

Writing papers and giving talks

Bill Freeman
MIT CSAIL
May 2, 2011

5-minute presentations of your class projects

short homework due (describe the main points of your 5-minute presentation)

Schedule the last week

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

Time and location of final class presentations:

1:00pm - 3:30 or 4:00pm Wednesday

NOTE LOCATION: 3-343,

<http://web.mit.edu/registrar/classrooms/rooms/roompages/Buildings/Building3/3-343.html>

Outline

- writing technical papers
- giving technical talks

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

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.

Special journal issues have some of the advantages of both



IEEE Transactions on Pattern Analysis and Machine Intelligence

Call for Papers

Special Issue on

Graphical Models in Computer Vision

IMPORTANT DATES



- Paper Submission Deadline: 24 June 2002
- Acceptance Notification: 24 January 2003
- Final Manuscript: 24 February 2003
- Publication Date: July 2003



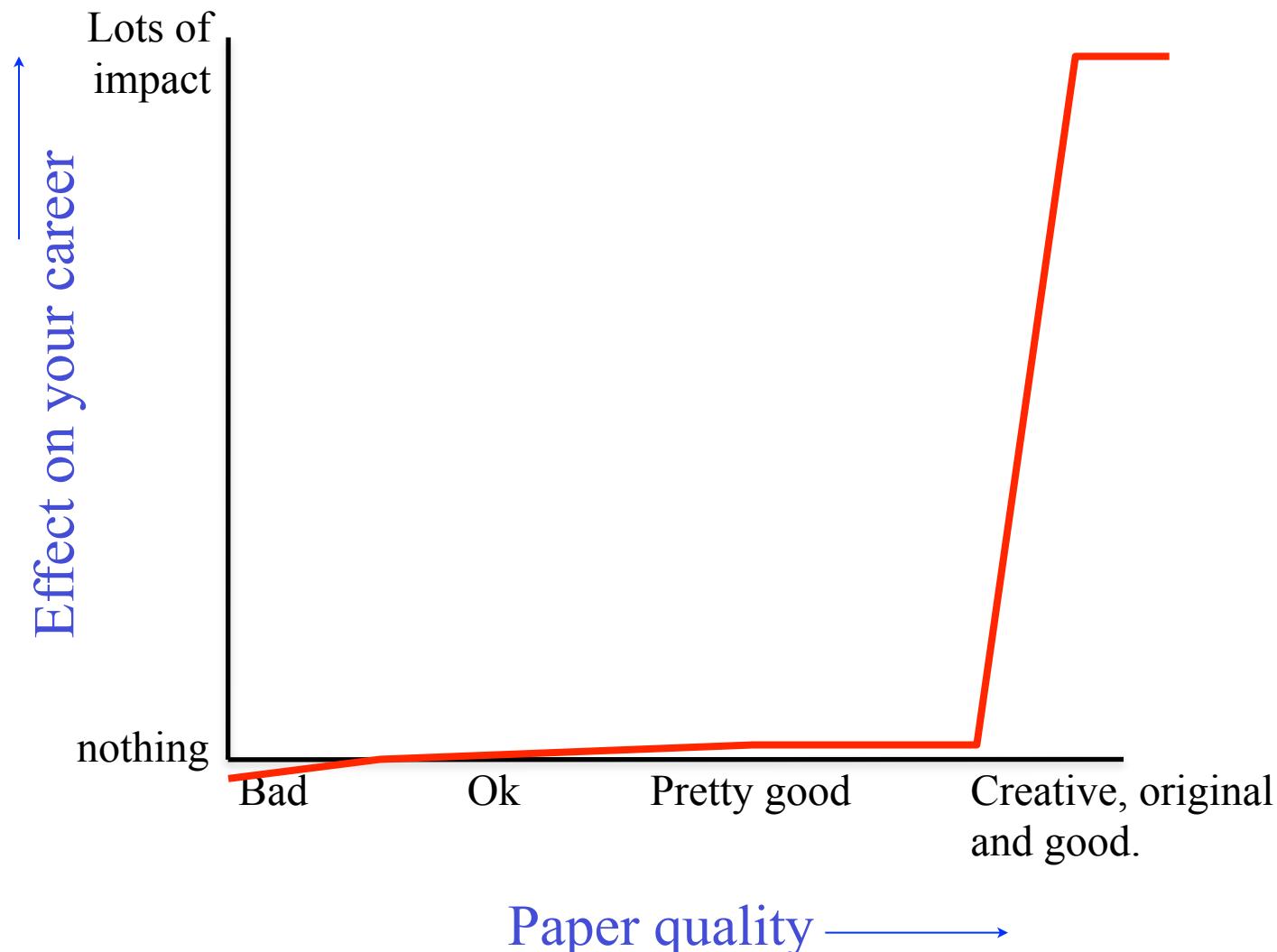
Conferences in computer vision and related areas

- CVPR/ICCV/ECCV (Computer Vision and Pattern Recognition/Intl. Conf. on Computer Vision/European Conf. on Computer Vision)
 - ~1600 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)
 - 650 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.

