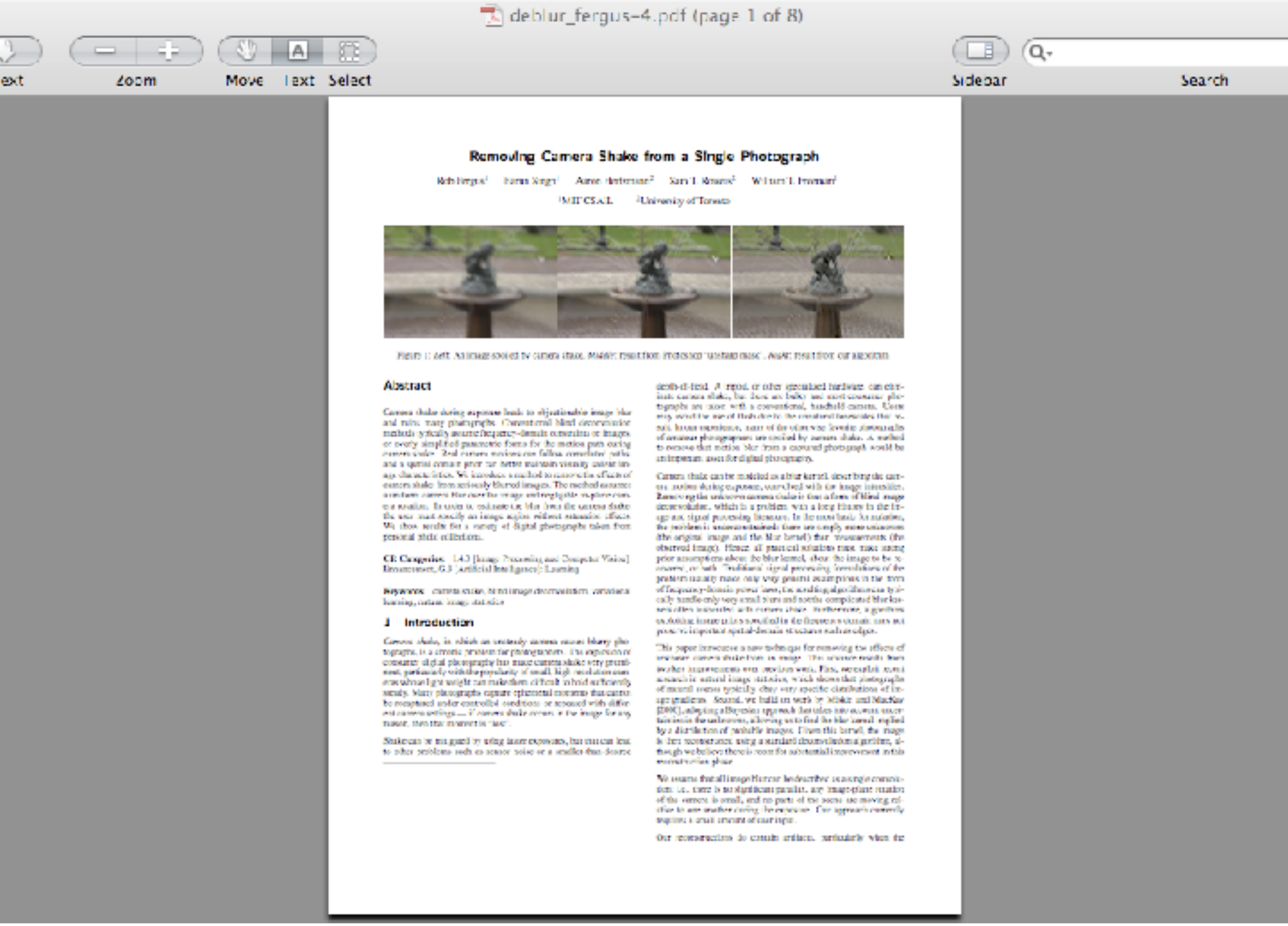
Figure 1: *Left*: An image spoiled by camera shake. *Middle*: result from Photoshop "unsharp mask". *Right*: result from our al

Abstract

Camera shake during exposure leads to objectionable image blur and ruins many photographs. Conventional blind deconvolution methods typically assume frequency-domain constraints on images, or overly simplified parametric forms for the motion path during camera shake. Real camera motions can follow convoluted paths, and a spatial domain prior can better maintain visually salient image characteristics. We introduce a method to remove the effects of camera shake from seriously blurred images. The method assumes a uniform camera blur over the image and negligible in-plane camera rotation. In order to estimate the blur from the camera shake, the user must specify an image region without saturation effects. We show results for a variety of digital photographs taken from

depth of field. A tripod, or other specialized hardwar
inate camera shake, but these are bulky and most co
tographs are taken with a conventional, handheld ca
may avoid the use of flash due to the unnatural tones
sult. In our experience, many of the otherwise favorite
of amateur photographers are spoiled by camera shak
to remove that motion blur from a captured photogra
an important asset for digital photography.

Camera shake can be modeled as a blur kernel, describ
era motion during exposure, convolved with the imag
Removing the unknown camera shake is thus a form of
deconvolution, which is a problem with a long histor
age and signal processing literature. In the most basic
the problem is underconstrained: there are simply mo



Removing Camera Shake from a Single Photograph

Rob Fergus[1] Barun Singh[1] Aaron Hertzmann[2] Sam T. Roweis[2] William T. Freeman[1]

[1]MIT CSAIL [2]University of Toronto

Figure 1: Left: An image spoiled by camera shake. Middle: result from Photoshop "unsharp mask". Right: result from our algorithm.

Abstract

Camera shake during exposure leads to objectionable image blur and ruins many photographs. Conventional blind deconvolution methods typically assume frequency-domain constraints on images, or overly simplified parametric forms for the motion path during camera shake. Real camera motions can follow convoluted paths, and a spatial domain prior can better maintain visually salient image characteristics. We introduce a method to remove the effects of camera shake from seriously blurred images. The method assumes a uniform camera blur over the image and negligible in-plane camera rotation. In order to estimate the blur from the camera shake, the user must specify an image region without saturation effects. We show results for a variety of digital photographs taken from personal photo collections.

CR Categories: I.4.3 [Image Processing and Computer Vision]: Enhancement, G.3 [Artificial Intelligence]: Learning

Keywords: camera shake, blind image deconvolution, variational learning, natural image statistics

1 Introduction

Camera shake, in which an unsteady camera causes blurry photographs, is a chronic problem for photographers. The explosion of consumer digital photography has made camera shake very prominent, particularly with the popularity of small, high-resolution cameras whose light weight can make them difficult to hold sufficiently steady. Many photographs capture ephemeral moments that cannot be recaptured under controlled conditions or repeated with different camera settings — if camera shake occurs in the image for any reason, then that moment is "lost".

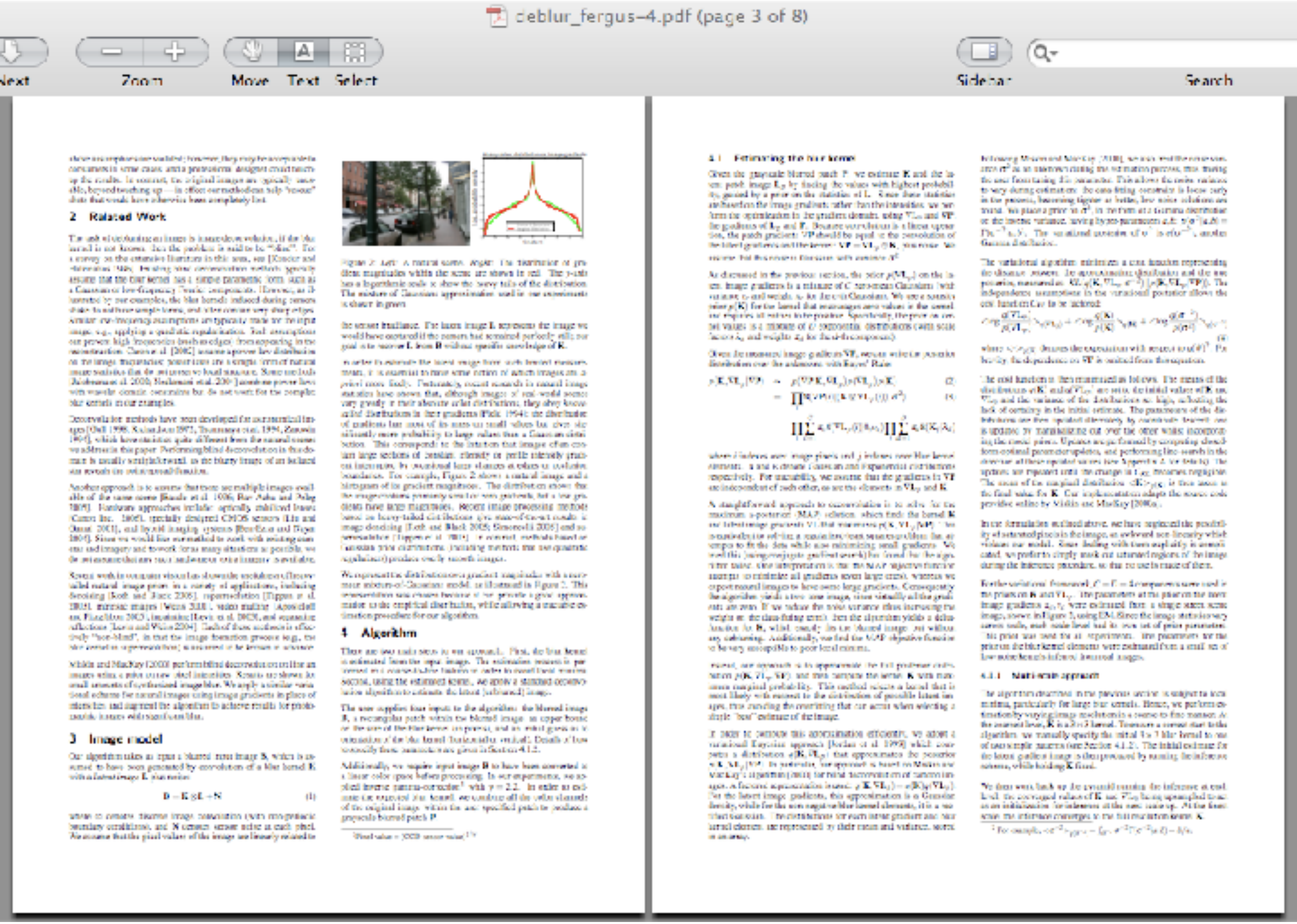
Shake can be mitigated by using faster exposures, but that can lead to other problems such as sensor noise or a smaller-than-desired depth-of-field. A tripod, or other specialized hardware, can eliminate camera shake, but these are bulky and most consumer photographs are taken with a conventional, handheld camera. Users may avoid the use of flash due to the unnatural tonescales that result. In our experience, many of the otherwise favorite photographs of amateur photographers are spoiled by camera shake. A method to remove that motion blur from a captured photograph would be an important asset for digital photography.

Camera shake can be modeled as a blur kernel, describing the camera motion during exposure, convolved with the image intensities. Removing the unknown camera shake is thus a form of blind image deconvolution, which is a problem with a long history in the image and signal processing literature. In the most basic formulation, the problem is underconstrained: there are simply more unknowns (the original image and the blur kernel) than measurements (the observed image). Hence, all practical solutions must make strong prior assumptions about the blur kernel, about the image to be recovered, or both. Traditional signal processing formulations of the problem usually make only very general assumptions in the form of frequency-domain power laws; the resulting algorithms can typically handle only very small blurs and not the complicated blur kernels often associated with camera shake. Furthermore, algorithms exploiting image priors specified in the frequency domain may not preserve important spatial-domain structures such as edges.

This paper introduces a new technique for removing the effects of unknown camera shake from an image. This advance results from two key improvements over previous work. First, we exploit recent research in natural image statistics, which shows that photographs of natural scenes typically obey very specific distributions of image gradients. Second, we build on work by Miskin and MacKay [2000], adopting a Bayesian approach that takes into account uncertainties in the unknowns, allowing us to find the blur kernel implied by a distribution of probable images. Given this kernel, the image is then reconstructed using a standard deconvolution algorithm, although we believe there is room for substantial improvement in this reconstruction phase.

We assume that all image blur can be described as a single convolution; i.e., there is no significant parallax, any image-plane rotation of the camera is small, and no parts of the scene are moving relative to one another during the exposure. Our approach currently requires a small amount of user input.

Our reconstructions do contain artifacts, particularly when the



these are implemented [illegible]; however, they may be acceptable in [illegible] some cases and a professional designer could touch up the results. In contrast, the original images are typically unusable, beyond touching up — in effect our method can help "rescue" shots that would have otherwise been completely lost.

2 Related Work

The task of deconvolution an image is [illegible]. If the blur kernel is not known, then the problem is said to be "blind". For a survey on the extensive literature in this area, see [Kundur and Hatzinakos 1996]. Existing blind deconvolution methods typically assume that the blur kernel has a simple parametric form, such as a Gaussian or low-frequency Fourier components. However, as illustrated by our examples, the blur kernels induced during camera shake do not have simple forms, and often contain very sharp edges. Similar low-frequency assumptions are typically made for the input image, e.g., applying a quadratic regularization. Such assumptions can prevent high frequencies (such as edges) from appearing in the reconstruction. Caron et al. [2002] assume a power-law distribution on the image frequencies; power-laws are a simple form of natural image statistics that do not preserve local structure. Some methods [Jalobeanu et al. 2002; Neelamani et al. 2004] combine power-laws with wavelet domain constraints but do not work for the complex blur kernels in our examples.

Deconvolution methods have been developed for astronomical images [Gull 1998; Richardson 1972; Tsumuraya et al. 1994; Zarowin 1994], which have statistics quite different from the natural scenes we address in this paper. [illegible]

[illegible]

[illegible]

[illegible] small amounts of synthesized image blur. We apply a similar variational scheme for natural images using image gradients in place of [illegible], and augment the algorithm to achieve results for photographic images with significant blur.

3 Image model

Our algorithm takes as input a blurred input image B, which is assumed to have been generated by convolution of a blur kernel K with a latent image L plus noise:

$$\mathbf{B} = \mathbf{K} \otimes \mathbf{L} + \mathbf{N} \tag{1}$$

where ⊗ denotes discrete image convolution (with non-periodic boundary conditions), and N denotes sensor noise at each pixel. We assume that the pixel values of the image are linearly related to

Figure 2: *Left:* A natural scene. *Right:* The distribution of gradient magnitudes within the scene are shown in red. The y-axis has a logarithmic scale to show the heavy tails of the distribution. The mixture of Gaussians approximation used in our experiments is shown in green.

the sensor irradiance. The latent image L represents the image we would have captured if the camera had remained perfectly still; our goal is to recover L from B without specific knowledge of K.

