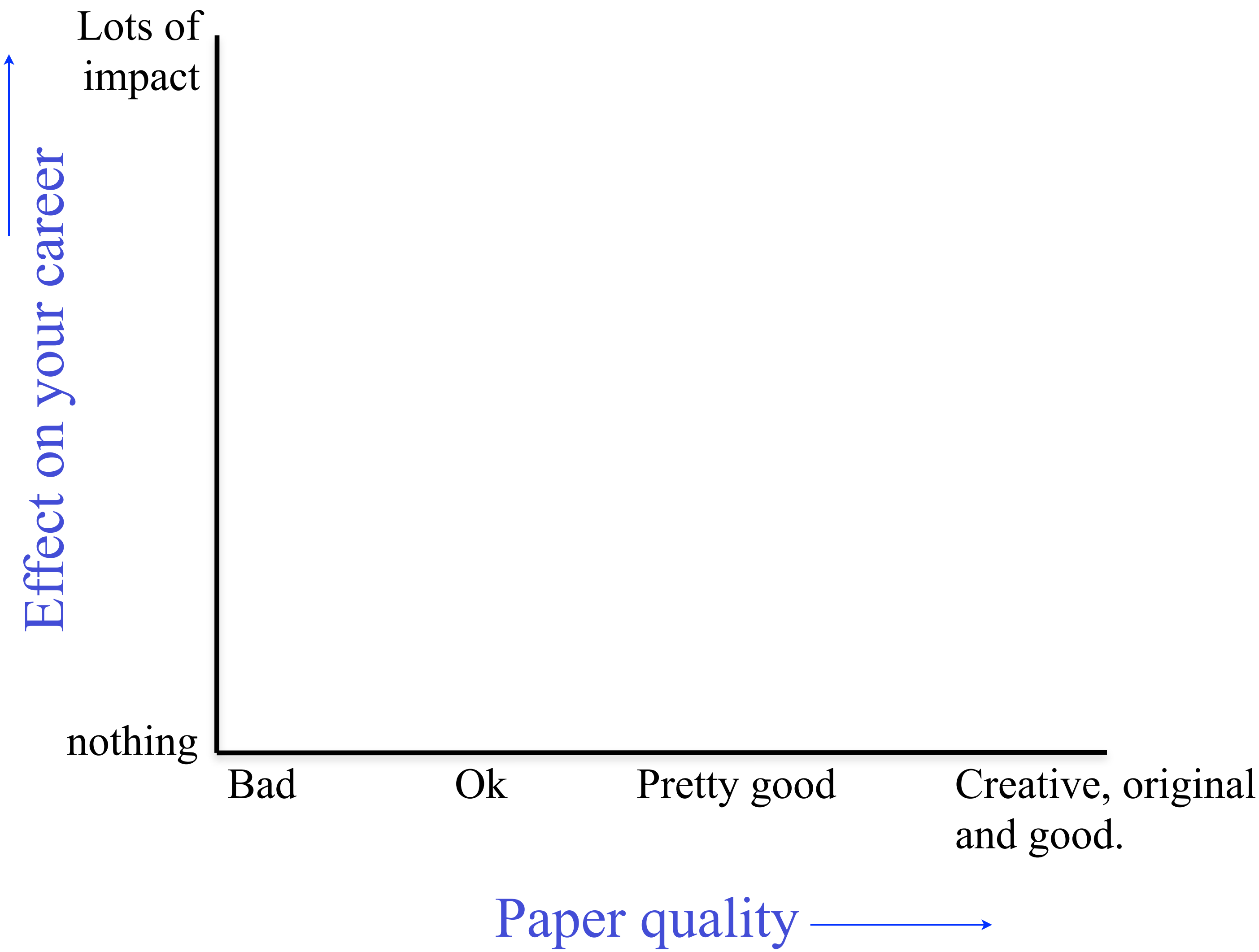
How to write a good CVPR submission

Bill Freeman

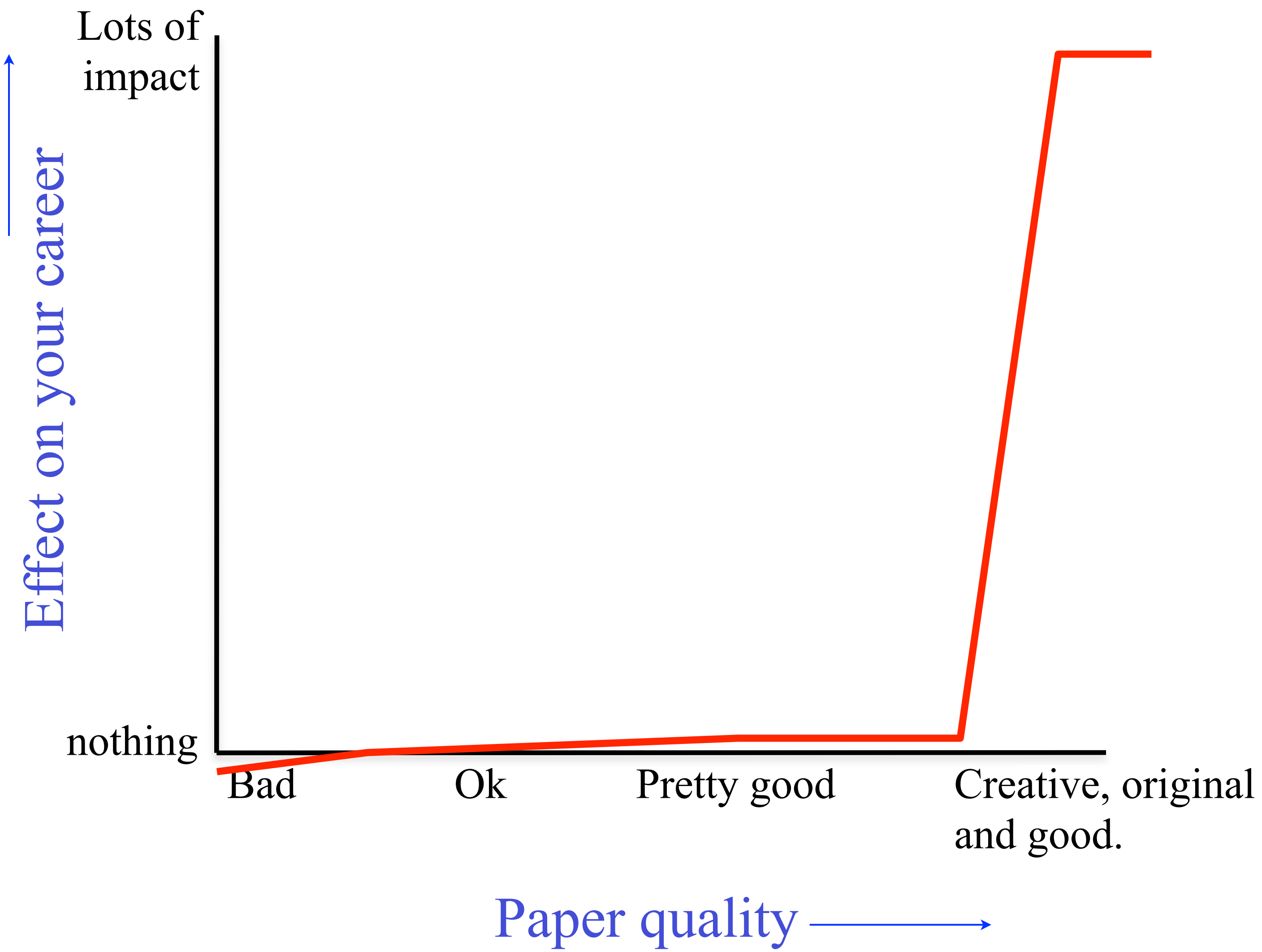
MIT CSAIL

Nov. 6, 2014

A paper's impact on your career



A paper's impact on your career



My experiences

- Review conference papers.
- Was an IEEE PAMI Associate Editor.
- Area chair for ICCV, CVPR, NIPS, SIGGRAPH several times each.
- Program co-chair for ICCV 2005 and CVPR 2013.

Where publish

- Journal
  - Long turn-around time
  - But “archival”
  - Counts more in tenure decisions, although university deans are being trained that many computer science conference venues are more competitive than journals.
  - Have a dialog with reviewers and editor.
- Conference
  - Immediate feedback
  - Publication within 6 or 7 months.
  - One-shot reviewing.   Sometimes the reviewing is sloppier.

Conferences in computer vision and related areas

- CVPR/ICCV/ECCV (Computer Vision and Pattern Recognition/Intl. Conf. on Computer Vision/European Conf. on Computer Vision)
  - ~2000 submissions, ~22% acceptance
  - Reviewing improving
  - The main venues for computer vision and machine learning applied to computer vision
-
- SIGGRAPH (ACM Special Interest Group on Graphics)
  - 550 submissions, 20% acceptance
  - Good, careful reviewing. Needs spectacular images.
  - Some vision-and-graphics and learning-and-graphics.
  - Also a journal, by the way (special issue of Trans. On Graphics)
- NIPS (Neural Information Processing Systems)
  - 1500 submissions, ~25% acceptance
  - Reasonable reviewing. Needs some math component.
  - Vision is a sidelight to the main machine learning show.

- 2nd tier: BVMC, German Signal Processing Society, Asian Conference on Computer Vision, and workshops associated with CVPR, ICCV, and ECCV.

How conferences are organized

- Program chairs for the conference are selected
  - SIGGRAPH, NIPS: by some overseeing organizing committee
  - CVPR, ICCV: by conference attendee vote at a previous conference. Selection of city and program chairs are coupled.
- The area chairs are selected by the program chairs.
- Submission deadlines strict.

How papers are evaluated

After the papers come in:

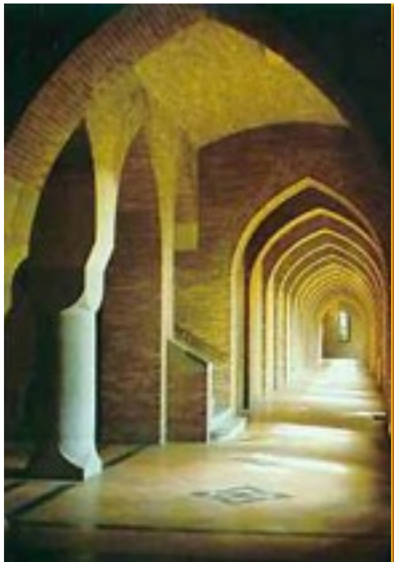
- Program chairs assign each paper to an area chair.
- Area chairs assign each of their papers to 3 (or for SIGGRAPH, 5) reviewers.