A paper's impact on your career



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

The screenshot shows a PDF document titled "Removing Camera Shake from a Single Photograph" by Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T. Roweis, and William T. Freeman. The document includes a comparison of three images: a blurry fountain statue on the left, a result from Photoshop's unsharp mask in the middle, and the authors' AI result on the right. The paper's abstract discusses the challenges of camera shake and the proposed method for removing it.

Removing Camera Shake from a Single Photograph

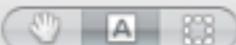
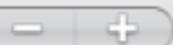
Rob Fergus¹ Barun Singh¹ Aaron Hertzmann² Sam T. Roweis² William T. Freeman¹

¹MIT CSAIL ²University of Toronto

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 hardware, can mitigate camera shake, but these are bulky and most consumer photographs are taken with a conventional, handheld camera. We may avoid the use of flash due to the unnatural tones it can result. In our experience, many of the otherwise favorite subjects of amateur photographers are spoiled by camera shake. Our goal is to remove that motion blur from a captured photograph, 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. Removing the unknown camera shake is thus a form of blind deconvolution, which is a problem with a long history in image and signal processing literature. In the most basic case, the problem is underconstrained: there are simply more equations than unknowns.



Zoom

Move Text Select

Sidebar

Search

Removing Camera Shake from a Single Photograph

Rob Fergus¹ Barun Singh¹ Aaron Hertzmann² Sam T. Roweis² William T. Freeman¹

¹MIT CSAIL ²University 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 cameras seldom can follow constrained 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 prevalent, 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 rephotographed 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 timescales 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 blur 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 essentially requires a small amount of user input.

Our reconstructions do contain artifacts, particularly when the

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 unsatisfactory, beyond touching up—in effect our method can help “rescue” shots that would have otherwise been completely lost.

2 Related Work

The task of deblurring an image is image deconvolution; 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 [Kondor and Hartmann 1998]. Existing blind deconvolution methods typically assume that the blur kernel has a simple parametric form, such as a Gaussian or low-frequency Fourier component. However, as illustrated by our examples, the blur kernels induced during camera shake do not have simple forms, and often contain very sharp peaks. 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 [Jalobea et al. 2002; Neelamani et al. 2006] 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; Tsunashima et al. 1994; Zarowin 1991], which have statistics quite different from the natural scenes we address in this paper. Performing blind deconvolution in this domain is usually straightforward, as the blurry image of an isolated star reveals the point-spread function.

Another approach is to assume that there are multiple images available of the same scene [Baudel et al. 1996; Rav-Acha and Peleg 2005]. Hardware approaches include optically stabilized lenses [Canon Inc. 2008], specially designed CMOS sensors [Liu and Gerspach 2001], and hybrid imaging systems [Bet-Elli and Noyar 2004]. Since we would like our method to work with existing cameras and imagery and to work for as many situations as possible, we do not assume that any such hardware or extra imagery is available.

Recent work in computer vision has shown the usefulness of heavy-tailed natural image priors in a variety of applications, including denoising [Roth and Black 2005], superresolution [Tappen et al. 2003], intrinsic images [Wills 2001], video matting [Apolloski and Fitzgibbon 2007], inpainting [Levin et al. 2005], and separating reflections [Lowe and Weiss 2004]. Each of these methods is effectively “non-blind”, in that the image formation process (e.g., the blur kernel in superresolution) is assumed to be known in advance.