In order to estimate the latent image from such limited measurements, it is essential to have some notion of which images are a priori more likely. Fortunately, recent research in natural image statistics have shown that, although images of real-world scenes vary greatly in their absolute color distributions, they share [illegible]

[illegible]

We represent the distribution over gradient magnitudes with a zero-mean mixture-of-Gaussians model, as illustrated in Figure 2. This representation was chosen because it can provide a good approximation to the empirical distribution, while allowing a tractable estimation procedure for our algorithm.

4 Algorithm

There are two main steps to our approach. First, the blur kernel is estimated from the input image. The estimation process is performed in a coarse-to-fine fashion in order to avoid local minima. Second, using the estimated kernel, we apply a standard deconvolution algorithm to estimate the latent (unblurred) image.

The user supplies four inputs to the algorithm: the blurred image B, a rectangular patch within the blurred image, an upper bound on the size of the blur kernel (in pixels), and an initial guess as to orientation of the blur kernel (horizontal or vertical). Details of how to specify these parameters are given in Section 4.1.2.

Additionally, we require input image B to have been converted to a linear color space before processing. In our experiments, we applied inverse gamma-correction[1] with $\gamma = 2.2$. In order to estimate the expected blur kernel, we combine all the color channels of the original image within the user specified patch to produce a grayscale blurred patch P.

[1] Pixel value = (CCD sensor value)$^{1/\gamma}$

4.1 Estimating the blur kernel

Given the grayscale blurred patch P, we estimate K and the latent patch image $\mathbf{L}_p$ by finding the values with highest probability, guided by a prior on the statistics of L. Since these statistics are based on the image gradients rather than the intensities, we perform the optimization in the gradient domain, using $\nabla\mathbf{L}_p$ and $\nabla\mathbf{P}$, the gradients of $\mathbf{L}_p$ and P. Because convolution is a linear operation, the patch gradients $\nabla\mathbf{P}$ should be equal to the convolution of the latent gradients and the kernel: $\nabla\mathbf{P} = \nabla\mathbf{L}_p \otimes \mathbf{K}$, plus noise. We assume that this noise is Gaussian with variance σ^2.

As discussed in the previous section, the prior $p(\nabla\mathbf{L}_p)$ on the latent image gradients is a mixture of C zero-mean Gaussians (with variance v_c and weight π_c for the c-th Gaussian). We use a sparsity prior $p(\mathbf{K})$ for the kernel that encourages zero values in the kernel, and requires all entries to be positive. Specifically, the prior on kernel values is a mixture of D exponential distributions (with scale factors λ_d and weights π_d for the d-th component).

Given the measured image gradients $\nabla\mathbf{P}$, we can write the posterior distribution over the unknowns with Bayes' Rule:

$$p(\mathbf{K},\nabla\mathbf{L}_p|\nabla\mathbf{P}) \propto p(\nabla\mathbf{P}|\mathbf{K},\nabla\mathbf{L}_p)p(\nabla\mathbf{L}_p)p(\mathbf{K}) \tag{2}$$

$$= \prod_i N(\nabla\mathbf{P}(i)|(\mathbf{K}\otimes\nabla\mathbf{L}_p(i)),\sigma^2) \tag{3}$$

$$\prod_i \sum_{c=1}^{C} \pi_c N(\nabla\mathbf{L}_p(i)|0,v_c) \prod_j \sum_{d=1}^{D} \pi_d E(\mathbf{K}_j|\lambda_d)$$

where i indexes over image pixels and j indexes over blur kernel elements. [illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible] narrow scale, each scale level had its own set of prior parameters. This prior was used for all experiments. The parameters for the prior on the blur kernel elements were estimated from a small set of low-noise kernels inferred from real images.

4.1.1 Multi-scale approach

The algorithm described in the previous section is subject to local minima, particularly for large blur kernels. Hence, we perform estimation by varying image resolution in a coarse-to-fine manner. At the coarsest level, K is a 3×3 kernel. To ensure a correct start to the algorithm, we manually specify the initial 3×3 blur kernel to one of two simple patterns (see Section 4.1.2). The initial estimate for the latent gradient image is then produced by running the inference scheme, while holding K fixed.

We then work back up the pyramid running the inference at each level; the converged values of K and $\nabla\mathbf{L}_p$ being upsampled to act as an initialization for inference at the next scale up. At the finest scale, the inference converges to the full resolution kernel K.

[2] For example, $\langle\sigma^{-2}\rangle_{q(\sigma^{-2})} = \int_{\sigma^{-2}} \sigma^{-2}\,\Gamma(\sigma^{-2}|a,b) = b/a$.

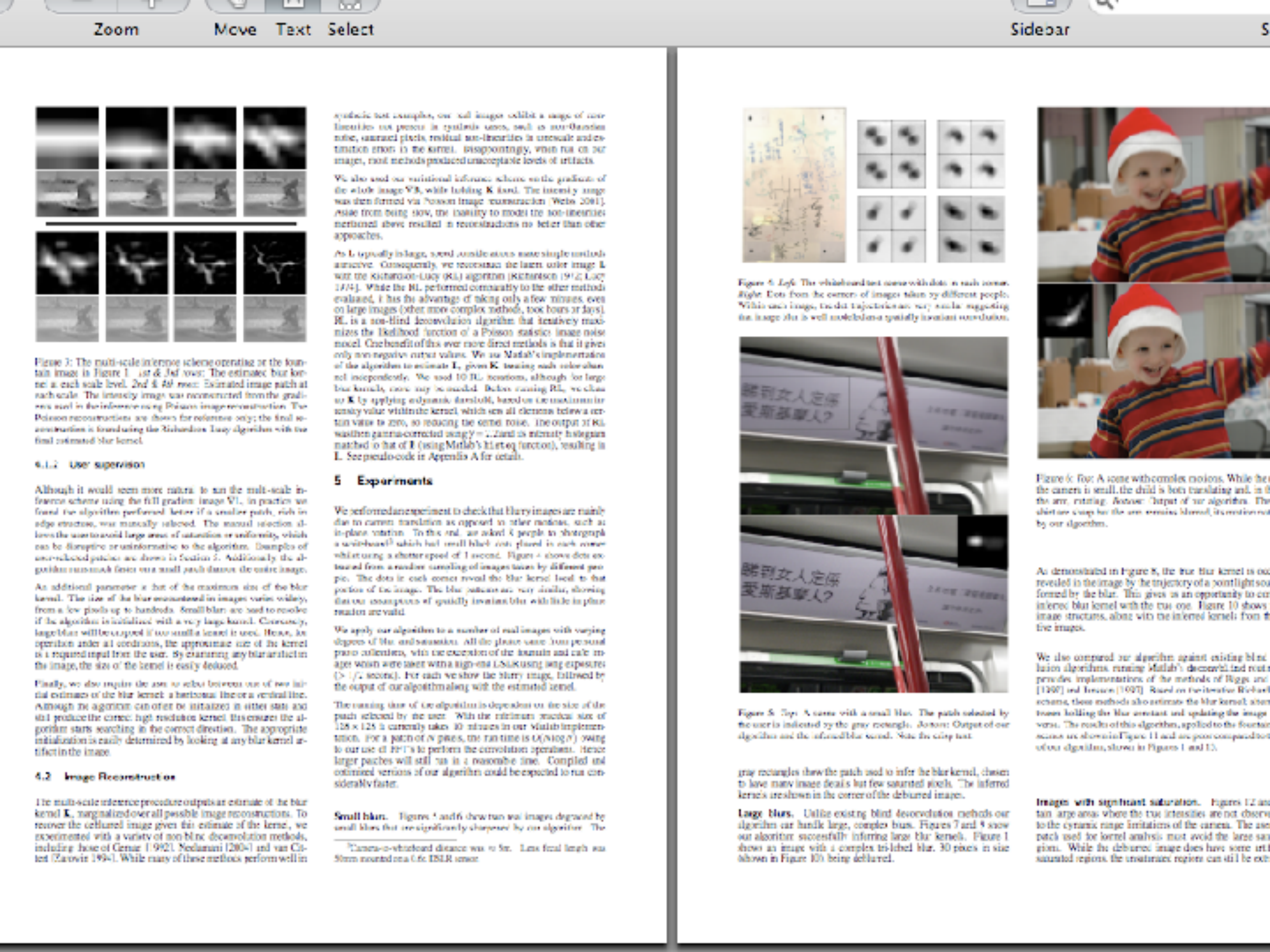


Figure 3: The multi-scale inference scheme operating on the fountain image in Figure 1. *1st & 3rd rows*: The estimated blur kernel at each scale level. *2nd & 4th rows*: Estimated image patch at each scale. The intensity image was reconstructed from the gradients used in the inference using Poisson image reconstruction. The Poisson reconstructions are shown for reference only; the final reconstruction is found using the Richardson-Lucy algorithm with the final estimated blur kernel.

4.1.2 User supervision

Although it would seem more natural to run the multi-scale inference scheme using the full gradient image ∇L, in practice we found the algorithm performed better if a smaller patch, rich in edge structure, was manually selected. The manual selection allows the user to avoid large areas of saturation or uniformity, which can be disruptive or uninformative to the algorithm. Examples of user-selected patches are shown in Section 5. Additionally the algorithm runs much faster on a small patch than on the entire image.

An additional parameter is that of the maximum size of the blur kernel. The size of the blur encountered in images varies widely, from a few pixels up to hundreds. Small blurs are hard to resolve if the algorithm is initialized with a very large kernel. Conversely, large blurs will be cropped if too small a kernel is used. Hence, for operation under all conditions, the approximate size of the kernel is a required input from the user. By examining any blur artifact in the image, the size of the kernel is easily deduced.

Finally, we also require the user to select between one of two initial estimates of the blur kernel: a horizontal line or a vertical line. Although the algorithm can often be initialized in either state and still produce the correct high resolution kernel, this ensures the algorithm starts searching in the correct direction. The appropriate initialization is easily determined by looking at any blur kernel artifact in the image.

4.2 Image Reconstruction

The multi-scale inference procedure outputs an estimate of the blur kernel **K**, marginalized over all possible image reconstructions. To recover the deblurred image given this estimate of the kernel, we experimented with a variety of non-blind deconvolution methods, including those of Geman [1992], Neelamani [2004] and van Cittert [Zarowin 1994]. While many of these methods perform well in synthetic test examples, our real images exhibit a range of non-linearities not present in synthetic cases, such as non-Gaussian noise, saturated pixels, residual non-linearities in tonescale and estimation errors in the kernel. Disappointingly, when run on our images, most methods produced unacceptable levels of artifacts.

We also used our variational inference scheme on the gradients of the whole image $\nabla \mathbf{B}$, while holding **K** fixed. The intensity image was then formed via Poisson image reconstruction [Weiss 2001]. Aside from being slow, the inability to model the non-linearities mentioned above resulted in reconstructions no better than other approaches.

As **L** typically is large, speed considerations make simple methods attractive. Consequently, we reconstruct the latent color image **L** with the Richardson-Lucy (RL) algorithm [Richardson 1972; Lucy 1974]. While the RL performed comparably to the other methods evaluated, it has the advantage of taking only a few minutes, even on large images (other, more complex methods, took hours or days). RL is a non-blind deconvolution algorithm that iteratively maximizes the likelihood function of a Poisson statistics image noise model. One benefit of this over more direct methods is that it gives only non-negative output values. We use Matlab's implementation of the algorithm to estimate **L**, given **K**, treating each color channel independently. We used 10 RL iterations, although for large blur kernels, more may be needed. Before running RL, we clean up **K** by applying a dynamic threshold, based on the maximum intensity value within the kernel, which sets all elements below a certain value to zero, so reducing the kernel noise. The output of RL was then gamma-corrected using $\gamma = 2.2$ and its intensity histogram matched to that of **B** (using Matlab's histeq function), resulting in **L**. See pseudo-code in Appendix A for details.

5 Experiments

We performed an experiment to check that blurry images are mainly due to camera translation as opposed to other motions, such as in-plane rotation. To this end, we asked 8 people to photograph a whiteboard[3] which had small black dots placed in each corner whilst using a shutter speed of 1 second. Figure 4 shows dots extracted from a random sampling of images taken by different people. The dots in each corner reveal the blur kernel local to that portion of the image. The blur patterns are very similar, showing that our assumptions of spatially invariant blur with little in plane rotation are valid.

We apply our algorithm to a number of real images with varying degrees of blur and saturation. All the photos came from personal photo collections, with the exception of the fountain and cafe images which were taken with a high-end DSLR using long exposures (> 1/2 second). For each we show the blurry image, followed by the output of our algorithm along with the estimated kernel.

The running time of the algorithm is dependent on the size of the patch selected by the user. With the minimum practical size of 128×128 it currently takes 10 minutes in our Matlab implementation. For a patch of N pixels, the run-time is $O(N \log N)$ owing to our use of FFT's to perform the convolution operations. Hence larger patches will still run in a reasonable time. Compiled and optimized versions of our algorithm could be expected to run considerably faster.

Small blurs. Figures 5 and 6 show two real images degraded by small blurs that are significantly sharpened by our algorithm. The gray rectangles show the patch used to infer the blur kernel, chosen to have many image details but few saturated pixels. The inferred kernels are shown in the corner of the deblurred images.

Large blurs. Unlike existing blind deconvolution methods our algorithm can handle large, complex blurs. Figures 7 and 9 show our algorithm successfully inferring large blur kernels. Figure 1 shows an image with a complex tri-lobed blur, 30 pixels in size (shown in Figure 10) being deblurred.

[3] Camera-to-whiteboard distance was ≈ 5m. Lens focal length was 50mm mounted on a 0.6x DSLR sensor.

Figure 4: *Left*: The whiteboard test scene with dots in each corner. *Right*: Dots from the corners of images taken by different people. Within each image, the dot trajectories are very similar suggesting that image blur is well modeled as a spatially invariant convolution.

Figure 5: *Top*: A scene with a small blur. The patch selected by the user is indicated by the gray rectangle. *Bottom*: Output of our algorithm and the inferred blur kernel. Note the crisp text.

Figure 6: *Top*: A scene with complex motions. While the
the camera is small, the child is both translating and, in t
the arm, rotating. *Bottom*: Output of our algorithm. The
shirt are sharp but the arm remains blurred, its motion not
by our algorithm.

As demonstrated in Figure 8, the true blur kernel is occ
revealed in the image by the trajectory of a point light sou
formed by the blur. This gives us an opportunity to co
inferred blur kernel with the true one. Figure 10 shows
image structures, along with the inferred kernels from th
five images.