- Reviewers read and review 5 – 15 papers.
- Authors respond to reviews.
- Area chairs read reviews and author/reviewer dialog and look at paper and decide whether to reject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

The conference paper selection meeting

- Area chairs meet to decide which papers to accept. The reviewers' scores give an initial ranking; the area chairs then push papers up or down. NIPS: not much discussion; the reviewers' scores carry a lot of weight. SIGGRAPH: lots of discussion. Highly ranked papers can get killed, low-ranked papers can get in. CVPR, ICCV: intermediate level of discussion.

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/

Kajiya on conference reviewing

"The reviewing process for SIGGRAPH is far from perfect, although most everyone is giving it their best effort.

The very nature of the process is such that many reviewers will not be able to spend nearly enough time weighing the nuances of your paper. This is something for which you must compensate in order to be successful."

Kajiya on SIGGRAPH reviewing (applies to vision conferences, too)

“The emphasis on both speed and quality makes the reviewing process for SIGGRAPH very different from of a journal or another conference.

The speed and quality emphasis also puts severe strains on the reviewing process.

In SIGGRAPH, if the reviewers misunderstand your paper, or if some flaw in your paper is found, you're dead.”

Kajiya description of what reviewers look for.

The most dangerous mistake you can make when writing your paper is assuming that the reviewer will understand the point of your paper. The complaint is often heard that the reviewer did not understand what an author was trying to say

Make it easy to see the main point

Your paper will get rejected unless you make it very clear, up front, what you think your paper has contributed. If you don't explicitly state the problem you're solving, the context of your problem and solution, and how your paper differs (and improves upon) previous work, you're trusting that the reviewers will figure it out.

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.

Kajiya

Paper organization

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?

Ted Adelson on paper organization.