Miskin and MacKay [2000] perform blind deconvolution on line art images using a prior on raw pixel intensities. Results are shown for small amounts of synthesized image blur. We apply a similar variational scheme for natural images using image gradients in place of intensities 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 \mathbf{B} , which is assumed to have been generated by convolution of a blur kernel \mathbf{K} with a latent image \mathbf{L} , plus noise:

$$\mathbf{B} = \mathbf{K} \otimes \mathbf{L} + \mathbf{N} \quad (1)$$

where \otimes denotes discrete image convolution (with non-periodic boundary conditions), and \mathbf{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 Gaussian approximation used in our experiments is shown in green.

the sensor irradiance. The latent image \mathbf{L} represents the image we would have captured if the camera had remained perfectly still; our goal is to recover \mathbf{L} from \mathbf{B} without specific knowledge of \mathbf{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 obey heavy-tailed distributions in their gradients [Field 1994]; the distribution of gradients has most of its mass on small values but gives significantly more probability to large values than a Gaussian distribution. This corresponds to the intuition that images often contain large sections of constant intensity or gentle intensity gradient interrupted by occasional large changes at edges or occlusion boundaries. For example, Figure 2 shows a natural image and a histogram of its gradient magnitudes. The distribution shows that the image contains primarily small or zero gradients, but a few gradients have large magnitudes. Recent image processing methods based on heavy-tailed distributions give state-of-the-art results in image denoising [Roth and Black 2005; Simoncelli 2005] and superresolution [Tappen et al. 2003]. In contrast, methods based on Gaussian prior distributions (including methods that use quadratic regularization) produce overly smooth images.

We represent the distribution over gradient magnitudes with a zero-mean mixture-of-Gaussian 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 optimization 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 \mathbf{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 \mathbf{B} to have been converted to a linear color space before processing. In our experiments, we applied inverse gamma-correction¹ 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 \mathbf{P} .

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

4.1 Estimating the blur kernel

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

As discussed in the previous section, the prior $p(\mathbf{V}\mathbf{L}_p)$ on the latent image gradients is a mixture of C zero-mean Gaussians (with variance λ_i) and weight π_i for the i -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 $\mathbf{V}\mathbf{P}$, we can write the posterior distribution over the unknowns with Bayes’ Rule:

$$p(\mathbf{K}, \mathbf{V}\mathbf{L}_p | \mathbf{V}\mathbf{P}) = p(\mathbf{V}\mathbf{P} | \mathbf{K}, \mathbf{V}\mathbf{L}_p) p(\mathbf{V}\mathbf{L}_p | \mathbf{K}) \quad (2)$$

$$= \prod_{i=1}^C p(\mathbf{V}\mathbf{L}_p(i) | \mathbf{K} \otimes \mathbf{V}\mathbf{L}_p(i), \sigma^2) \quad (3)$$

$$\prod_{d=1}^D p(\mathbf{K}(d) | \lambda_d) \quad (4)$$

where i indexes over image pixels and j indexes over blur kernel elements. \mathbf{X} and \mathbf{E} denote Gaussian and Exponential distributions respectively. For tractability, we assume that the gradients in $\mathbf{V}\mathbf{P}$ are independent of each other, as are the elements in $\mathbf{V}\mathbf{L}_p$ and \mathbf{K} .

A straightforward approach to deconvolution is to solve for the maximum a-posterior (MAP) solution, which finds the kernel \mathbf{K} and latent image gradients $\mathbf{V}\mathbf{L}_p$ that maximizes $p(\mathbf{K}, \mathbf{V}\mathbf{L}_p | \mathbf{V}\mathbf{P})$. This is equivalent to solving a regularized least-squares problem that attempts to fit the data while also minimizing small gradients. We tried this (using conjugate gradient search) but found that the algorithm failed. One interpretation is that the MAP objective function attempts to minimize all gradients (even large ones), whereas we expect natural images to have some large gradients. Consequently, the algorithm yields a two-tone image, since virtually all the gradients are zero. If we reduce the noise variance (thus increasing the weight on the data-fitting term), then the algorithm yields a deblurring function for \mathbf{K} , which exactly fits the blurred image, but without any deblurring. Additionally, we find the MAP objective function to be very susceptible to poor local minima.

Instead, our approach is to approximate the full posterior distribution $p(\mathbf{K}, \mathbf{V}\mathbf{L}_p | \mathbf{V}\mathbf{P})$, and then compute the kernel \mathbf{K} with maximum marginal probability. This method selects a kernel that is most likely with respect to the distribution of possible latent images, thus avoiding the overfitting that can occur when selecting a single “best” estimate of the image.