We also compared our algorithm against existing blind
lution algorithms, running Matlab's deconvblind rout
provides implementations of the methods of Biggs and
[1997] and Jansson [1997]. Based on the iterative Richardl
scheme, these methods also estimate the blur kernel; alter
tween holding the blur constant and updating the image
versa. The results of this algorithm, applied to the fountai
scenes are shown in Figure 11 and are poor compared to t
of our algorithm, shown in Figures 1 and 13.

Images with significant saturation. Figures 12 an
tain large areas where the true intensities are not observ
to the dynamic range limitations of the camera. The use
patch used for kernel analysis must avoid the large sat
gions. While the deblurred image does have some art
saturated regions, the unsaturated regions can still be ext

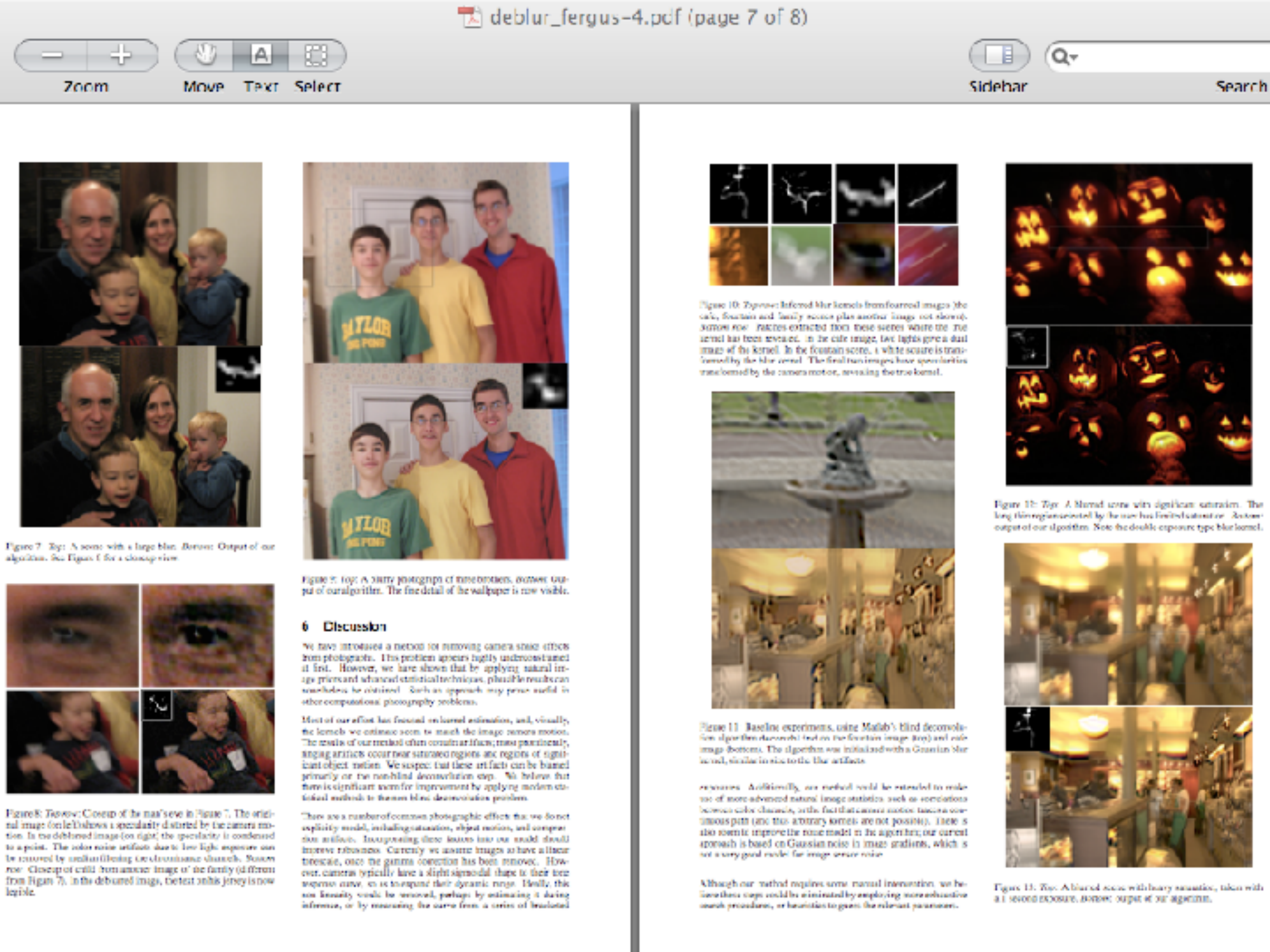


Figure 7: *Top:* A scene with a large blur. *Bottom:* Output of our algorithm. See Figure 8 for a closeup view.

Figure 8: *Top row:* Closeup of the man's eye in Figure 7. The original image (on left) shows a specularity distorted by the camera motion. In the deblurred image (on right) the specularity is condensed to a point. The color noise artifacts due to low light exposure can be removed by median filtering the chrominance channels. *Bottom row:* Closeup of child from another image of the family (different from Figure 7). In the deblurred image, the text on his jersey is now legible.

Figure 9: *Top:* A blurry photograph of three brothers. *Bottom:* Output of our algorithm. The fine detail of the wallpaper is now visible.

6 Discussion

We have introduced a method for removing camera shake effects from photographs. This problem appears highly underconstrained at first. However, we have shown that by applying natural image priors and advanced statistical techniques, plausible results can nonetheless be obtained. Such an approach may prove useful in other computational photography problems.

Most of our effort has focused on kernel estimation, and, visually, the kernels we estimate seem to match the image camera motion. The results of our method often contain artifacts; most prominently, ringing artifacts occur near saturated regions and regions of significant object motion. We suspect that these artifacts can be blamed primarily on the non-blind deconvolution step. We believe that there is significant room for improvement by applying modern statistical methods to the non-blind deconvolution problem.

There are a number of common photographic effects that we do not explicitly model, including saturation, object motion, and compression artifacts. Incorporating these factors into our model should improve robustness. Currently we assume images to have a linear tonescale, once the gamma correction has been removed. However, cameras typically have a slight sigmoidal shape to their tone response curve, so as to expand their dynamic range. Ideally, this non-linearity would be removed, perhaps by estimating it during inference, or by measuring the curve from a series of bracketed exposures. Additionally, our method could be extended to make use of more advanced natural image statistics, such as correlations between color channels, or the fact that camera motion traces a continuous path (and thus arbitrary kernels are not possible). There is also room to improve the noise model in the algorithm; our current approach is based on Gaussian noise in image gradients, which is not a very good model for image sensor noise.

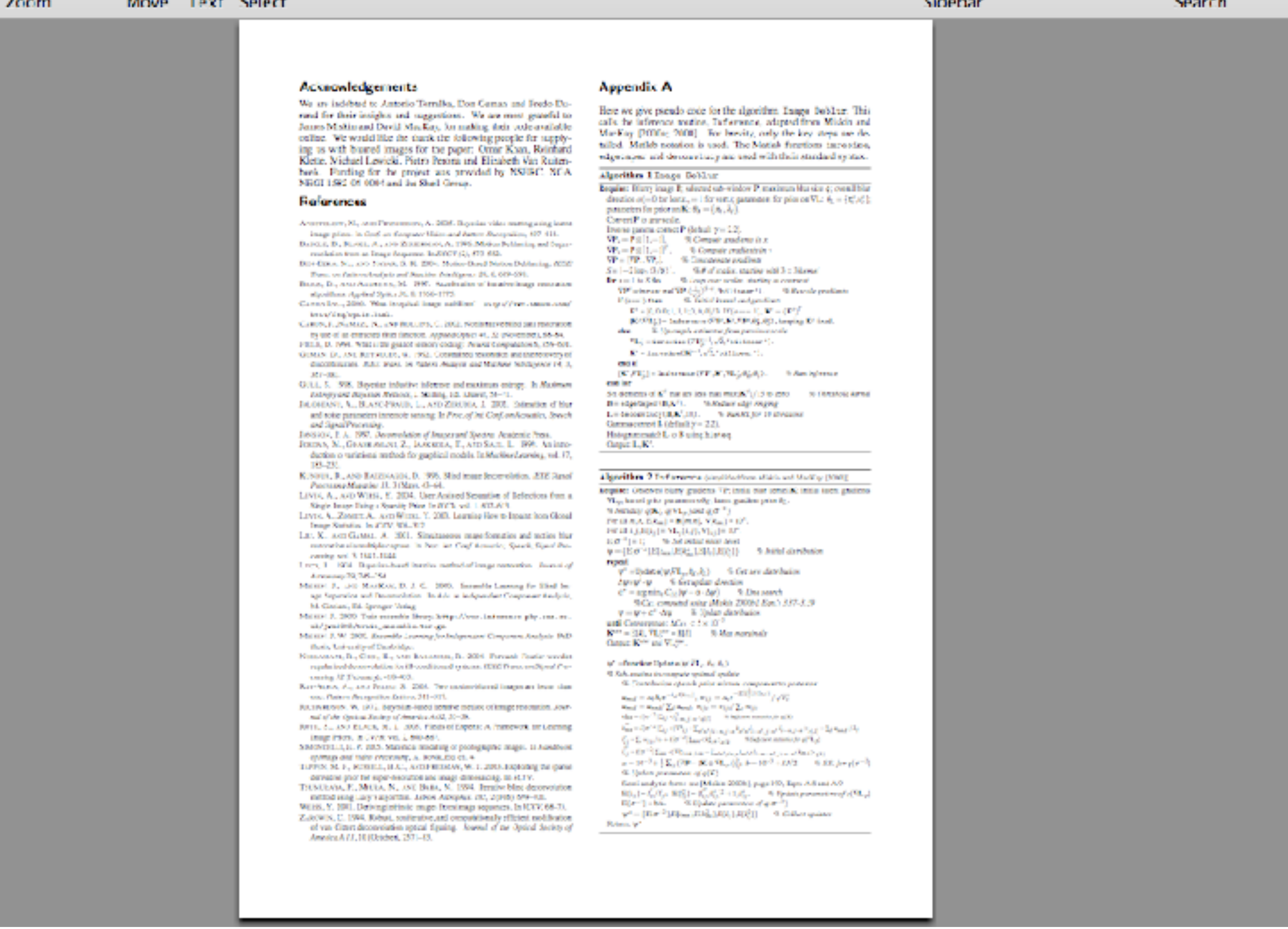
Although our method requires some manual intervention, we believe these steps could be eliminated by employing more exhaustive search procedures, or heuristics to guess the relevant parameters.

Figure 10: *Top row:* Inferred blur kernels from four real images (the cafe, fountain and family scenes plus another image not shown). *Bottom row:* Patches extracted from these scenes where the true kernel has been revealed. In the cafe image, two lights give a dual image of the kernel. In the fountain scene, a white square is transformed by the blur kernel. The final two images have specularities transformed by the camera motion, revealing the true kernel.

Figure 11: Baseline experiments, using Matlab's blind deconvolution algorithm deconvblind on the fountain image (top) and cafe image (bottom). The algorithm was initialized with a Gaussian blur kernel, similar in size to the blur artifacts.

Figure 12: *Top:* A blurred scene with significant saturation. The long thin region selected by the user has limited saturation. *Bottom:* output of our algorithm. Note the double exposure type blur kernel.

Figure 13: *Top:* A blurred scene with heavy saturation, taken with a 1 second exposure. *Bottom:* output of our algorithm.



Acknowledgements

We are indebted to Antonio Torralba, Don Geman and Fredo Durand for their insights and suggestions. We are most grateful to James Miskin and David MacKay, for making their code available online. We would like the thank the following people for supplying us with blurred images for the paper: Omar Khan, Reinhard Klette, Michael Lewicki, Pietro Perona and Elizabeth Van Ruitenbeek. Funding for the project was provided by NSERC, NGA NEGI-1582-04-0004 and the Shell Group.

References

Apostoloff, N., and Fitzgibbon, A. 2005. Bayesian video matting using learnt image priors. In Conf. on Computer Vision and Pattern Recognition, 407–414.

Bascle, B., Blake, A., and Zisserman, A. 1996. Motion Deblurring and Super-resolution from an Image Sequence. In ECCV (2), 573–582.

Ben-Ezra, M., and Nayar, S. K. 2004. Motion-Based Motion Deblurring. IEEE Trans. on Pattern Analysis and Machine Intelligence 26, 6, 689–698.

Biggs, D., and Andrews, M. 1997. Acceleration of iterative image restoration algorithms. Applied Optics 36, 8, 1766–1775.

Canon Inc., 2006. What is optical image stabilizer? http://www.canon.com/bctv/faq/optis.html.

Caron, J., Namazi, N., and Rollins, C. 2002. Noniterative blind data restoration by use of an extracted filter function. Applied Optics 41, 32 (November), 68–84.

Field, D. 1994. What is the goal of sensory coding? Neural Computation 6, 559–601.

Geman, D., and Reynolds, G. 1992. Constrained restoration and the recovery of discontinuities. IEEE Trans. on Pattern Analysis and Machine Intelligence 14, 3, 367–383.

Gull, S. 1998. Bayesian inductive inference and maximum entropy. In Maximum Entropy and Bayesian Methods, J. Skilling, Ed. Kluwer, 54–71.

Jalobeanu, A., Blanc-Féraud, L., and Zerubia, J. 2002. Estimation of blur and noise parameters in remote sensing. In Proc. of Int. Conf. on Acoustics, Speech and Signal Processing.

Jansson, P. A. 1997. Deconvolution of Image and Spectra. Academic Press.

Jordan, M., Ghahramani, Z., Jaakkola, T., and Saul, L. 1999. An introduction to variational methods for graphical models. In Machine Learning, vol. 37, 183–233.

Kundur, D., and Hatzinakos, D. 1996. Blind image deconvolution. IEEE Signal Processing Magazine 13, 3 (May), 43–64.

Levin, A., and Weiss, Y. 2004. User Assisted Separation of Reflections from a Single Image Using a Sparsity Prior. In ECCV, vol. 1, 602–613.

Levin, A., Zomet, A., and Weiss, Y. 2003. Learning How to Inpaint from Global Image Statistics. In ICCV, 305–312.

Liu, X., and Gamal, A. 2001. Simultaneous image formation and motion blur restoration via multiple capture. In Proc. Int. Conf. Acoustics, Speech, Signal Processing, vol. 3, 1841–1844.

Lucy, L. 1974. Bayesian-based iterative method of image restoration. Journal of Astronomy 79, 745–754.

Miskin, J., and MacKay, D. J. C. 2000. Ensemble Learning for Blind Image Separation and Deconvolution. In Adv. in Independent Component Analysis, M. Girolani, Ed. Springer-Verlag.

Miskin, J., 2000. Train ensemble library. http://www.inference.phy.cam.ac.uk/jwm1003/train_ensemble.tar.gz.