(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

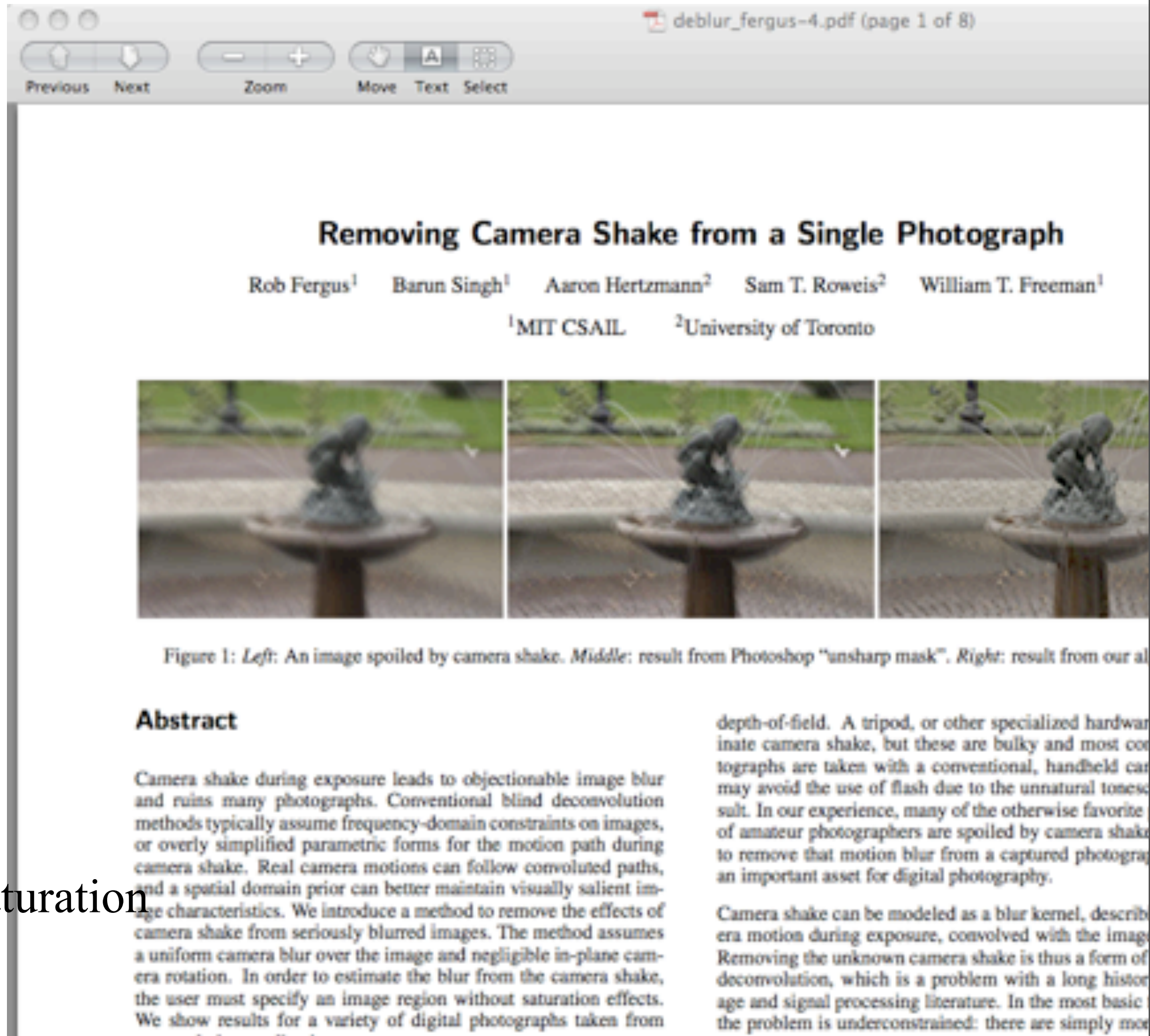
4 Algorithm

- 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

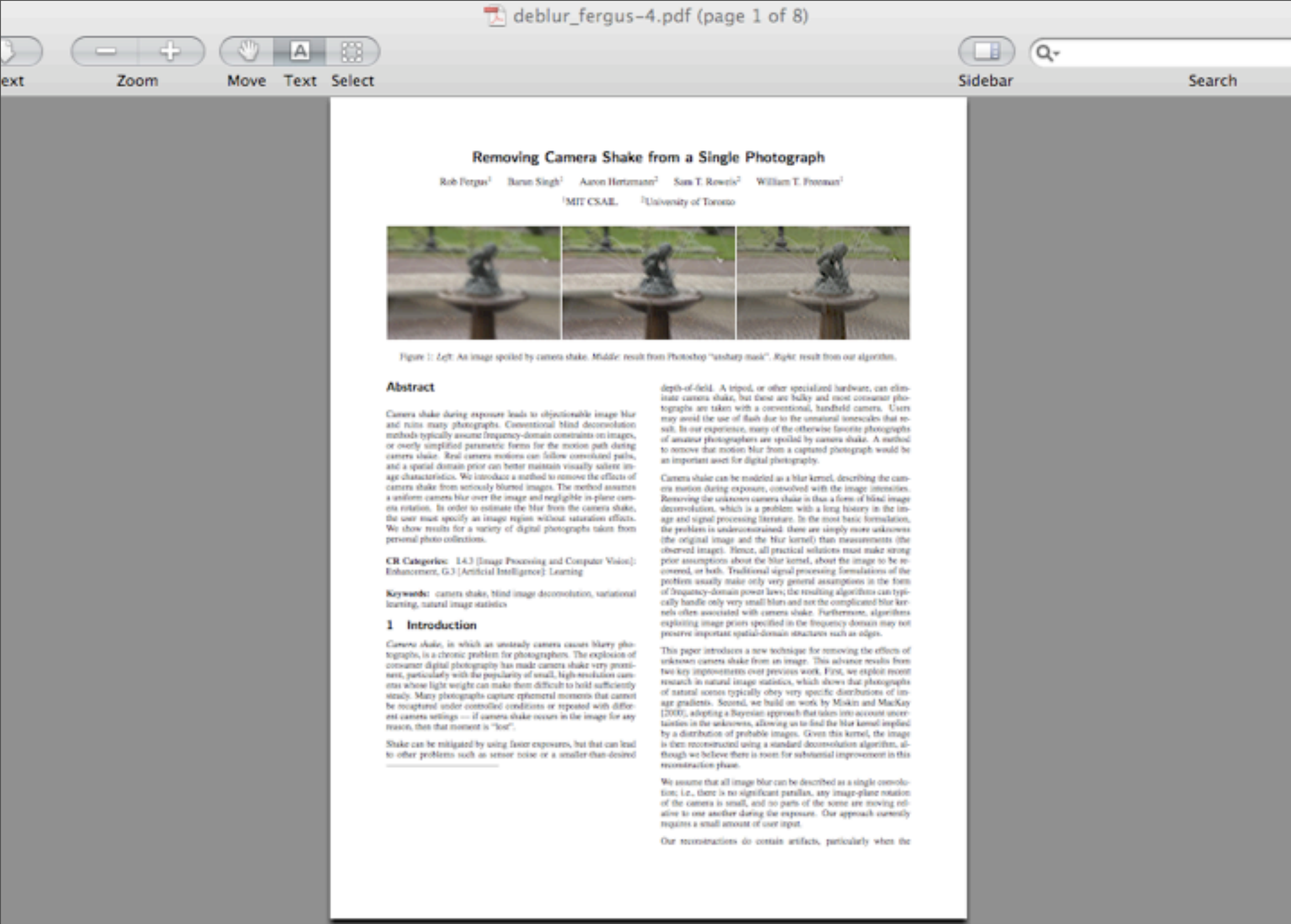
Rob Fergus[1] Barun Singh[1] Aaron Hertzmann[2] Sam T. Roweis[2] William T. Freeman[1]