In order to compute this approximation efficiently, we adopt a variational Bayesian approach [Jordan et al. 1999] which computes a distribution $p(\mathbf{K}, \mathbf{V}\mathbf{L}_p)$ that approximates the posterior $p(\mathbf{K}, \mathbf{V}\mathbf{L}_p | \mathbf{V}\mathbf{P})$. In particular, our approach is based on Miskin and MacKay’s algorithm [2000] for blind deconvolution of cartoon images. A factored representation is used: $p(\mathbf{K}, \mathbf{V}\mathbf{L}_p) = p(\mathbf{K}|p(\mathbf{V}\mathbf{L}_p))$. For the latent image gradients, this approximation is a Gaussian density, while for the non-negative blur kernel elements, it is a truncated Gaussian. The distributions for each latent gradient and blur kernel element are represented by their mean and variance, stored in an array.

Following Miskin and MacKay [2000], we also treat the noise variance σ^2 as an unknown during the estimation process, thus freeing the user from tuning this parameter. This allows the noise variance to vary during estimation: the data-fitting constraint is loose early in the process, becoming tighter as better, low-noise solutions are found. We place a prior on σ^2 , in the form of a Gamma distribution on the inverse variance, having hyper-parameters a, b : $p(\sigma^2 | a, b) = T^{(a-2)/2} a^{-b/2} / \Gamma(a)$. The variational posterior of σ^2 is $q(\sigma^2 | \mathbf{V}\mathbf{P})$, another Gamma distribution.

The variational algorithm minimizes a cost function representing the distance between the approximating distribution and the true posterior, measured as: $KL(q(\mathbf{K}, \mathbf{V}\mathbf{L}_p, \sigma^2) || p(\mathbf{K}, \mathbf{V}\mathbf{L}_p | \mathbf{V}\mathbf{P}))$. The independence assumption in the variational posterior allows the cost function C_{KL} to be factored:

$$<\log \frac{q(\mathbf{V}\mathbf{L}_p)}{p(\mathbf{V}\mathbf{L}_p)}>_{\mathbf{V}\mathbf{L}_p} >_C C_{KL} <\log \frac{q(\mathbf{K})}{p(\mathbf{K})}>_{\mathbf{K}} + <\log \frac{q(\sigma^2)}{p(\sigma^2)}>_{\sigma^2} \quad (5)$$

where $<\cdot>_{\mathbf{V}\mathbf{L}_p}$ denotes the expectation with respect to $q(\mathbf{V}\mathbf{L}_p)$. For brevity, the dependence on $\mathbf{V}\mathbf{P}$ is omitted from this equation.

The cost function is then minimized as follows. The means of the distributions $q(\mathbf{K})$ and $q(\mathbf{V}\mathbf{L}_p)$ are set to the initial values of \mathbf{K} and $\mathbf{V}\mathbf{L}_p$, and the variance of the distributions is high, reflecting the lack of certainty in the initial estimate. The parameters of the distributions are then updated alternately by coordinate descent; one is updated by marginalizing out over the other whilst incorporating the model priors. Updates are performed by computing closed-form optimal parameter updates, and performing line-search in the direction of these updated values (see Appendix A for details). The updates are repeated until the change in C_{KL} becomes negligible. The mean of the marginal distribution $<\mathbf{K}>_{q(\mathbf{K})}$ is then taken as the final value for \mathbf{K} . Our implementation adapts the source code provided online by Miskin and MacKay [2000].

In the formulation outlined above, we have neglected the possibility of saturated pixels in the image, an awkward non-linearity which violates our model. Since dealing with them explicitly is complicated, we prefer to simply mask out saturated regions of the image during the inference procedure, so that no use is made of them.

For the variational framework, $C = D = 4$ components were used in the prior on \mathbf{K} and $\mathbf{V}\mathbf{L}_p$. The parameters of the prior on the latent image gradients $\mathbf{V}\mathbf{L}_p$ were estimated from a single street scene image, shown in Figure 2, using EM. Since the image statistics vary across 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-sensor-kernel images 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, \mathbf{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 algorithm, while holding \mathbf{K} fixed.

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

² For example, $<\sigma^{-2}>_{q(\sigma^2)} = \int_{\sigma^2} \sigma^{-2} T(\sigma^{-2}) q(\sigma^2 | a, b) d\sigma^2$.