Miskin, J. W. 2000. Ensemble Learning for Independent Component Analysis. PhD thesis, University of Cambridge.

Neelamani, R., Choi, H., and Baraniuk, R. 2004. Forward: Fourier-wavelet regularized deconvolution for ill-conditioned systems. IEEE Trans. on Signal Processing 52 (February), 418–433.

Rav-Acha, A., and Peleg, S. 2005. Two motion-blurred images are better than one. Pattern Recognition Letters, 311–317.

Richardson, W. 1972. Bayesian-based iterative method of image restoration. Journal of the Optical Society of America A 62, 55–59.

Roth, S., and Black, M. J. 2005. Fields of Experts: A Framework for Learning Image Priors. In CVPR, vol. 2, 860–867.

Simoncelli, E. P. 2005. Statistical modeling of photographic images. In Handbook of Image and Video Processing, A. Bovik, Ed. ch. 4.

Tappen, M. F., Russell, B. C., and Freeman, W. T. 2003. Exploiting the sparse derivative prior for super-resolution and image demosaicing. In SCTV.

Tsumuraya, F., Miura, N., and Baba, N. 1994. Iterative blind deconvolution method using Lucy's algorithm. Astron. Astrophys. 282, 2 (Feb), 599–618.

Weiss, Y. 2001. Deriving intrinsic images from image sequences. In ICCV, 68–75.

Zarowin, C. 1994. Robust, noniterative, and computationally efficient modification of van Cittert deconvolution optical figuring. Journal of the Optical Society of America A 11, 10 (October), 2571–83.

Appendix A

Here we give pseudo code for the algorithm, Image Deblur. This calls the inference routine, Inference, adapted from Miskin and MacKay [2000a; 2000]. For brevity, only the key steps are detailed. Matlab notation is used. The Matlab functions imresize, edgetaper and deconvlucy are used with their standard syntax.

Algorithm 1 Image Deblur

```
Require: Blurry image $\mathbf{B}$; selected sub-window $\mathbf{P}$; maximum blur size $\phi$; overall blur direction $o$ (= 0 for horiz., = 1 for vert.); parameters for prior on $\nabla\mathbf{L}$: $\theta_L = \{\pi_c^s, v_c^s\}$; parameters for prior on $\mathbf{K}$: $\theta_K = \{\pi_d, \lambda_d\}$.
  Convert $\mathbf{P}$ to grayscale.
  Inverse gamma correct $\mathbf{P}$ (default $\gamma = 2.2$).
  $\nabla\mathbf{P}_x = \mathbf{P} \otimes [1, -1]$,   % Compute gradients in $x$
  $\nabla\mathbf{P}_y = \mathbf{P} \otimes [1, -1]^T$,   % Compute gradients in $y$
  $\nabla\mathbf{P} = [\nabla\mathbf{P}_x, \nabla\mathbf{P}_y]$.   % Concatenate gradients
  $S = \lceil -2\log_2(3/\phi) \rceil$.   % # of scales, starting with 3 × 3 kernel
  for $s = 1$ to $S$ do   % Loop over scales, starting at coarsest
    $\nabla\mathbf{P}^s \leftarrow$ imresize($\nabla\mathbf{P}$, $(\frac{1}{\sqrt{2}})^{S-s}$, 'bilinear').   % Rescale gradients
    if ($s == 1$) then   % Initial kernel and gradients
      $\mathbf{K}^s \leftarrow [0, 0, 0; 1, 1, 1; 0, 0, 0]/3$. If ($o == 1$), $\mathbf{K}^s = (\mathbf{K}^s)^T$.
      $[\mathbf{K}^s, \nabla\mathbf{L}_p^s] \leftarrow$ Inference($\nabla\mathbf{P}^s$, $\mathbf{K}^s$, $\nabla\mathbf{P}^s$, $\theta_K^s$, $\theta_L^s$), keeping $\mathbf{K}^s$ fixed.
    else   % Upsample estimates from previous scale
      $\nabla\mathbf{L}_p^s \leftarrow$ imresize($\nabla\mathbf{L}_p^{s-1}$, $\sqrt{2}$, 'bilinear').
      $\mathbf{K}^s \leftarrow$ imresize($\mathbf{K}^{s-1}$, $\sqrt{2}$, 'bilinear').
    end if
    $[\mathbf{K}^s, \nabla\mathbf{L}_p^s] \leftarrow$ Inference($\nabla\mathbf{P}^s$, $\mathbf{K}^s$, $\nabla\mathbf{L}_p^s$, $\theta_K^s$, $\theta_L^s$).   % Run inference
  end for
  Set elements of $\mathbf{K}^S$ that are less than max($\mathbf{K}^S$)/15 to zero.   % Threshold kernel
  $\mathbf{B}$ = edgetaper($\mathbf{B}$, $\mathbf{K}^S$).   % Reduce edge ringing
  $\mathbf{L}$ = deconvlucy($\mathbf{B}$, $\mathbf{K}^S$, 10).   % Run RL for 10 iterations
  Gamma correct $\mathbf{L}$ (default $\gamma = 2.2$).
  Histogram match $\mathbf{L}$ to $\mathbf{B}$ using histeq.
  Output: $\mathbf{L}$, $\mathbf{K}^S$.
```

Algorithm 2 Inference (simplified from Miskin and MacKay [2000])

```
Require: Observed blurry gradients $\nabla\mathbf{P}$; initial blur kernel $\mathbf{K}$; initial latent gradients $\nabla\mathbf{L}_p$; kernel prior parameters $\theta_K$; latent gradient prior $\theta_L$.
  % Initialize $q(\mathbf{K})$, $q(\nabla\mathbf{L}_p)$ and $q(\sigma^{-2})$
  For all $m, n$, $E[k_{mn}] = \mathbf{K}(m, n)$, $V[k_{mn}] = 10^4$.
  For all $i, j$, $E[l_{ij}] = \nabla\mathbf{L}_p(i, j)$, $V[l_{ij}] = 10^4$.
  $E[\sigma^{-2}] = 1$;   % Set initial noise level
  $\psi = \{E[\sigma^{-2}], E[k_{mn}], E[k_{mn}^2], E[l_{ij}], E[l_{ij}^2]\}$   % Initial distribution
  repeat
    $\psi^* = $ Update($\psi$, $\nabla\mathbf{L}_p$, $\theta_K$, $\theta_L$)   % Get new distribution
    $\Delta\psi = \psi^* - \psi$   % Get update direction
    $\alpha^* = \arg\min_\alpha C_{KL}(\psi + \alpha \cdot \Delta\psi)$   % Line search
      % $C_{KL}$ computed using [Miskin 2000b], Eqn.'s 3.37–3.39
    $\psi = \psi + \alpha^* \cdot \Delta\psi$   % Update distribution
  until Convergence: $\Delta C_{KL} < 5 \times 10^{-3}$
  $\mathbf{K}^{new} = E[k]$, $\nabla\mathbf{L}_p^{new} = E[l]$.   % Max marginals
  Output: $\mathbf{K}^{new}$ and $\nabla\mathbf{L}_p^{new}$.

  $\psi^*$ = Function Update($\psi$, $\nabla\mathbf{L}_p$, $\theta_K$, $\theta_L$)
  % Sub-routine to compute optimal update
    % Contribution of each prior mixture component to posterior
    $u_{mnd} = \pi_d \lambda_d e^{-\lambda_d E[k_{mn}]}$; $w_{ijc} = \pi_c e^{-(E[l_{ij}^2]/(2v_c))}/\sqrt{v_c}$
    $u_{mnd} = u_{mnd}/\sum_d u_{mnd}$; $w_{ijc} = w_{ijc}/\sum_c w_{ijc}$
    $k'_{mn} = E[\sigma^{-2}] \sum_{ij} \langle l_{i-m,j-n}^2 \rangle_{q(l)}$   % Sufficient statistics for $q(\mathbf{K})$
    $k''_{mn} = E[\sigma^{-2}] \sum_{ij} \langle (\nabla P_{ij} - \sum_{m'n' \neq i-m,j-n} k_{m'n'} l_{i-m',j-n'}) l_{i-m,j-n} \rangle_{q(l)} - \sum_d u_{mnd} \lambda_d$
    $l'_{ij} = \sum_c w_{ijc}/v_c + E[\sigma^{-2}] \sum_{mn} \langle k_{mn}^2 \rangle_{q(k)}$   % Sufficient statistics for $q(\nabla\mathbf{L}_p)$
    $l''_{ij} = E[\sigma^{-2}] \sum_{mn} \langle (\nabla P_{i+m,j+n} - \sum_{m'n' \neq m,n} k_{m'n'} l_{i+m-m',j+n-n'}) k_{mn} \rangle_{q(k)}$
    $a = 10^{-3} + \frac{1}{2} \sum_{ij} (\nabla\mathbf{P} - (\mathbf{K} \otimes \nabla\mathbf{L}_p))_{ij}^2$; $b = 10^{-3} + IJ/2$   % S.S. for $q(\sigma^{-2})$
    % Update parameters of $q(\mathbf{K})$
    Semi-analytic form: see [Miskin 2000b], page 199, Eqns A.8 and A.9
    $E[l_{ij}] = l''_{ij}/l'_{ij}$; $E[l_{ij}^2] = (l''_{ij}/l'_{ij})^2 + 1/l'_{ij}$.   % Update parameters of $q(\nabla\mathbf{L}_p)$
    $E[\sigma^{-2}] = b/a$.   % Update parameters of $q(\sigma^{-2})$
    $\psi^* = \{E[\sigma^{-2}], E[k_{mn}], E[k_{mn}^2], E[l_{ij}], E[l_{ij}^2]\}$   % Collect updates
  Return $\psi^*$
```

The introduction

1 Introduction

2 Related work

3 --Main idea--

4 Algorithm

- Estimating the blur kernel
  - Multi-scale approach
  - User supervision
- Image reconstruction

5 Experiments

- Small blurs
- Large blurs
- Images with significant saturation

6 Discussion

Jim Kajiya: write a dynamite introduction

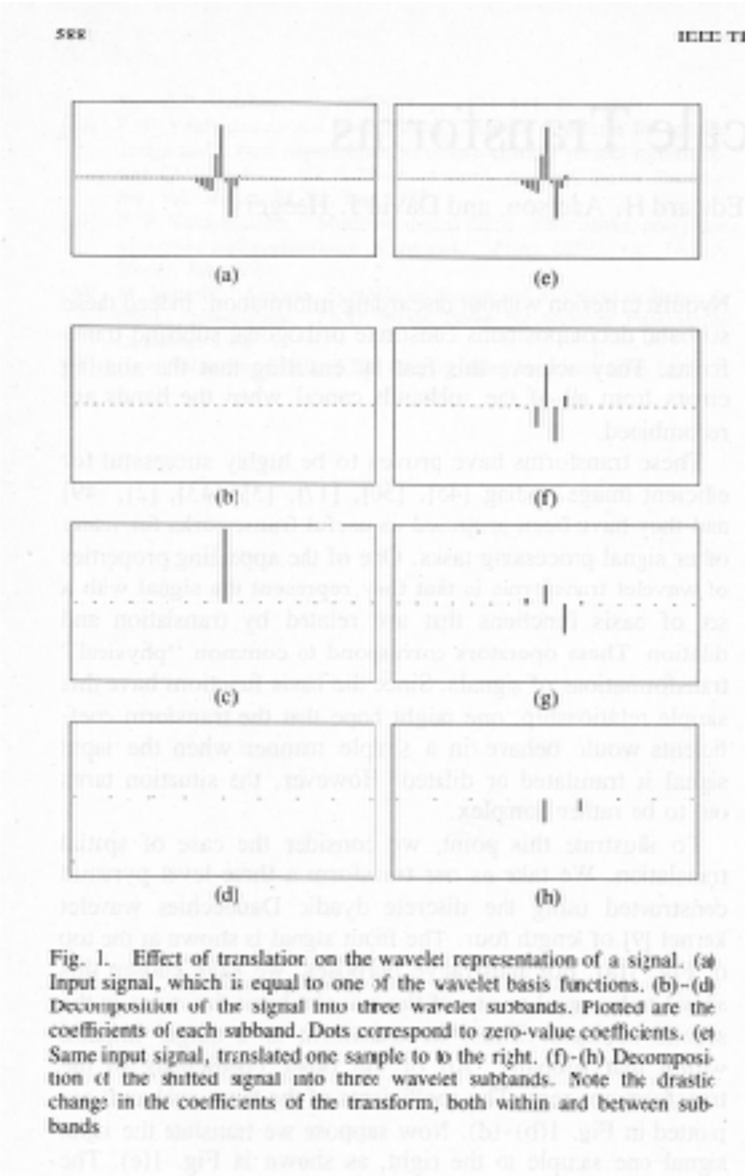
You must make your paper easy to read. You've got to make it easy for anyone to tell what your paper is about, what problem it solves, why the problem is interesting, what is really new in your paper (and what isn't), why it's so neat. And you must do it up front. In other words, you must write a dynamite introduction.

Underutilized technique: explain the main idea with a simple, toy example.

- 1 Introduction
- 2 Related work
- 3 Main idea
- 4 Algorithm
  - Estimating the blur kernel
    - Multi-scale approach
    - User supervision
  - Image reconstruction
- 5 Experiments
  - Small blurs
  - Large blurs
  - Images with significant saturation
- 6 Discussion

Often useful here.