[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



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.

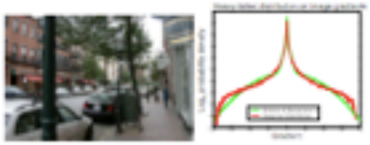
Our reconstructions do contain artifacts, particularly when the

Next Zoom Move Text Select Sidebar Search

above assumptions are violated; however, they may be acceptable to consumers in 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

3 Image model



4 Algorithm

4.1 Estimating the blur kernel

4.1.1 Multi-scale approach

Zoom Move Text Select Sidebar

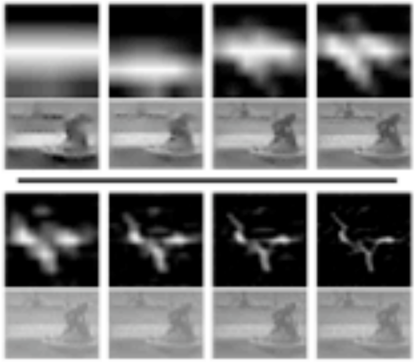


Figure 3: The multi-scale inference scheme operating on the fountain image in Figure 1. *1st & 3rd rows:* The estimated blur kernel at each scale level. *2nd & 4th rows:* Estimated image patch at each scale. The intensity image was reconstructed from the gradients used in the inference using Poisson 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 $\nabla \mathbf{L}$, 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 initalized 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 $\mathbf{K}$, 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 $\mathbf{K}$ 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 $\mathbf{L}$ typically is large, speed considerations make simple methods attractive. Consequently, we reconstruct the latent color image $\mathbf{L}$ 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 $\mathbf{L}$, given $\mathbf{K}$, 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 $\mathbf{K}$ 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 $\mathbf{B}$ (using Matlab's `histeq` function), resulting in $\mathbf{L}$. 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

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

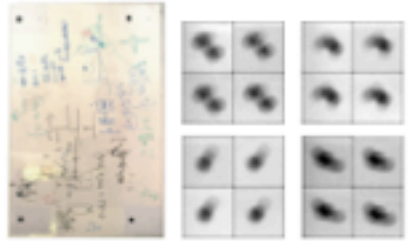


Figure 4: *Left:* The whiteboard test scene with dots in each corner. *Right:* Dots from the corners of images taken by different people. Within each image, the dot trajectories are very similar suggesting that image blur is well modeled as a spatially invariant convolution.

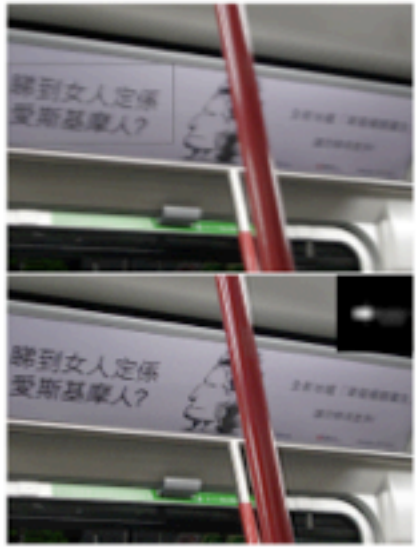


Figure 5: *Top:* A scene with a small blur. The patch selected by the user is indicated by the gray rectangle. *Bottom:* Output of our algorithm and the inferred blur kernel. Note the crisp text.

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.

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 no
by our algorithm.

As demonstrated in Figure 8, the true blur kernel is oc
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 t
five images.

We also compared our algorithm against existing blind
lution algorithms, running Matlab's `deconvblind` routi
provides implementations of the methods of Biggs and
[1997] and Jansson [1997]. Based on the iterative Richard
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
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 arti
saturated regions, the unsaturated regions can still be ext

Thursday, November 6, 14

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.

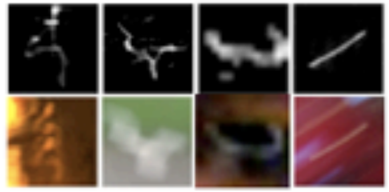


Figure 10: *Top row*: Inferred blur kernels from four real images (the cafe, fountain and family scenes plus another image not shown). *Bottom row*: Patches extracted from these scenes where the true kernel has been revealed. In the cafe image, two lights give a dual image of the kernel. In the fountain 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.

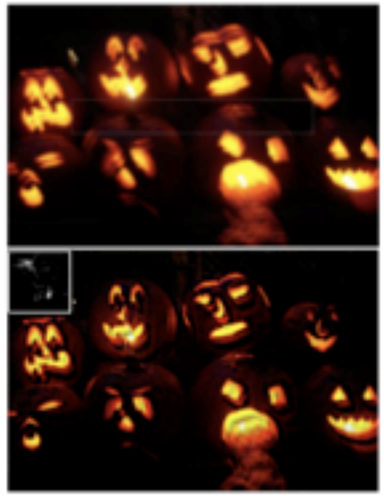


Figure 12: *Top*: A blurred scene with significant saturation. The long thin region selected by the user has limited saturation. *Bottom*: output of our algorithm. Note the double exposure type blur kernel.

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

Zoom Move Text Select Sidebar Search

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 Images 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 *ICCV*, 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), 699–708.