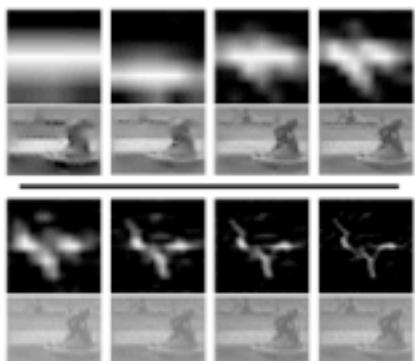


Figure 3: The multi-scale inference scheme operating on the fountain image in Figure 1. *1st & 3rd rows:* The estimated blur kernel at each scale level. *2nd & 4th rows:* Estimated image patch at each scale. The intensity image was reconstructed from the gradients used in the inference using Poisson 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 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 \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 Gerencser [1992], Neelamani [2004] and van Cittert [Zanowin 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 lenses 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 $\nabla \mathbf{L}$, while holding \mathbf{K} fixed. The intensity image was then formed via Poisson image reconstruction [Wain 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} . Iterating 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 whiteboard⁵ 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

⁵Camera-to-whiteboard distance was $\approx 5m$. Lens focal length was 50mm mounted on a 0.6x DSLR sensor.

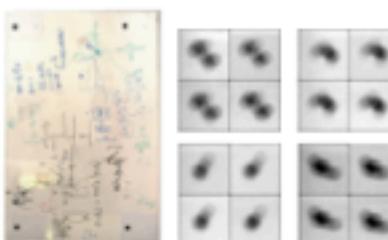


Figure 4: *Left:* The whiteboard test scene with dots in each corner. *Right:* Dots from the corners of images taken by different people. Within each image, the dot trajectories are very similar suggesting that image blur is well modeled as a spatially invariant convolution.



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 user.

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 trilobed blur, 30 pixels in size (shown in Figure 10), being deblurred.



Figure 6: *Top:* A scene with complex motions. While the camera is small, the child is both translating and in 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 only revealed in the image by the trajectory of a point light source formed by the blur. This gives us an opportunity to compare inferred blur kernel with the true one. Figure 10 shows image structures, along with the inferred kernels from the five images.

We also compared our algorithm against existing blind deconvolution algorithms, running Matlab's deconvolution routine provides implementations of the methods of Biggs and Jansson [1997]. Based on the iterative Richardson scheme, these methods also estimate the blur kernel, alternating holding the blur constant and updating the image and vice versa. The results of this algorithm, applied to the fountain scenes are shown in Figure 11 and are poor compared to ours, shown in Figures 1 and 13.

Images with significant saturation. Figures 12 and 13 are images where the true intensities are not observable due to the dynamic range limitations of the camera. The user patch used for kernel analysis must avoid the large saturated regions. While the deblurred image does have some unsaturated regions, the unsaturated regions can still be ex-



Zoom

Move Text Select Sidebar Search



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 a child from another image of the family (different from Figure 7). In the deblurred image, the text on his jersey is now legible.

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 tonemap, 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

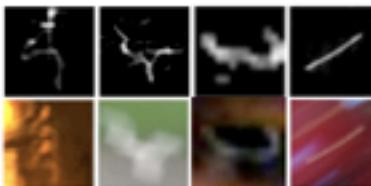


Figure 10: Top row: Inferred blur kernels from four real images (the cafe, fountain and family scenes plus another image not shown). Bottom row: Patches extracted from these scenes where the true kernel has been revealed. In the cafe image, two lights give a dual image of the kernel. In the fountain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

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 artifact 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 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 to thank the following people for supplying us with blurred images for the paper: Omar Khan, Reinhard Klette, Michael Lewinski, Pietro Perona and Elizabeth Van Rostenberg. Funding for the project was provided by NSERC, NGA NSIG-1582-04-0004 and the Shell Cross.

References

- ANJALIEWAN, N., AND DURRANT-SCOTT, A.: 2005, Bayesian video warping using learnt image priors, In *Conf. on Computer Vision and Pattern Recognition*, 407–414.

BAILIE, B., BLAKE, A., AND ZUHOURI, S.: 1998, Motion Deblurring and Super-resolution from 3D Image Sequences, In *ECCV*, 473–482.

BALI-EDEKA, M., AND NAYAK, S. K.: 2004, Motion Based Motion Deblurring, *IEEE Trans. on Pattern Analysis and Machine Intelligence* 26, 5, 659–675.

BEGIS, D., AND ANDREWS, M.: 1995, Acceleration of iterative image restoration algorithms, *Applied Optics* 34, 17, 1768–1775.

CANNON INTEL, 2005, What is spatial image multithread? <http://www.cannon.com/index/TechPapers/Intel.html>.

CARON, J., NAMELAKI, N., AND BULLING, C.: 2002, Noniterative blind deconvolution by use of extracted filter function, *Applied Optics* 41, 32 (November), 684–694.

FELD, D.: 1994, What is the goal of motion coding? *Annals of Computation* 1, 559–665.

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.