Show simple toy examples to let people get the main idea



588

IEEE T

Fig. 1. Effect of translation on the wavelet representation of a signal. (a) Input signal, which is equal to one of the wavelet basis functions. (b)-(d) Decomposition of the signal into three wavelet subbands. Plotted are the coefficients of each subband. Dots correspond to zero-value coefficients. (e) Same input signal, translated one sample to to the right. (f)-(h) Decomposition of the shifted signal into three wavelet subbands. Note the drastic change in the coefficients of the transform, both within and between subbands

From "Shiftable multiscale transforms"

Steerable filters simple example

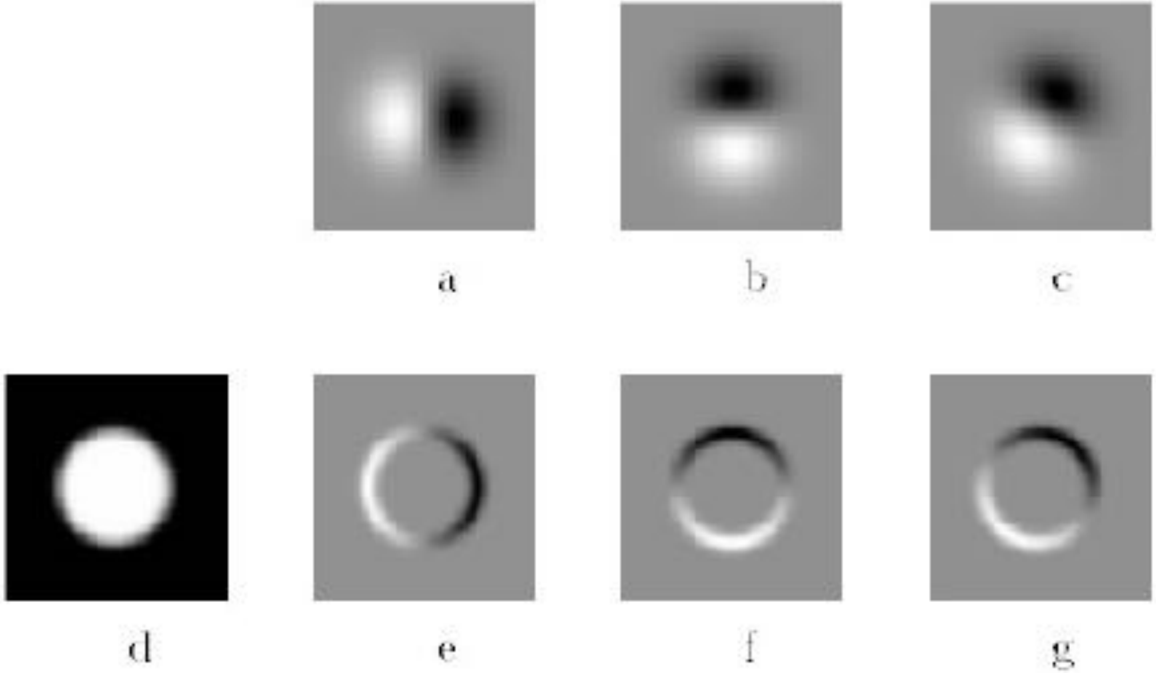


Fig. 1. Example of steerable filters: (a) $G_1^{0^\circ}$ first derivative with respect to x (horizontal) of a Gaussian; (b) $G_1^{90^\circ}$, which is $G_1^{0^\circ}$, rotated by 90°. From a linear combination of these two filters, one can create G_1^{θ}, which is an arbitrary rotation of the first derivative of a Gaussian: (c) $G_1^{60^\circ}$, formed by $\frac{1}{2}G_1^{0^\circ} + \frac{\sqrt{3}}{2}G_1^{90^\circ}$. The same linear combinations used to synthesize G_1^{θ} from the basis filters will also synthesize the response of an image to G_1^{θ} from the responses of the image to the basis filters; (d) image of circular disk; (e) G^{0° (at a smaller scale than pictured above) convolved with the disk (d); (f) $G_1^{90^\circ}$ convolved with (d); (g) $G_1^{60^\circ}$ convolved with (d), obtained from $\frac{1}{2}$ (image (e)) + $\frac{\sqrt{3}}{2}$ (image (f)).

Experimental results are critical now at CVPR

1 Introduction

2 Related work

3 Image model

4 Algorithm

- Estimating the blur kernel
  - Multi-scale approach
  - User supervision
- Image reconstruction

5 Experiments

- Small blurs
- Large blurs
- Images with significant saturation

6 Discussion

Gone are the days of, "We think this is a great idea and we expect it will be very useful in computer vision. See how it works on this meaningless, contrived problem?"

Experimental results from Fergus et al paper

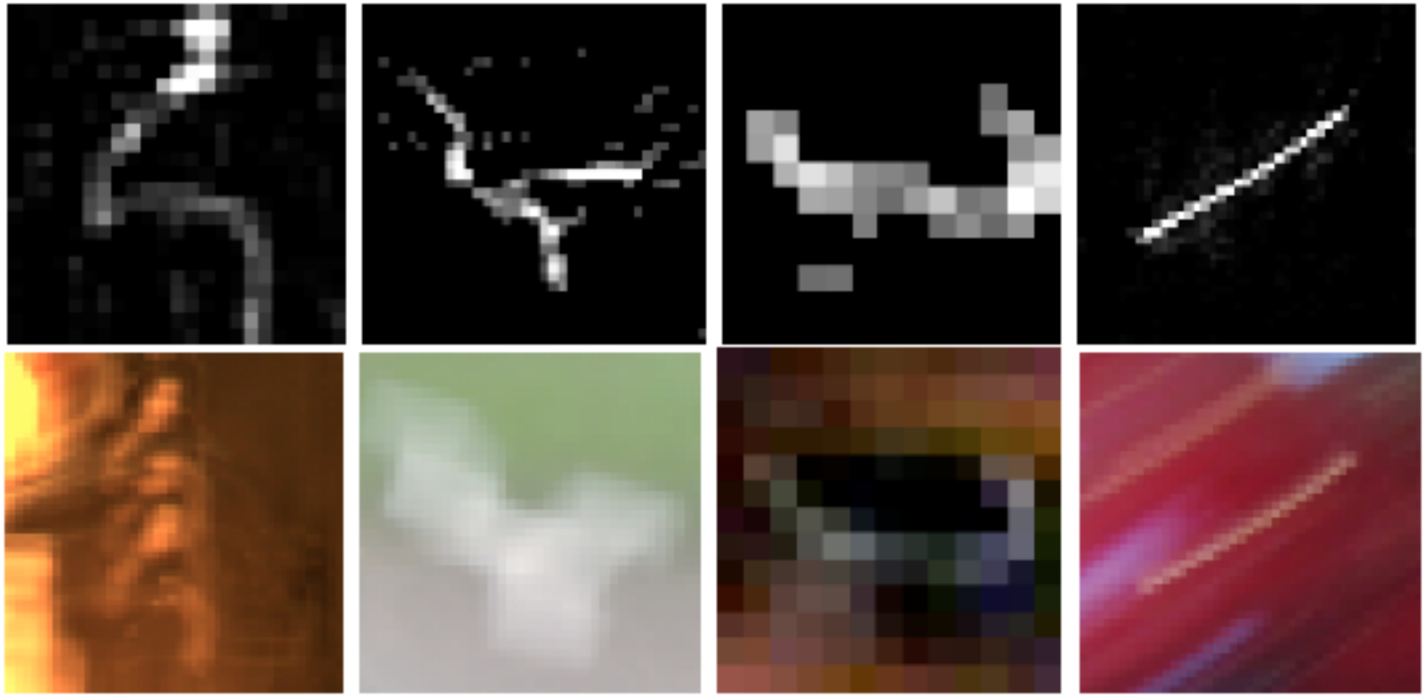


Figure 10: *Top row*: Inferred blur kernels from four real images (the cafe, fountain and family scenes plus another image not shown). *Bottom row*: Patches extracted from these scenes where the true kernel has been revealed. In the cafe image, two lights give a dual image of the kernel. In the fountain scene, a white square is trans formed by the blur kernel. The final two images have specularities transformed by the camera motion, revealing the true kernel.

Experimental results from a later deblurring paper

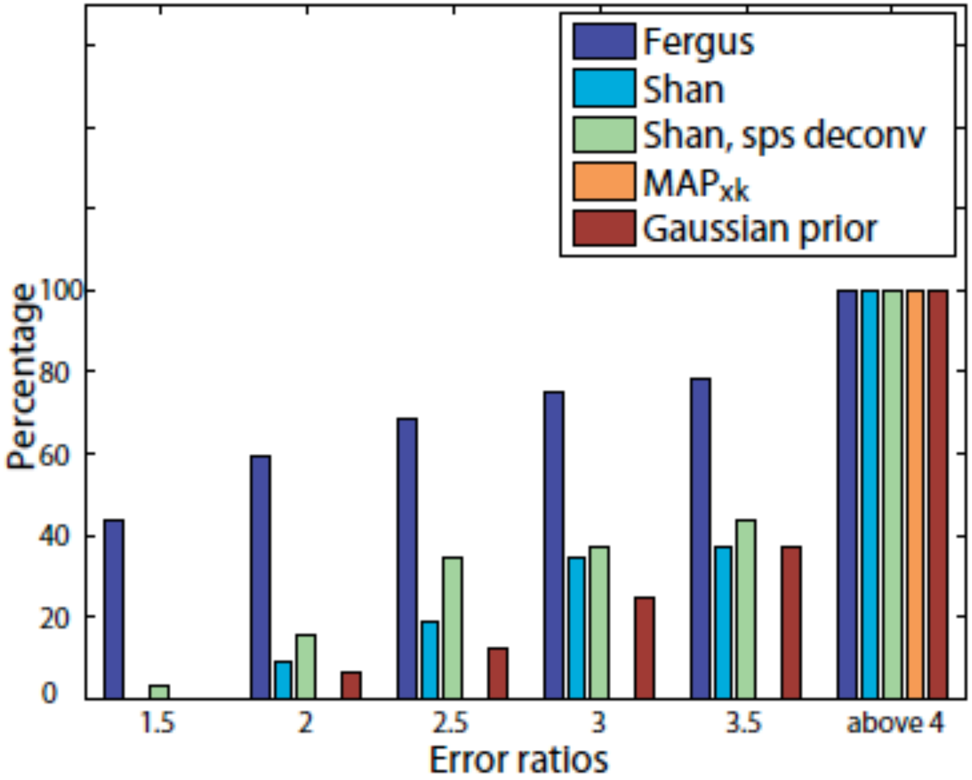


Figure 9. Evaluation results: Cumulative histogram of the deconvolution error ratio across test examples.

How to end a paper

1 Introduction

2 Related work

3 Image model

4 Algorithm

- Estimating the blur kernel
  - Multi-scale approach
  - User supervision
- Image reconstruction

5 Experiments

- Small blurs
- Large blurs
- Images with significant saturation

6 Discussion

Conclusions, or what this opens up, or how this can change how we approach computer vision problems.

How not to end a paper

1 Introduction
2 Related work
3 Image model
4 Algorithm
    Estimating the blur kernel
        Multi-scale approach
        User supervision
    Image reconstruction
5 Experiments
    Small blurs
    Large blurs
    Images with saturation
6 Discussion
Future work?

I can't stand "future work" sections. It's hard to think of a weaker way to end a paper.

"Here's a list all the ideas we wanted to do but couldn't get to work in time for the conference submission deadline. We didn't do any of the following things: (1)..."

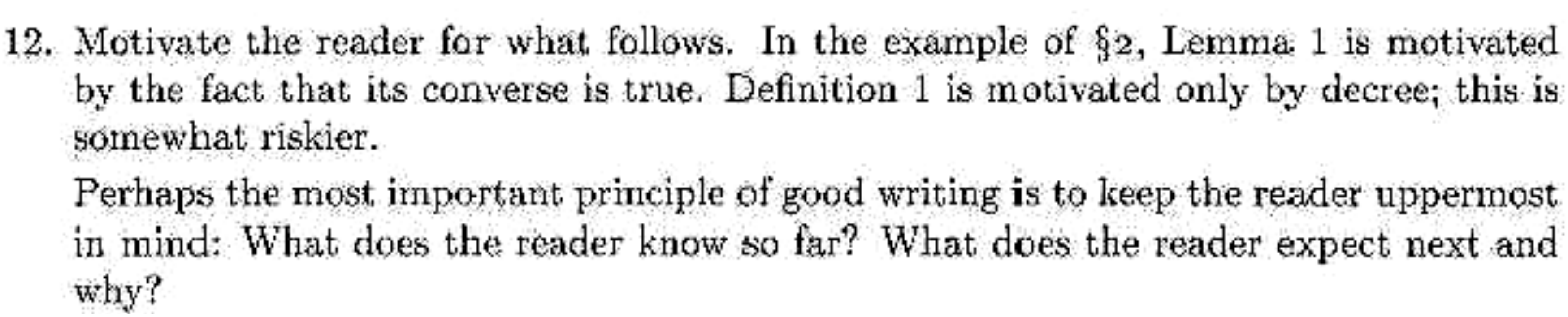
(You get no "partial credit" from reviewers and readers for neat things you wanted to do, but didn't.)

"Here's a list of good ideas that you should now go and do before we get a chance."

Better to end with a conclusion or a summary, or you can say in general terms where the work may lead.

General writing tips

Knuth: keep the reader upper-most in your mind.



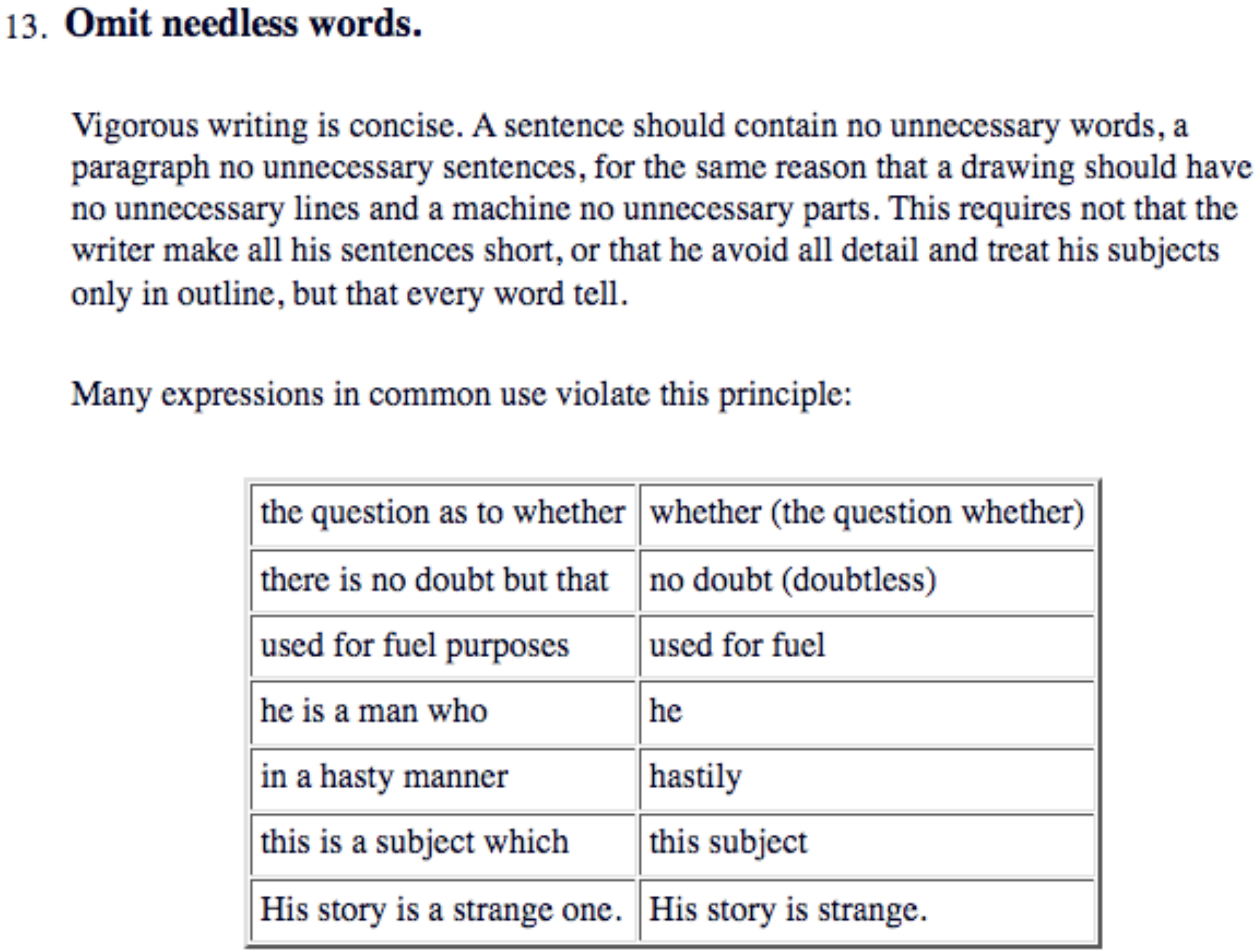
12. Motivate the reader for what follows. In the example of §2, Lemma 1 is motivated by the fact that its converse is true. Definition 1 is motivated only by decree; this is somewhat riskier.