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 ϕ; 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
  $\nabla\mathbf{P}^s$ = `imresize`($\nabla\mathbf{P}, (\frac{1}{\sqrt{2}})^{S-s}$,`'bilinear'`). % Rescale gradients
  if (s==1) **then** % Initial kernel and gradients
    $\mathbf{K}^s = [0,0,0;1,1,1;0,0,0]/3$. If ($o$ == 1), $\mathbf{K}^s = (\mathbf{K}^s)^T$.
    $[\mathbf{K}^s, \nabla\mathbf{L}_p^s]$ = `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
    $\nabla\mathbf{L}_p^s$ = `imresize`($\nabla\mathbf{L}_p^{s-1}, \sqrt{2}$,`'bilinear'`).
    $\mathbf{K}^s$ = `imresize`($\mathbf{K}^{s-1}, \sqrt{2}$,`'bilinear'`).
  end if
  $[\mathbf{K}^s, \nabla\mathbf{L}_p^s]$ = `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 θ_K; latent gradient prior θ_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
  ψ^* = `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 $\nabla\mathbf{L}_p^{new}$.

ψ^* =**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
  [illegible]
  [illegible]
  [illegible] % Sufficient statistics for $q(\mathbf{K})$
  [illegible]
  [illegible] % Sufficient statistics for $q(\nabla\mathbf{L}_p)$
  [illegible]
  $a = 10^{-3} + \frac{1}{2}\sum_{ij}(\nabla\mathbf{P} - (\mathbf{K} \otimes \nabla\mathbf{L}_p))_{ij}^2$; $b = 10^{-3} + IJ/2$ % 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
  $E[l_{ij}] = \zeta'_{ij}/\zeta_{ij}$; $E[l_{ij}^2] = (\zeta'_{ij}/\zeta_{ij})^2 + 1/\zeta_{ij}$. % 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: ψ^*

Write a dynamite 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

Kajiya description of what reviewers look for.

Again, stating the problem and its context is important. But what you want to do here is to state the "implications" of your solution. Sure it's obvious....to you. But you run the risk of misunderstanding and rejection if you don't spell it out explicitly in your introduction.

Kajiya: write a dynamite introduction

How can you protect yourself against these mistakes? 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. In your introduction you can address most of the points we talked about in the last section. If you do it clearly and succinctly, you set the proper context for understanding the rest of your paper. Only then should you go about describing what you've done.

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

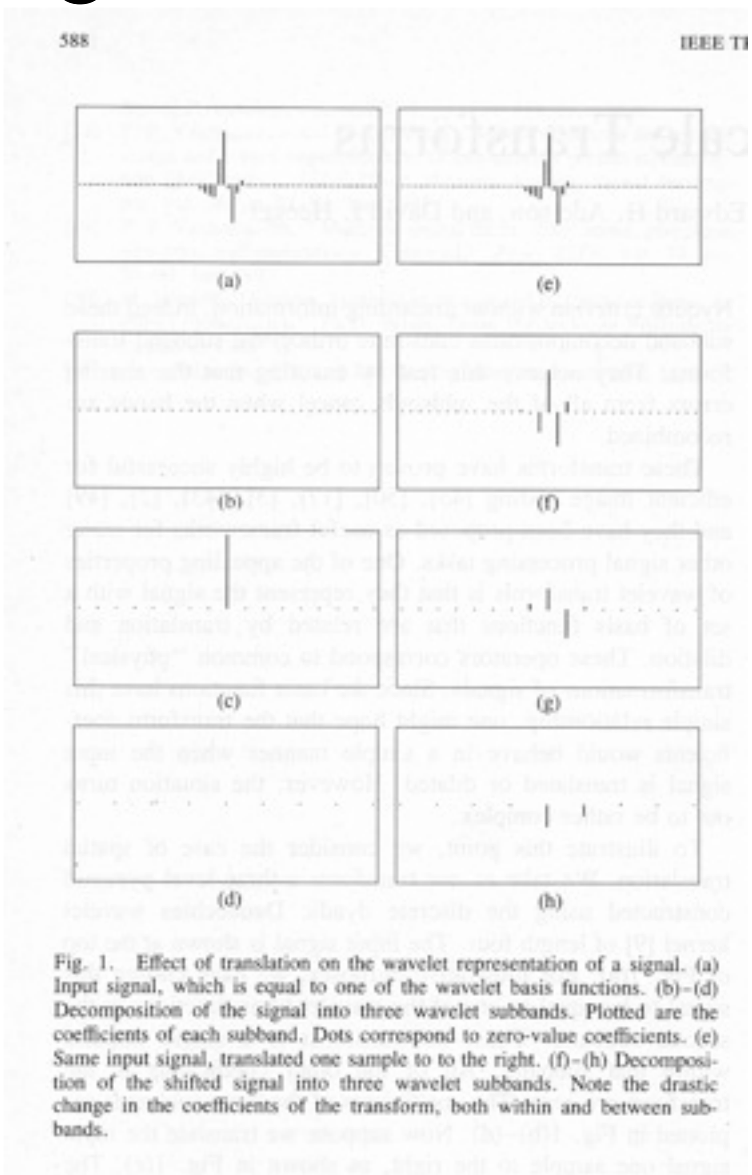
5 Experiments

- 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

(a) (e)

(b) (f)

(c) (g)

(d) (h)