GILL, S.: 1996, Bayesian inductive inference and maximum entropy, In *Maximum Entropy and Bayesian Methods*, T. Skilling Ed. Kluwer, 56–71.

GRADSHTEYN, A., BLUM-FRAUD, L., AND EHRLICH, J.: 2002, Estimation of blur and noise parameters in sensor sensing, In Proc. of the Conf. on Acoustics, Speech and Signal Processing.

HARRISON, P. A.: 1987, *Bayesian Combination of Images and Spectra*, Academic Press.

HORNIK, M., GHOSHAL, S., BANERJEE, T., AND SAUL, L.: 1998, An introduction to variational methods for graphical models, In *Machine Learning*, vol. 37, 185–233.

KENDON, J., AND HATZINIKOULIS, D.: 1998, Blind image deconvolution, *IEEE Signal Processing Magazine* 15, 3 (May), 45–64.

LEVY, A., AND WEISS, Y.: 2004, Using Attenuated Separation of Reflections from a Single Image Using a Sparsity Prior, In *IJCVA*, vol. 1, 613–623.

LEVY, A., ZEMER, A., AND WEISS, Y.: 2005, Learning How to Inpaint from Global Depth Statistics, In *IJCVA*, 305–312.

LEVY, X., AND GAMAL, A.: 2003, Simultaneous image denoising 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* 7, 747–754.

MERIKO, J., AND NEAUKH, D. J. C.: 2003, Ensemble Learning for Blind Image Separation and Reconstruction, In *Adv. in Independent Component Analysis*, M. Girolami, Ed. Springer-Verlag.

MERIKO, J.: 2003, Thesis ensemble theory, http://www.inference.phy.cam.ac.uk/jewel030/thesis_merikola.tar.gz.

MERIKO, J.: 2005, Ensemble Learning for Independence Component Analysis, PhD thesis, University of Cambridge.

NEALANDER, B., COELHO, E., AND BANERJEE, S.: 2004, General Transfer learning regularized deconvolution for ill-conditioned systems, *ICIP2004*, Int. Conf. on Signal Processing, 427–430.

RETHFELD, A., AND PHAM, B.: 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 12, 55–58.

ROTH, S., AND BLACK, M. J.: 2002, Fields of Experts: A Framework for Learning Image Priors, In *CVPR*, vol. 2, 890–897.

SEPHILOGLU, E. P.: 2005, Statistical modeling of photographic images, In *Handbook of Image and Multimed. Processing*, A. Boulo, Ed., ch. 4.

TAPEN, P. M., RUECKL, B. C., AND FREIDMAN, W. T.: 2003, Exploiting the sparse structure prior for super-resolution and image denoising, In *IJCVA*.

THIRUMALA, P., MURA, N., AND BABA, N.: 1994, Iterative blind deconvolution method using Lucy's algorithm, *Appl. Optics* 33, 212 (February), 769–778.

WEISS, Y.: 2003, Denoising multi-frame images from image sequences, In *IJCVA*, 68–75.

ZACHOROV, C.: 1994, Robust, noniterative, and computationally efficient localisation of van-Cittert deconvolution optical signal refraction, *Journal of the Optical Society of America A* 11, 10 (October), 2021–2031.

Appendix A

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

Algorithm 1 Image Delimit

```

Requires: Image map R, selected sub-windows P, maximum bit size k, overall bit
dimension M = 0 for binary, 1 for soft; parameter for prior on VL:  $\theta_0 = [\theta_1^T, \theta_2^T]$ ;
parameters for prior on K:  $\theta_K = [\theta_{K1}^T, \theta_{K2}^T]$ .
Convert F to grayscale.
Inverse gamma correct F (stabilize  $\mu = 2.2$ ).
 $YF = F \otimes [1, -1]$ . % Compute gradient in x
 $YB = F \otimes [1, 0]^T$ . % Compute gradient in y
 $YFB = [YF, YB]^T$ . % Concatenate gradients
 $S = 1 - \text{log}(1/\|YFB\|)$ . % # of scales, starting with 1 x 3 kernel
For  $i = 1$  to  $S-1$  do
    % Loop over scales, starting at coarsest
     $YFB_i = \text{convolve}(YFB, \frac{1}{S_i} \otimes [1, 1, 1, 1, 1, 1]^T)$ . % Convolve gradient
    If  $i > 1$  then
        % Reduce kernel and produce
         $K^i = D(K^i, 1, 1, 1, 1, 1, 1)^T$ ;  $H^i = H^i \otimes K^i$ ;  $K^i = [K^i]^T$ .
         $[K^i, VL^i] = \text{Inference}(YFB_i, YFB, P, R, \theta_0)$ , keeping  $K^i$  fixed.
    else
        % Compute estimate from previous scale
         $VL^i_0 = \text{inference}(YFB_i, \frac{1}{S_i} \otimes [1, 1, 1, 1, 1, 1]^T)$ .
         $K^i = \text{Inference}(K^{i-1}, \frac{1}{S_i} \otimes [1, 1, 1, 1, 1, 1]^T)$ .
    end if
     $[K^i, VL^i] = \text{Inference}(YFB_i, K^i, VL^i, R, \theta_0)$ . % Run inference
end for

% Selection of K^i that are less than max(K^i)/C to zero. % Threshold kernel
R = 0.5 * diag(K^S). % R = 0.5 times diagonal of K^S
% Reduce step along
L = diag(1/(K^S * R)). % L = diag(1/(K^S * R))
% Run AL for 10 iterations
Gauss correct L (stabilize  $\mu = 2.2$ ).
Histogram match K to R using histeq.
Denoise R, K^S.