Perhaps the most important principle of good writing is to keep the reader uppermost in mind: What does the reader know so far? What does the reader expect next and why?

Treat the reader as you would a guest in your house

Anticipate their needs: would you like something to drink? Something to eat? Perhaps now, after eating, you'd like to rest?

Writing style, from the elements of style, Stunk and White



13. **Omit needless words.**

Vigorous writing is concise. A sentence should contain no unnecessary words, a paragraph no unnecessary sentences, for the same reason that a drawing should have no unnecessary lines and a machine no unnecessary parts. This requires not that the writer make all his sentences short, or that he avoid all detail and treat his subjects only in outline, but that every word tell.

Many expressions in common use violate this principle:

the question as to whether	whether (the question whether)
there is no doubt but that	no doubt (doubtless)
used for fuel purposes	used for fuel
he is a man who	he
in a hasty manner	hastily
this is a subject which	this subject
His story is a strange one.	His story is strange.

Re-writing exercise

Text from a CVPR Workshop paper I’m co-author on.

The underlying assumption of this work is that the estimate of a given node will only depend on nodes within a patch: this is a locality assumption imposed at the patch-level. This assumption can be justified in case of skin images since a pixel in one corner of the image is likely to have small effect on a different pixel far away from itself. Therefore, we can crop the image into smaller windows, as shown in Figure 5, and compute the inverse J matrix of the cropped window. Since the cropped window is much smaller than the input image, the inversion of J matrix is computationally cheaper. Since we are inferring on blocks of image patches (i.e. ignoring pixels outside of the cropped window), the interpolated image will have blocky artifacts. Therefore, only part of xMAP is used to interpolate the image, as shown in Figure 5.

Re-writing exercise

Original:

The underlying assumption of this work is that the estimate of a given node will only depend on nodes within a patch: this is a locality assumption imposed at the patch-level. This assumption can be justified in case of skin images since a pixel in one corner of the image is likely to have small effect on a different pixel far away from itself.

Re-writing exercise

Original:

Therefore, we can crop the image into smaller windows, as shown in Figure 5, and compute the inverse J matrix of the cropped window. Since the cropped window is much smaller than the input image, the inversion of J matrix is computationally cheaper.

Re-writing exercise

Original:

Since we
are inferring on blocks of image patches (i.e. ignoring pixels outside of the cropped window), the interpolated image will have blocky artifacts. Therefore, only part of xMAP is used to interpolate the image, as shown in Figure 5.

Re-writing exercise

The underlying assumption of this work is that the estimate of a given node will only depend on nodes within a patch: this is a locality assumption imposed at the patch-level. This assumption can be justified in case of skin images since a pixel in one corner of the image is likely to have small effect on a different pixel far away from itself. Therefore, we can crop the image into smaller windows, as shown in Figure 5, and compute the inverse J matrix of the cropped window. Since the cropped window is much smaller than the input image, the inversion of J matrix is computationally cheaper. Since we are inferring on blocks of image patches (i.e. ignoring pixels outside of the cropped window), the interpolated image will have blocky artifacts. Therefore, only part of xMAP is used to interpolate the image, as shown in Figure 5.

Before

We assume local influence--that nodes only depend on other nodes within a patch. This condition often holds for skin images, which have few long edges or structures. We crop the image into small windows, as shown in Fig. 5, and compute the inverse J matrix of each small window. This is much faster than computing the inverse J matrix for the input image. To avoid artifacts from the block processing, only the center region of xMAP is used in the final image, as shown in Fig. 5.

After

This editing benefits you twice: (1) you have 50% more space to tell your story, and (2) the text is easier for the reader to understand.

Figures and captions

It should be easy to read the paper in a big hurry and still learn the main points. Probably most of your readers will be skimming the paper.

The figures and captions can help tell the story.

So the figure captions should be self-contained and the caption should tell the reader what to notice about the figure.

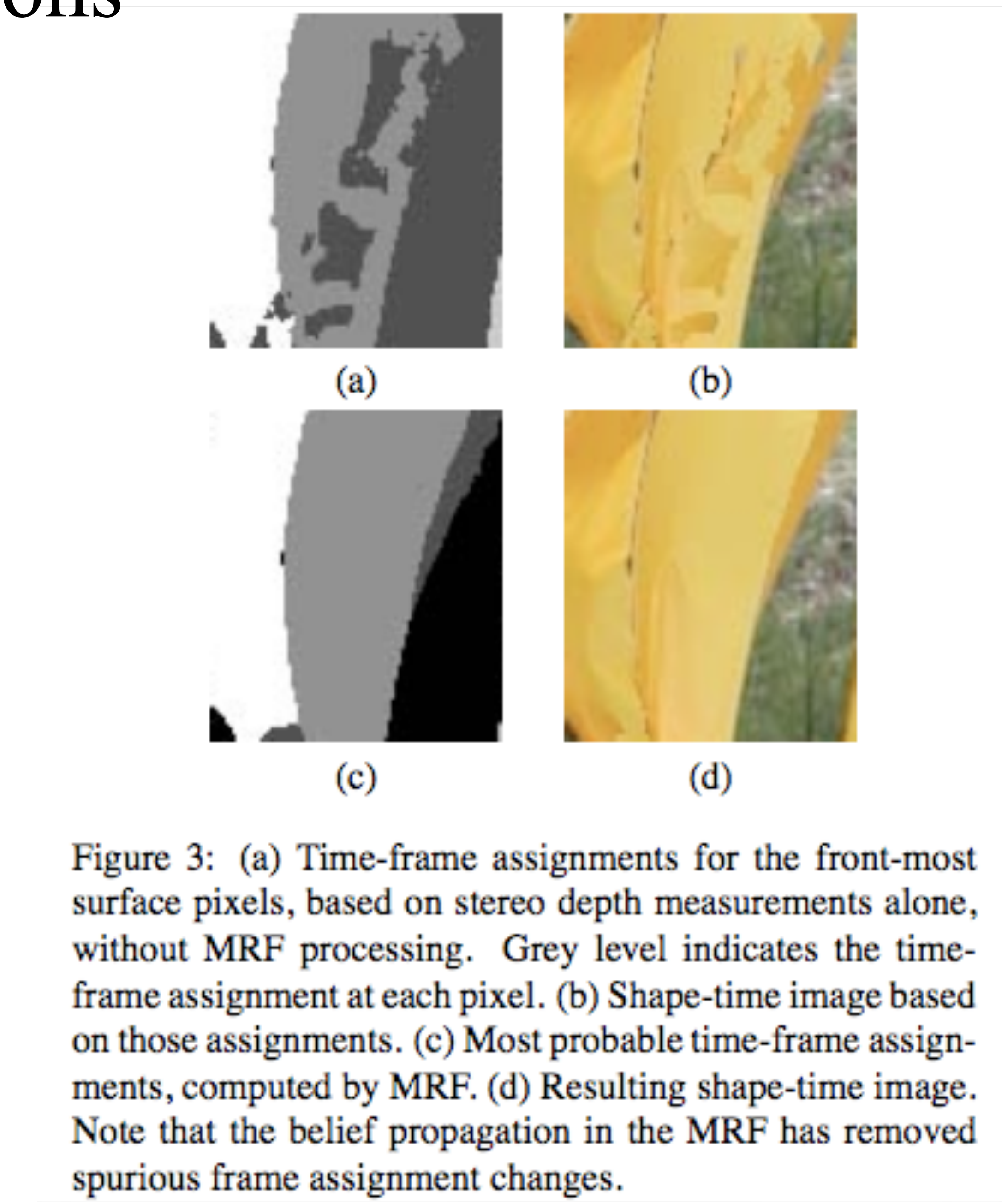
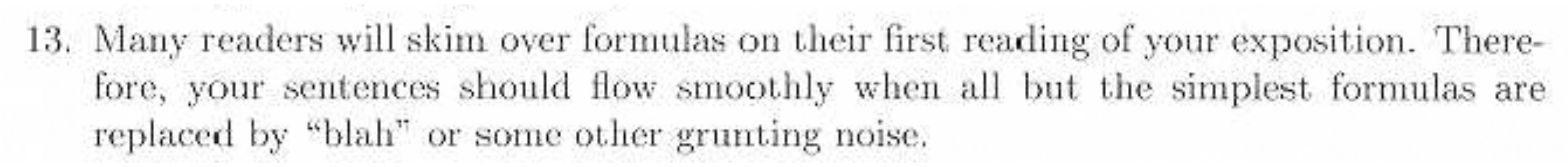


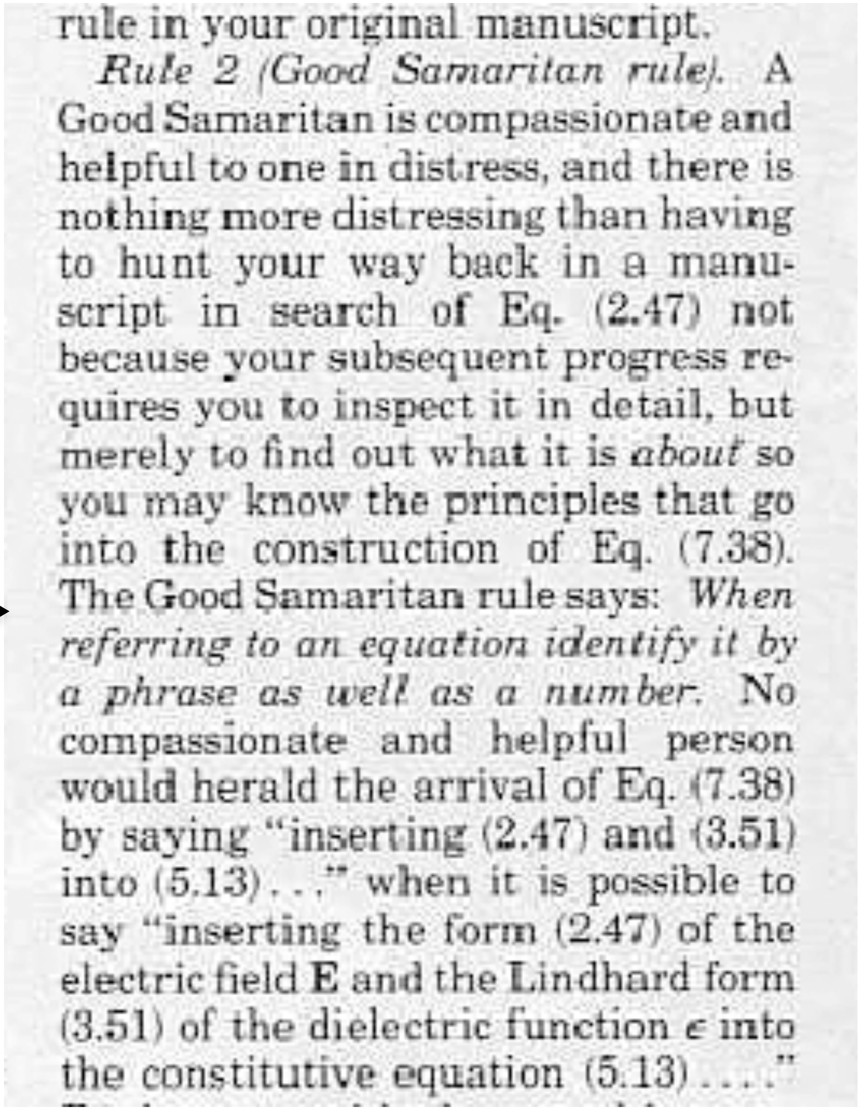
Figure 3: (a) Time-frame assignments for the front-most surface pixels, based on stereo depth measurements alone, without MRF processing. Grey level indicates the time-frame assignment at each pixel. (b) Shape-time image based on those assignments. (c) Most probable time-frame assignments, computed by MRF. (d) Resulting shape-time image. Note that the belief propagation in the MRF has removed spurious frame assignment changes.

Knuth on equations



13. Many readers will skim over formulas on their first reading of your exposition. Therefore, your sentences should flow smoothly when all but the simplest formulas are replaced by "blah" or some other grunting noise.

Mermin on equations



rule in your original manuscript.

Rule 2 (Good Samaritan rule). A Good Samaritan is compassionate and helpful to one in distress, and there is nothing more distressing than having to hunt your way back in a manuscript in search of Eq. (2.47) not because your subsequent progress requires you to inspect it in detail, but merely to find out what it is *about* so you may know the principles that go into the construction of Eq. (7.38). The Good Samaritan rule says: *When referring to an equation identify it by a phrase as well as a number.* No compassionate and helpful person would herald the arrival of Eq. (7.38) by saying "inserting (2.47) and (3.51) into (5.13) . . ." when it is possible to say "inserting the form (2.47) of the electric field **E** and the Lindhard form (3.51) of the dielectric function ϵ into the constitutive equation (5.13) . . . ."

Tone: be kind and gracious

- My initial comments.
- My advisor’s comments to me.