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

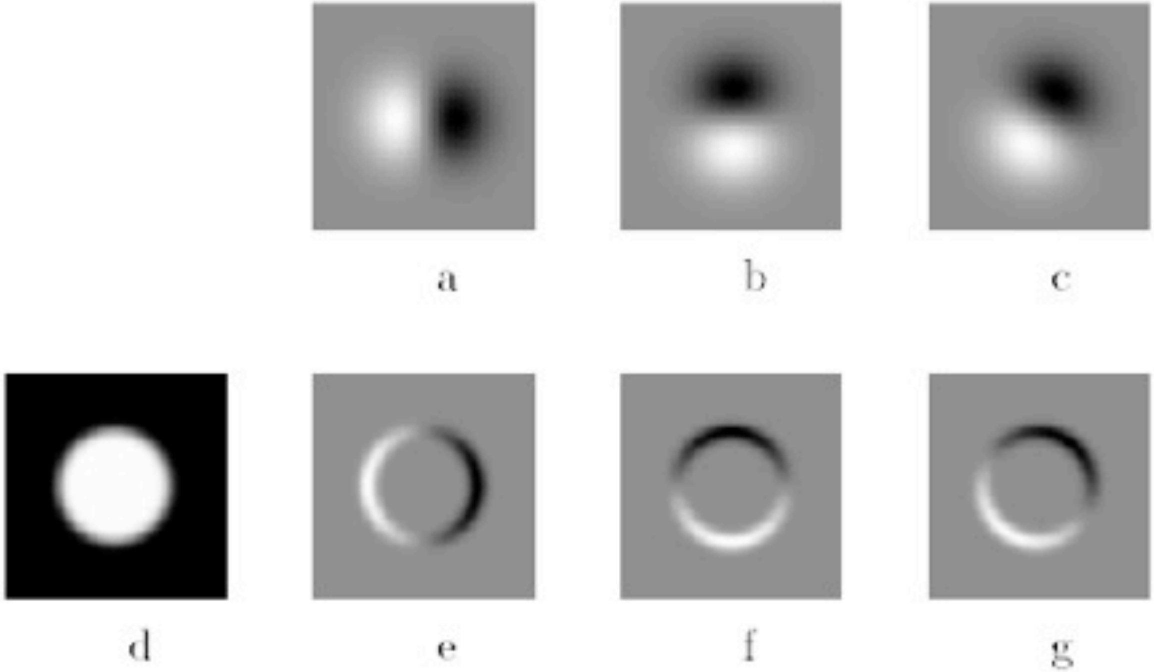


Fig. 1. Example of steerable filters: (a) $G_1^{0^\circ}$ first derivative with respect to x (horizontal) of a Gaussian; (b) $G_1^{90^\circ}$, which is $G_1^{0^\circ}$, rotated by 90°. From a linear combination of these two filters, one can create G_1^{θ}, which is an arbitrary rotation of the first 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) $G_1^{0^\circ}$ (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)).

Comments on writing

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

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.

e

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.

Revised:

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.

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:

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.

Revised:

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.

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

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.

Revised:

To avoid artifacts from the block processing, only the center region of xMAP is used in the final image, as shown in Fig. 5.

Kajiya

Is the paper well written?

Your ideas may be great, the problem of burning interest to a lot of people, but your paper might be so poorly written that no one could figure out what you were saying. If English isn't your native tongue, you should be especially sensitive to this issue. Many otherwise good papers have floundered on an atrocious text. If you have a planned organization for your discussion and you not only stick to it, but tell your readers over and over where you are in that organization, you'll have a well written paper. Really, you don't have to have a literary masterpiece with sparkling prose.

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?

    When describing the work of other people it is sometimes safe to provide motivation by simply stating that it is "interesting" or "remarkable"; but it is best to let the results speak for themselves or to give *reasons* why the things seem interesting or remarkable.

    When describing your own work, be humble and don't use superlatives of praise, either explicitly or implicitly, even if you are enthusiastic.

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 transformed 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

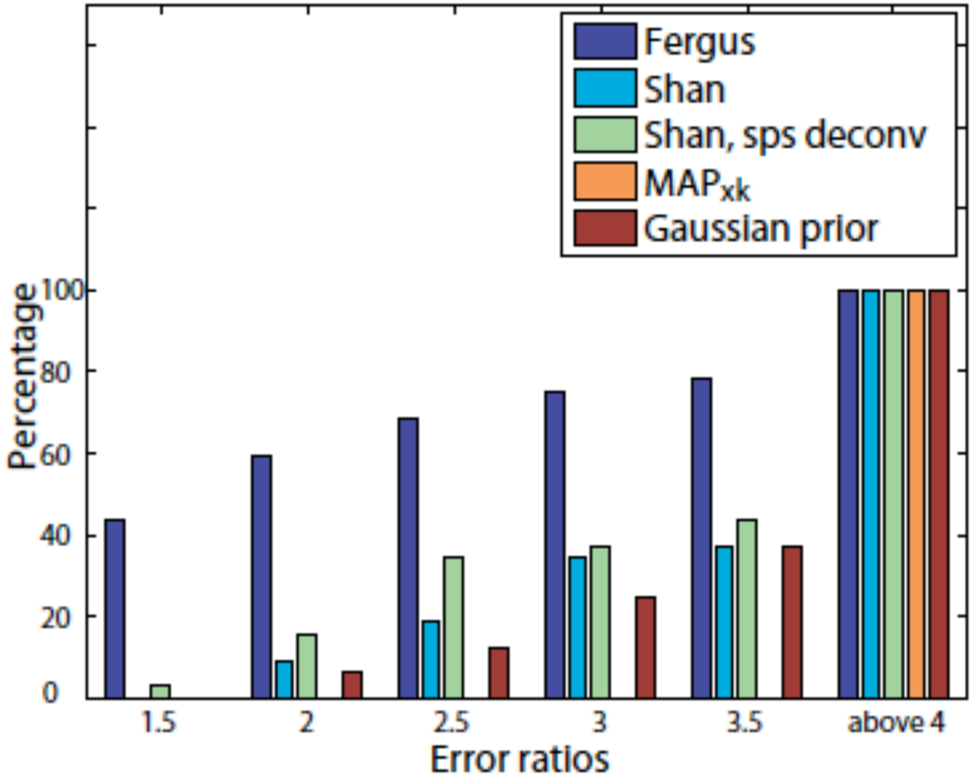


Figure 9. Evaluation results: Cumulative histogram of the deconvolution error ratio across test examples.