```

Algorithm 2 Inference (adapted from Minka and Winkler 2005)

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.

Kajiyा 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)

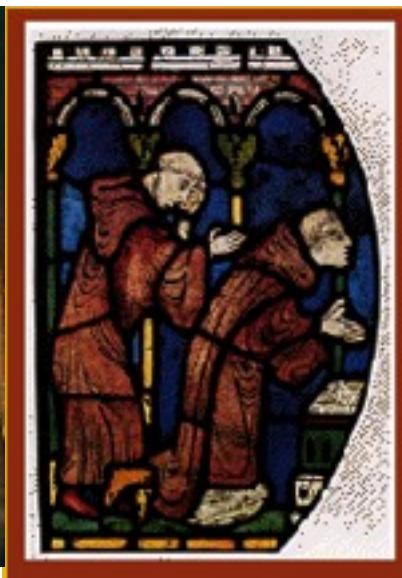
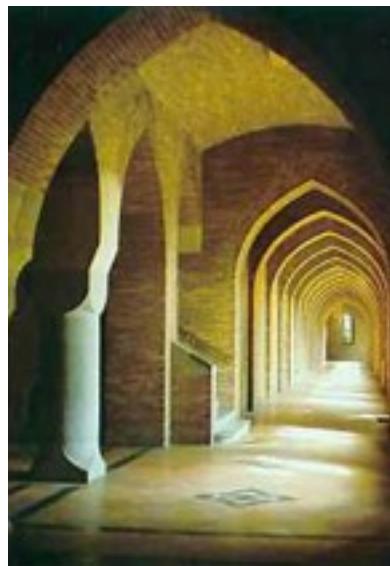
“The emphasis on both speed and quality makes the reviewing process for SIGGRAPH very different from of a journal or another conference.

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.”

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/>

Kajiyा 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

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.

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: 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.

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).