Image Quilting for Texture Synthesis and Transfer

Alexei A. Efros[1,2] William T. Freeman[2]

[1]University of California, Berkeley [2]Mitsubishi Electric Research Laboratories

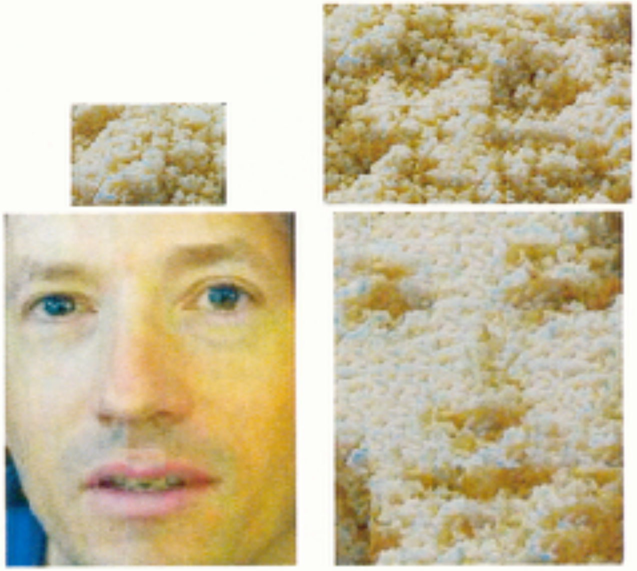
Abstract

We present a simple image-based method of generating novel visual appearance in which a new image is synthesized by stitching together small patches of existing images. We call this process *image quilting*. First, we use quilting as a fast and very simple texture synthesis algorithm which produces surprisingly good results for a wide range of textures. Second, we extend the algorithm to perform texture transfer – rendering an object with a texture taken from a different object. More generally, we demonstrate how an image can be re-rendered in the style of a different image. The method works directly on the images and does not require 3D information.

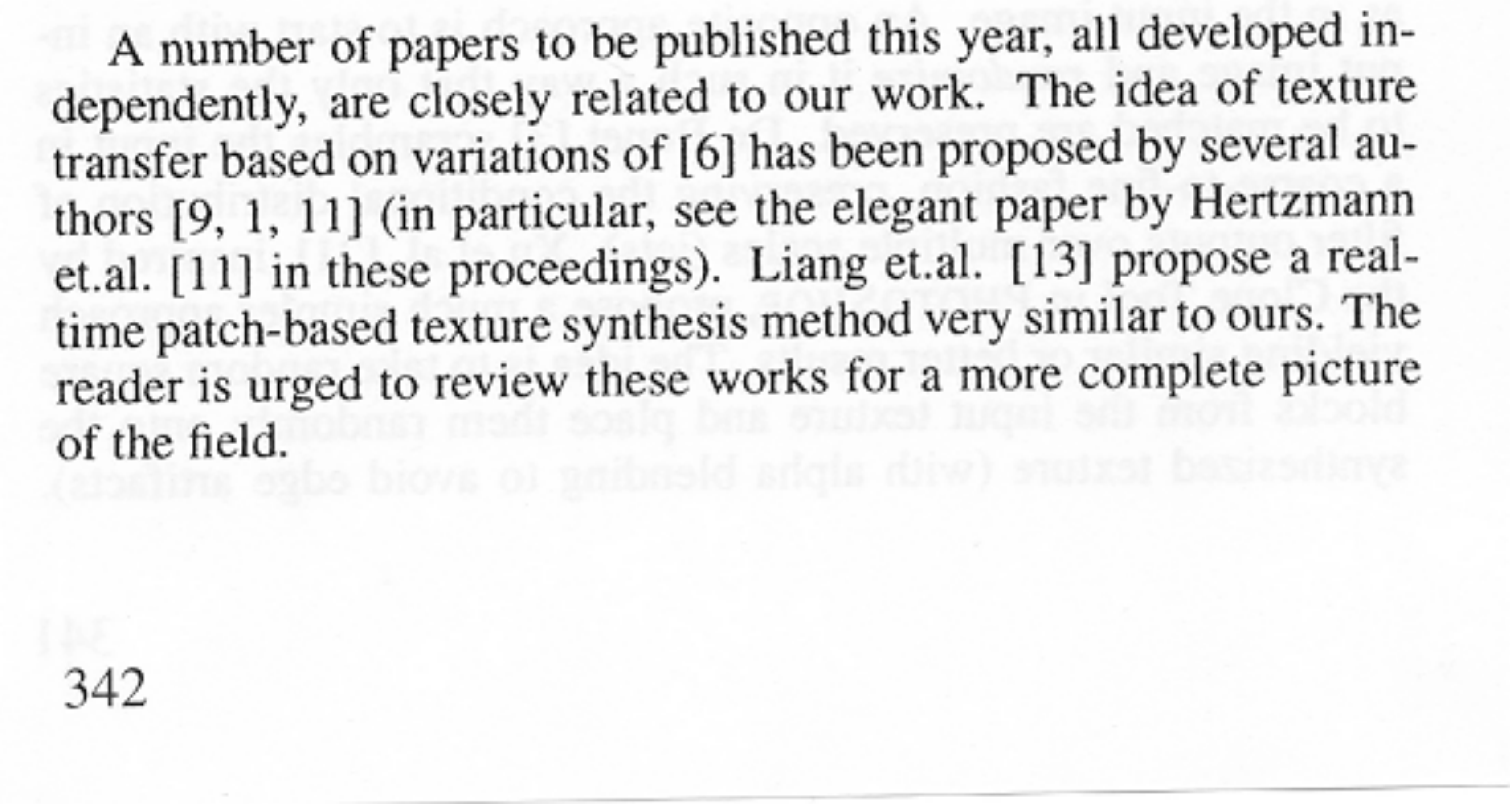
Keywords: Texture Synthesis, Texture Mapping, Image-based Rendering

1 Introduction

In the past decade computer graphics experienced a wave of activity in the area of image-based rendering as researchers explored the idea of capturing samples of the real world as images and using them to synthesize novel views rather than recreating the entire



Efros's comments within our texture synthesis paper about competing methods.



A number of papers to be published this year, all developed independently, are closely related to our work. The idea of texture transfer based on variations of [6] has been proposed by several authors [9, 1, 11] (in particular, see the elegant paper by Hertzmann et.al. [11] in these proceedings). Liang et.al. [13] propose a real-time patch-based texture synthesis method very similar to ours. The reader is urged to review these works for a more complete picture of the field.

Written from a position of security, not competition

Develop a reputation for being clear and reliable

(and for doing creative, good work…)

- There are perceived pressures to over-sell, hide drawbacks, and disparage others' work. Don't succumb. (That's in both your long and short-term interests).
- "because the author was [redacted], I knew I could trust the results." [a conference chair discussing some of the reasons behind a best paper prize selection].

Be honest, scrupulously honest

Convey the right impression of performance.

MAP estimation of deblurring. We didn't know why it didn't work, but we reported that it didn't work. Now we think we know why. Others have gone through contortions to show why they worked.

Author order

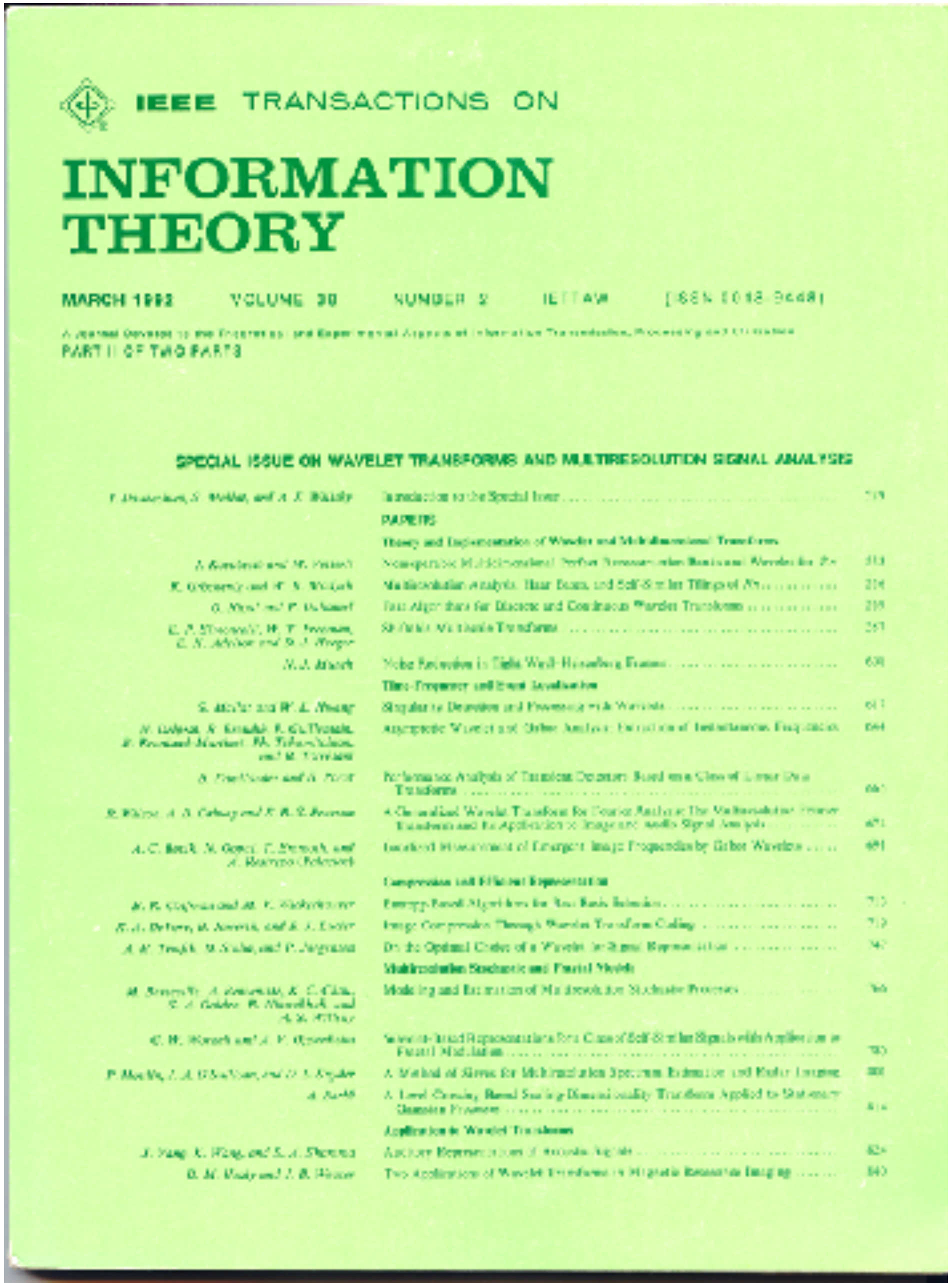
- Some communities use alphabetical order (physics, math).
- For biology, it's like bidding in bridge.
- Engineering seems to be: in descending order of contribution.
- Should the advisor be on the paper?
  - Did they frame the problem?
  - Do they know anything about the paper?
  - Do they need their name to appear on the papers for continued grant support?

My experiences with having names on papers

Author list

- My rule of thumb: All that matters is how good the paper is. If more authors make the paper better, add more authors. If someone feels they should be an author, and you trust them and you're on the fence, add them
- It's much better to be one of many authors on a great paper than to be one of just a few authors on a mediocre paper.
- The benefit of a paper to you is a very non-linear function of its quality:
  - A mediocre paper is worth nothing.
  - Only really good papers are worth anything.

Title?



IEEE TRANSACTIONS ON

INFORMATION THEORY

MARCH 1992 VOLUME 38 NUMBER 2 IETTAW (ISSN 0018-9448)

A Journal Devoted to the Theoretical and Experimental Aspects of Information Transmission, Processing and Utilization

PART II OF TWO PARTS

SPECIAL ISSUE ON WAVELET TRANSFORMS AND MULTIRESOLUTION SIGNAL ANALYSIS

PAPERS

Theory and Implementation of Wavelet and Multidimensional Transforms

Time-Frequency and Space Localization

Compression and Efficient Representation

Multiresolution Stochastic and Fractal Models

Applications of Wavelet Transforms

Our title

- Was:
  - Shiftable Multiscale Transforms.
- Should have been:
  - What’s Wrong with Wavelets?

How papers are evaluated

After the papers come in:

- Program chairs assign each paper to an area chair.
- Area chairs assign each of their papers to 3 (or for SIGGRAPH, 5) reviewers.
- Reviewers read and review 5 – 15 papers.
- Authors respond to reviews.
- Area chairs read reviews and author/reviewer dialog and look at paper and decide whether to reject or accept as poster or oral talk. The area chair may have 30 or so papers to handle.

Strategy tips

From an area chair’s point of view, the types of papers in your pile

- About 1/3 are obvious rejects
- In the whole set, maybe 1 is a really nice paper--well-written, great results, good idea. That will be an oral presentation.
- The rest are borderline, and these fall into two camps...

From an area chair’s point of view, the two types of borderline papers...

Quick and easy reasons to reject a paper

With the task of rejecting at least 75% of the submissions, area chairs are groping for reasons to reject a paper. Here’s a summary of reasons that are commonly used:

- Do the authors not deliver what they promise?
- Are important references missing (and therefore one suspects the authors not up on the state-of-the-art for this problem)?
- Are the results too incremental (too similar to previous work)
- Are the results believable (too different than previous work)?
- Is the paper poorly written?
- Are there mistakes or incorrect statements?

Sources on writing technical papers

- How to Get Your SIGGRAPH Paper Rejected, Jim Kajiya, SIGGRAPH 1993 Papers Chair, http://www.siggraph.org/publications/instructions/rejected.html
- Ted Adelson's Informal guidelines for writing a paper, 1991. http://www.ai.mit.edu/courses/6.899/papers/ted.htm
- Notes on technical writing, Don Knuth, 1989.

  http://www.ai.mit.edu/courses/6.899/papers/knuthAll.pdf

- What's wrong with these equations, David Mermin, Physics Today, Oct., 1989. http://www.ai.mit.edu/courses/6.899/papers/mermin.pdf
- Notes on writing by Fredo Durand, people.csail.mit.edu/fredo/PUBLI/writing.pdf and Aaron Hertzmann, http://www.dgp.toronto.edu/~hertzman/advice/writing-technical-papers.pdf
- Three sins of authors in computer science and math, Jonathan Shewchuck, http://www.cs.cmu.edu/~jrs/sins.html
- Ten Simple Rules for Mathematical Writing, Dimitri P. Bertsekas http://www.mit.edu:8001/people/dimitrib/Ten_Rules.html

Outline

- writing technical papers
- giving technical talks

How to give talks

- Giving good talks is important for a researcher.
- You might think, “the work itself is what really counts. Giving the talk is secondary”.
- But the ability to give a good talk is like having a big serve in tennis—by itself, it doesn’t win the game for you. But it sure helps. And the very best tennis players all have great serves.
- Researchers as little corporations.

http://imagesource.allposters.com/images/pic/SSPOD/superstock_294-341c_b~Tennis-Serve-Posters.jpg

Sources on giving talks

Patrick Winston's annual IAP talk on how to give talks.

Books on speaking.

Suggestions from your advisor or helpful audience members.

Analyzing good talks that others give.