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 on equations

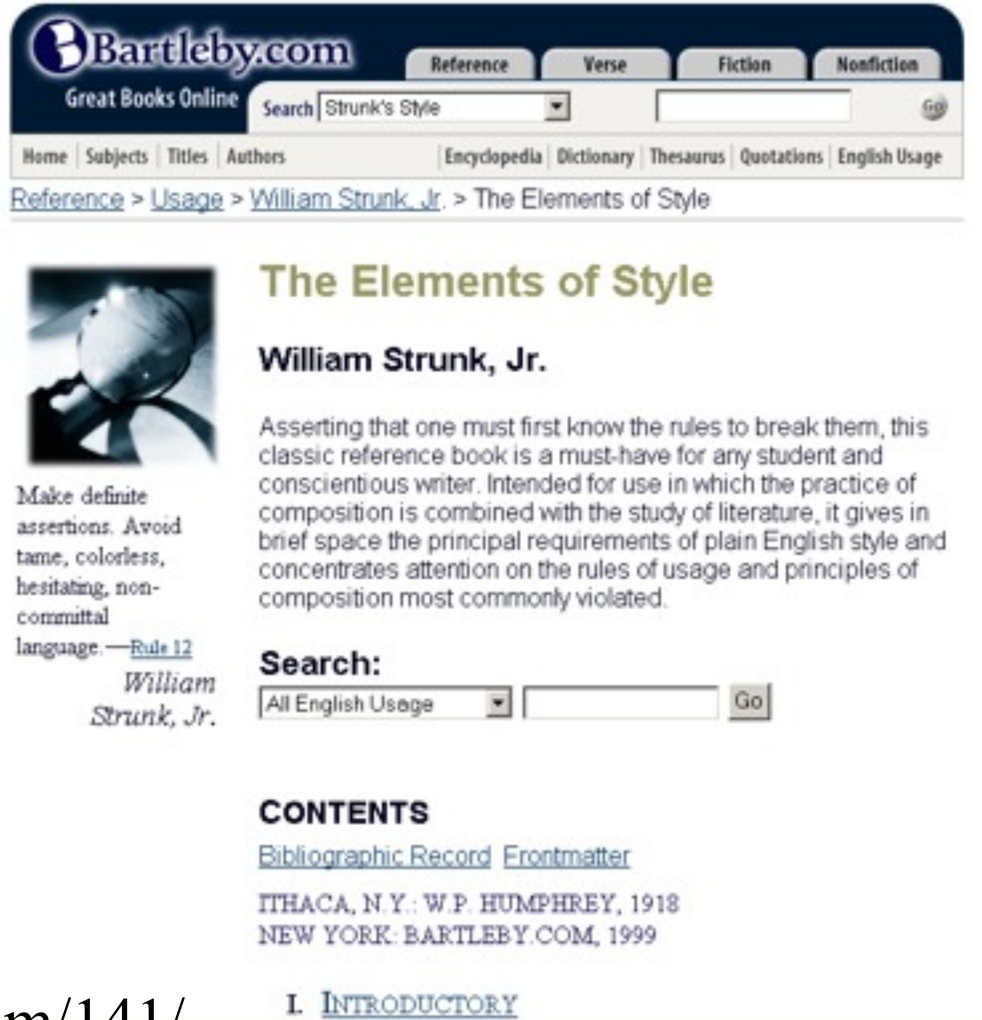
13. Many readers will skim over formulas on their first reading of your exposition. Therefore, your sentences should flow smoothly when all but the simplest formulas are replaced by "blah" or some other grunting noise.

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)...."

The elements of style, Stunk and White



http://www.bartleby.com/141/

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.

Figures

It should be easy to read the paper in a big hurry and still learn the main points.

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.

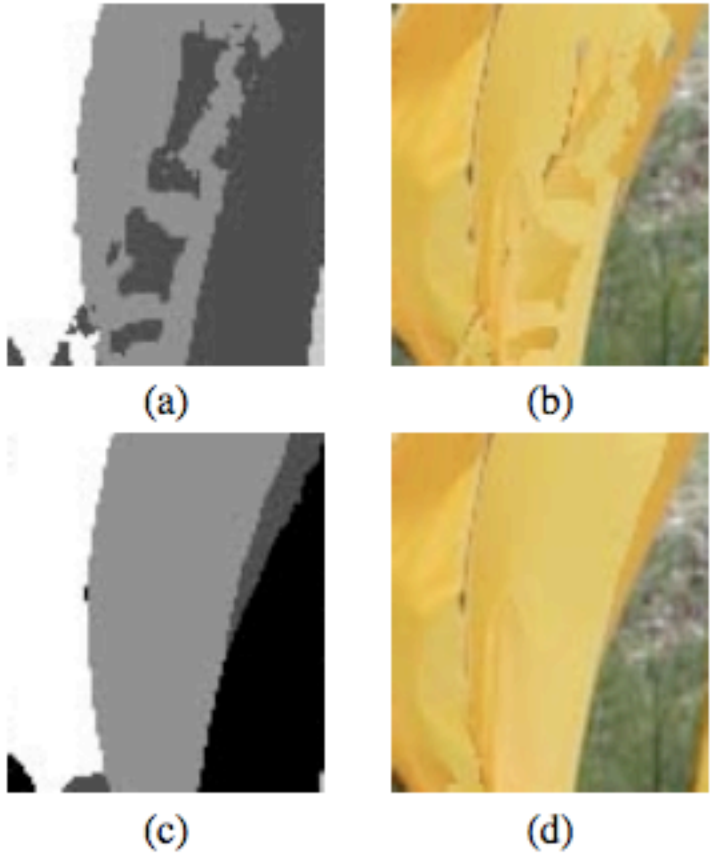
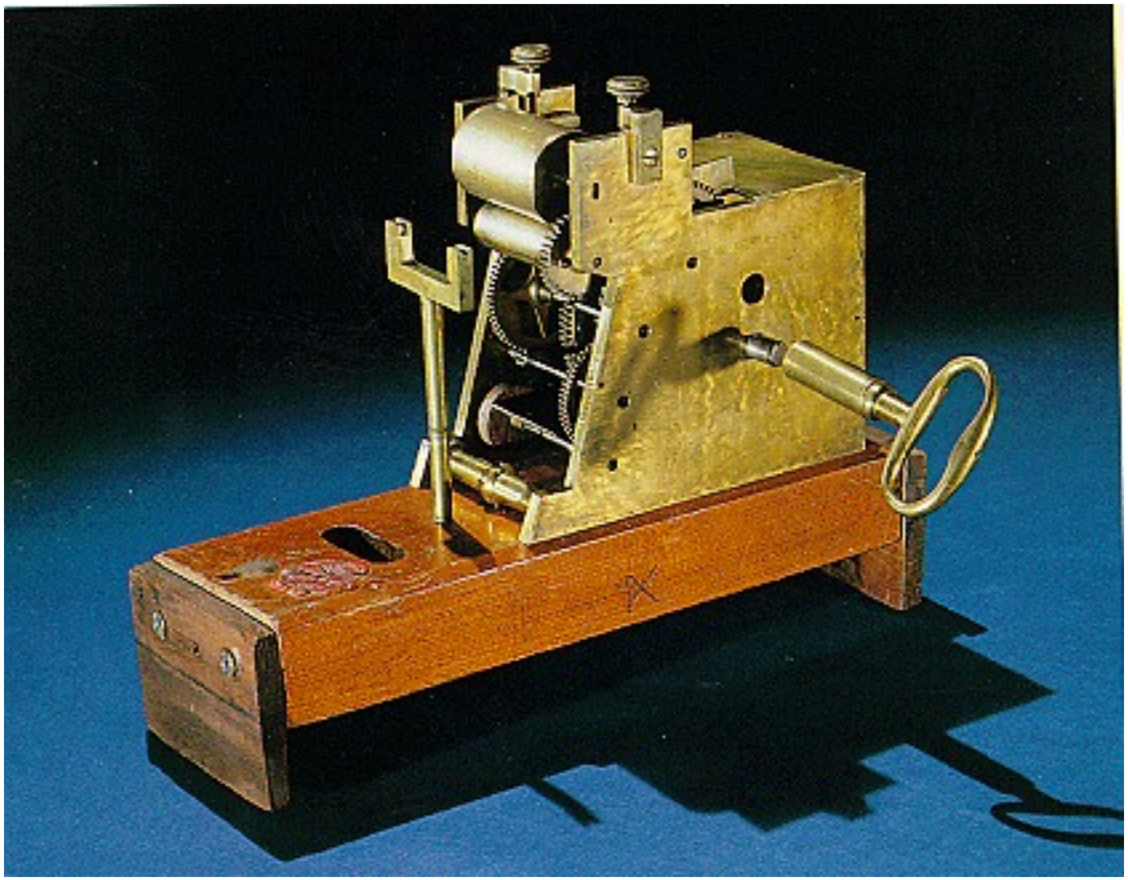