High order bit: prepare

http://tbn0.google.com/images?q=tbn:pfwAIhkEy8t0EM:http://
www.itcstirlingspeaking.org.uk/images/woman%2520speaker.jpg

- Practice by yourself.
- Give practice versions to your friends.
- Think through your talk.
- You can write out verbatim what you want to say in the difficult parts.
- Ahead of time, visit where you’ll be giving the talk and identify any issues that may come up.
- Preparation is a great cure for nervousness.

A tip to not be nervous that I found useful

- Get over it. They’re not there to see you, they’re there to hear the information. Just convey the information to them.

The different kinds of talks you’ll have to give as a researcher

- 2-5 minute talks
- 10 -20 minute conference presentations
- 30-60 minute colloquia

Very short talks

- Rehearse it.
- Cut things out that aren’t essential. You can refer to them at a high level.

- You might focus on answering just a few questions, eg: what is the problem? Why is it interesting? Why is it hard?
- Typically these talks are just little advertisements for a poster or for some other (longer) talk. So you just need to show people that the problem is interesting and that you’re fun to talk with.
- These talks can convey important info--note popularity of SIGGRAPH fast forward session.

Recommendation

- For your five-minute talks, write down:
  - what problem did you address?
  - why is it interesting?
  - why is it hard?
  - what was the key to your approach?
  - how well did it work?

The different kinds of talks you'll have to give as a researcher

- 2-5 minute talks
- 10-20 minute conference presentations
- 30-60 minute colloquia

David Jacob’s bad news

The more you work on a talk, the better it gets: if you work on it for 3 hours, the talk you give will be better than if you had only worked on it for 2 hours. If you work on it for 5 hours, it will be better still. 7 hours, better yet…

(told to me by David on a beach in Greece, a few hours before my oral presentation at ICCV. That motivated me to leave the beach and go back to my room to work more on my talk, which paid off).

Figure out how one part follows from another

Ahead of time, think through how each part motivates the next, and point that out during the talk. If one part doesn’t motivate the next, consider re-ordering the talk until it has that feel.

Your audience

- Your image of your audience:
  - Paying attention, listening to every word

- Your audience in reality:
  - Tired, hungry, not wanting to sit through yet another talk at the conference…

Layering the talk. When we read a paper, headings and sections help us follow the paper. You should provide the verbal equivalents of headings to the listener.

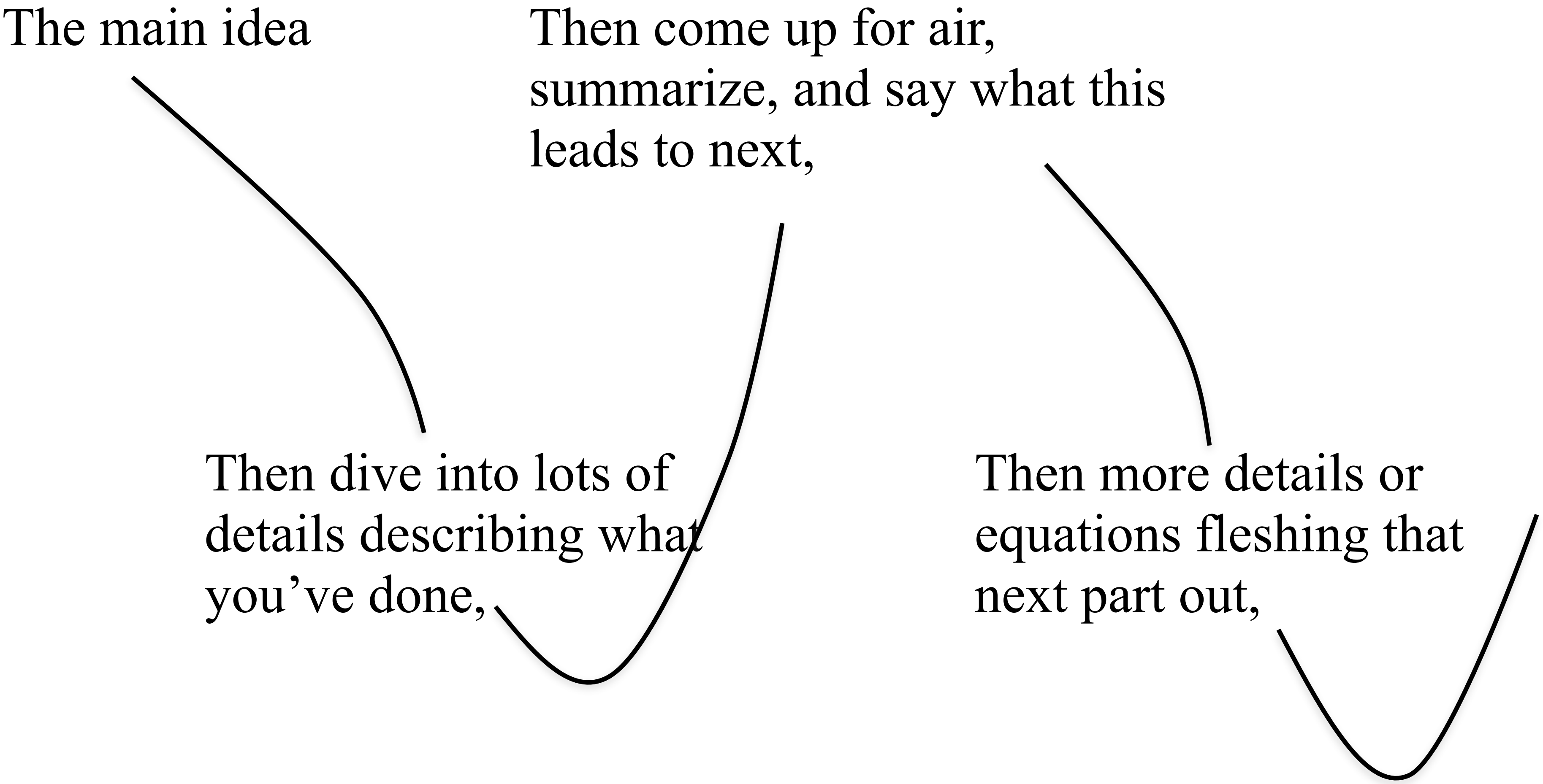
http://tbn0.google.com/images?q=tbn:4oWYOjaSp4vopM:http://bakery.grillsforallseasons.com/photos/wedding_cake3.jpg

The probability of an observation has three terms to it. Blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah

So that gives us the objective function we want to optimize. Now, how do we find the optimal value? There are two approaches you can take. blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah

So now, with these tools in hand, we can apply this methods to real images. blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah

You tell the story at several different levels of detail



The main idea

Then dive into lots of details describing what you've done,

Then come up for air, summarize, and say what this leads to next,

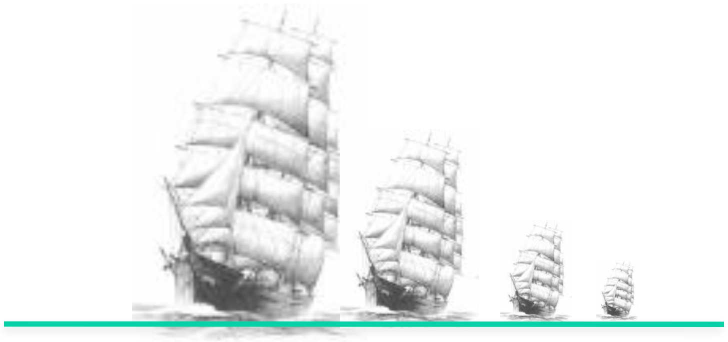
Then more details or equations fleshing that next part out,

Ways to engage the audience

- So you’ve been talking on and on. You want to break things up and keep the audience engaged. Can you think of a way to bring the audience into the talk?
- Demos can also help.
- Or add audience participation components to the talk. For human or computer vision talks, you can often present to the audience what the task is that the human or computer has to solve.
- The audience loves to figures things out, to solve puzzles, to make guesses. Feed those desires.
- The response-meter.

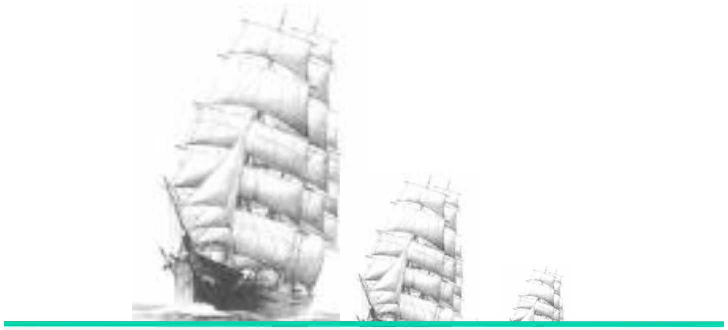
Ted Adelson

- “people like to see a good fight”
- The flat earth theory predicts that ships will appear on the horizon as small versions of the complete ship. Under that theory, you’d expect approaching ships to look like this:



Present a fight

Whereas the round earth theory predicts that the top of the sails will appear first, then gradually the rest of the ship below it.



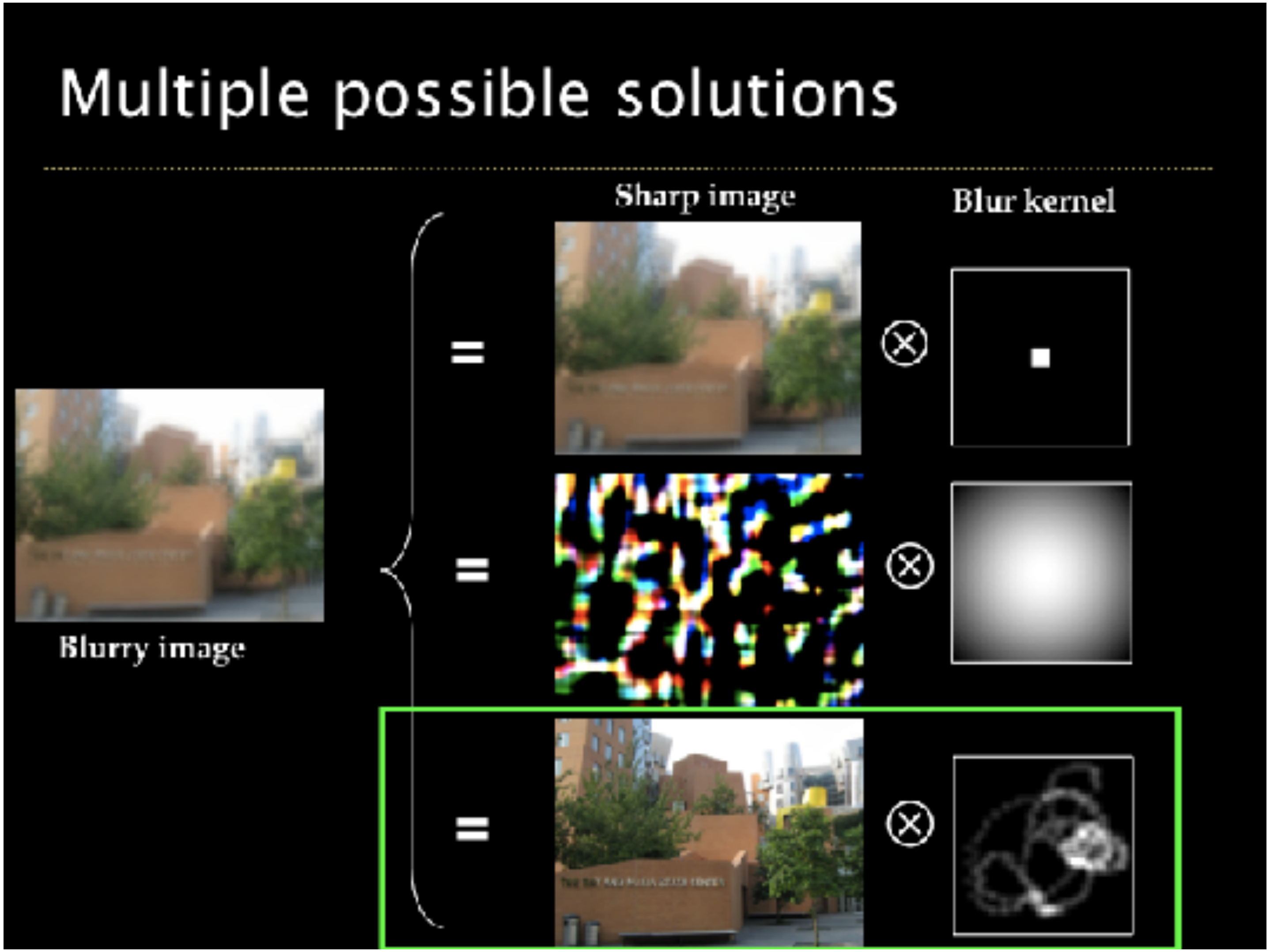
http://www.flickr.com/photos/mnsomero/
2738807250/

Add dynamics to the talk

http://images.google.com/imgres?imgurl=http://operachic.typepad.com/photos/uncategorized/ulster01.jpg&imgrefurl=http://operachic.typepad.com/opera_chic/2007/01/
index.html&h=321&w=490&sz=57&hl=en&start=2&tbnid=6seUYhUX6x2DrM:&tbnh=85&tbnw=130&prev=/
images%3Fq%3Dconductor%2Borchestra%2Bquiet%26gbv%3D2%26svnum%3D10%26hl%3Den%26safe%3Doff%26sa%3DG

http://www.nch.ie/dynamic/img/Mariss%20Jansons%20%20new.jpg

- A talk is a story. As in a story, there can be different levels of excitement or tension in different parts of the talk. This makes it easier for the audience to pay attention to what you’re saying. Perhaps move to another location.
- I like to find some part of the work that really grabs me, that I’m really excited about, and let that show through. (The audience loves to see you be excited. Not all the time, but when appropriate). “I love this problem; it’s beautifully underdetermined. There are lots of different ways we can explain the observed blurry image. It could be that that’s what was there in the world, and we took a sharp picture of it….”



Multiple possible solutions

Sharp image

Blur kernel

Blurry image

What I think the audience wants

To have everything follow and make sense

To learn something

To connect with the speaker, to share their excitement.

They want to watch you love something!

Alan Alda’s comments (see http://mcgovern.mit.edu/video-gallery, starting at 18 minutes in (but earlier is good, too).)

Present to the mean.

Let the audience see your personality

- They want to see you enjoy yourself.
- They want to see what you love about the work.
- People really respond to the human parts of a talk. Those parts help the audience with their difficult task of listening to an hour-long talk on a technical subject. What was easy, what was fun, what was hard about the work?
- Don’t be afraid to be yourself and to be quirky.

http://is3.okcupid.com/users/112/250/11225140098321842389/mt1112532356.jpg

How to end a talk

- People often say “are there any questions?” but then people don’t know whether to applaud or to raise their hand.
- If you say “thank you”, then everyone knows that they’re supposed to applaud now. After that is over, then you can ask for questions.