Figure 3: (a) Time-frame assignments for the front-most surface pixels, based on stereo depth measurements alone, without MRF processing. Grey level indicates the time-frame assignment at each pixel. (b) Shape-time image based on those assignments. (c) Most probable time-frame assignments, computed by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

Strategy tips

How do you evaluate this complex thing, this paper?

(and with 70-80% rejection rates, the question is, "How can I reject this paper?")



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

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

- About 1/3 are obvious rejects

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.

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.
- The rest are borderline, and these fall into two camps...

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

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

http://www.amazon.com/Fun-World-Costumes-Cockroach-Costume/dp/B0038ZQYRC

You try, but you can't find a way to kill this paper. While there's nothing too exciting about it, it's pretty well written, the reviews are ok, the results show an incremental improvement. Yet another kind of boring CVPR paper.

- The Cockroach

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

http://www.amazon.com/Fun-World-Costumes-Cockroach-Costume/dp/B0038ZQYRC

You try, but you can't find a way to kill this paper. While there's nothing too exciting about it, it's pretty well written, the reviews are ok, the results show an incremental improvement. Yet another kind of boring CVPR paper.

- The Cockroach
- The Puppy with 6 toes

http://www.imgion.com/white-cute-puppy/

A delightful paper, but with some easy-to-point-to flaw. This flaw may not be important, but it makes it easy to kill the paper, and sometimes you have to reject that paper, even though it's so fresh and wonderful.

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:

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 promise more than they deliver?

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 promise more than they deliver?
- Are there some important references that they don't mention (and therefore they're not up on the state-of-the-art for this problem)?

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 promise more than they deliver?
- Are there some important references that they don't mention (and therefore they're not up on the state-of-the-art for this problem)?
- Has their main idea been done before by someone else?

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 promise more than they deliver?
- Are there some important references that they don't mention (and therefore they're not up on the state-of-the-art for this problem)?
- Has their main idea been done before by someone else?
- Are the results incremental (too similar to previous work)?

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 promise more than they deliver?
- Are there some important references that they don't mention (and therefore they're not up on the state-of-the-art for this problem)?
- Has their main idea been done before by someone else?
- Are the results incremental (too similar to previous work)?
- Are the results believable (too different than previous work)?

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 promise more than they deliver?
- Are there some important references that they don't mention (and therefore they're not up on the state-of-the-art for this problem)?
- Has their main idea been done before by someone else?
- Are the results incremental (too similar to previous work)?
- Are the results believable (too different than previous work)?
- Is the paper poorly written?

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 promise more than they deliver?
- Are there some important references that they don't mention (and therefore they're not up on the state-of-the-art for this problem)?
- Has their main idea been done before by someone else?
- Are the results incremental (too similar to previous work)?
- Are the results believable (too different than previous work)?
- Is the paper poorly written?
- Do they make incorrect statements?

Promise only what you deliver

Promise only what you deliver

Learning local evidence for shading and reflectance

Matt Bell- and William T. Freeman
Mitsubishi Electric Research Labs (MERL)
201 Broadway
Cambridge, MA 02139

Abstract

A fundamental, unsolved vision problem is to distinguish image intensity variations caused by surface normal variations from those caused by reflectance changes–ie, to tell shading from paint. A solution to this problem is necessary for machines to interpret images as people do and could have many applications.

We take a learning-based approach. We generate a training set of synthetic images containing both shading and reflectance variations. We label the interpretations by indicating which coefficients in a steerable pyramid representation of the image were caused by shading and which by paint.

To analyze local image evidence for shading or reflectance, we study the outputs of two layers of filters, each followed by rectification. We fit a probability density model to the filter outputs using a mixture of factor analyzers. The resulting model indicates the probability, based on local image evidence, that a pyramid coefficient at any orientation and scale was caused by shading or by reflectance variations. We take the lighting direction to be that which generates the most shape-like labelling.

The labelling allows us to reconstruct bandpassed images containing only those parts of the input image caused

tensity changes are due to surface norm
construct spurious shapes when confront
changes. Here, we restrict ourselves to d
ing from paint.

Figure 1 (a) illustrates the problem.
intensity changes are caused by the graf
the intensity variations are caused by th
on which the paint was sprayed. Some l
effects. (b) shows the same location a
after an attempt was made to enforce a u
over the rock. It is simple to see the und
in the image (a); we want to develop a co
do the same thing.

This problem has not yet been solv
Sinha and Adelson [11] solved the pr
world domain, based on heuristic rules
tions and contours, which were pre-ident
other blocks world vision solutions, th
analogous solution for real images.

Freeman and Viola [4] proposed a p
shapes which penalized the elaborate sh
quired to explain images made by re
Their method assumed each image was
or all paint and couldn't process an ima

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

input images   quilting results

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

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.

342

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 Fleet, I knew I could trust it." [recent conference chair discussing some of the reasons behind a best paper prize].

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 second author on a great paper than first author 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

Our title

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

Sources on writing technical papers

I found this group most useful:

- 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

These were also helpful:

- 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, Fredo Durand, people.csail.mit.edu/fredo/PUBLI/writing.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

My first drafts are so-so, but I think I re-write pretty well. Good writing is re-writing. This means you need to start writing the paper early!