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

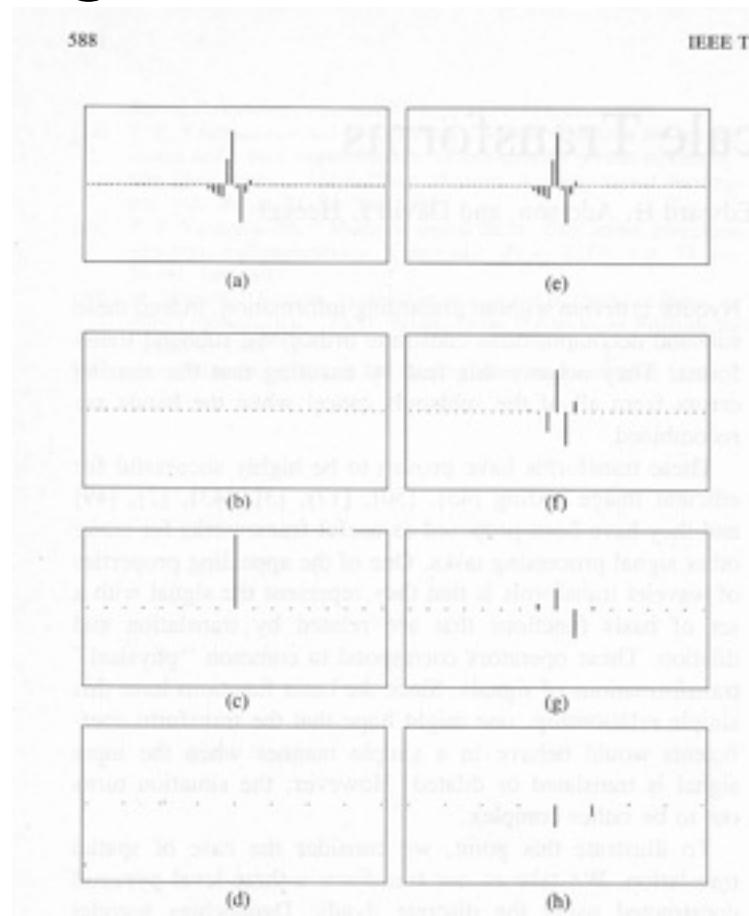


Fig. 1. Effect of translation on the wavelet representation of a signal. (a) Input signal, which is equal to one of the wavelet basis functions. (b)-(d) Decomposition of the signal into three wavelet subbands. Plotted are the coefficients of each subband. Dots correspond to zero-value coefficients. (e) Same input signal, translated one sample to 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

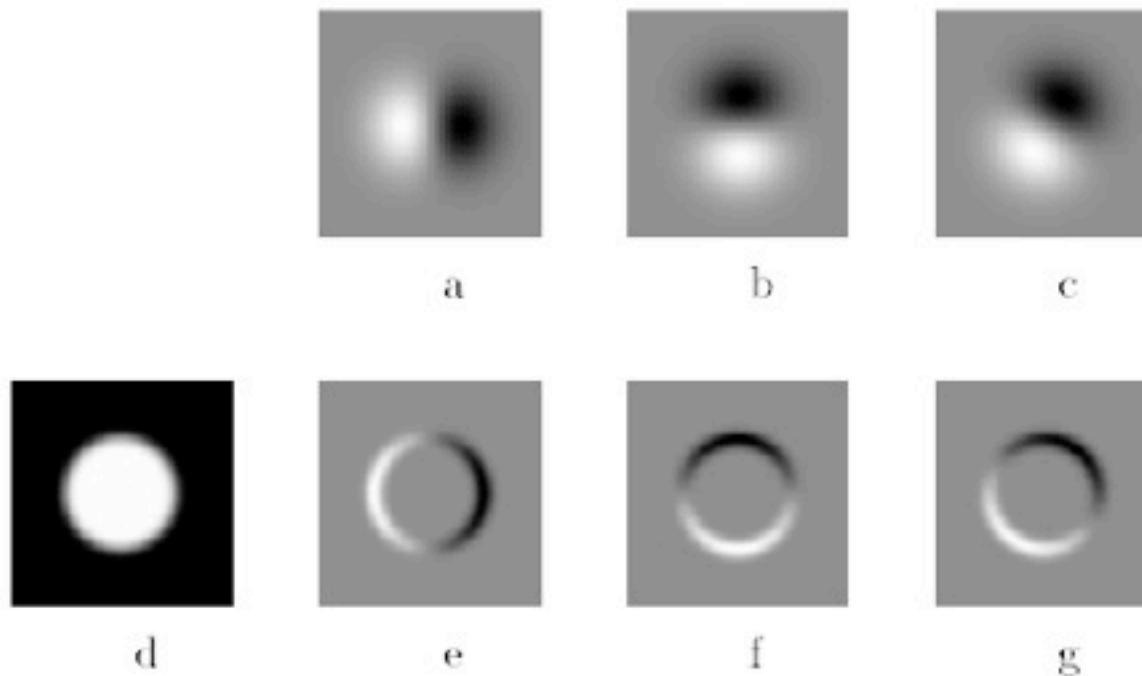


Fig. 1. Example of steerable filters: (a) $G_1^{0^\circ}$ first derivative with respect to x (horizontal) of a Gaussian; (b) $G_1^{90^\circ}$, which is $G_1^{0^\circ}$, rotated by 90° . From a linear combination of these two filters, one can create G_1^θ , which is an arbitrary rotation of the first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

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

24. The opening paragraph should be your best paragraph, and its first sentence should be your best sentence. If a paper starts badly, the reader will wince and be resigned to a difficult job of fighting with your prose. Conversely, if the beginning flows smoothly, the reader will be hooked and won't notice occasional lapses in the later parts.
Probably the worst way to start is with a sentence of the form "An *x* is *y*." For example,

Bad: An important method for internal sorting is quicksort.

Good: Quicksort is an important method for internal sorting, because ...

Bad: A commonly used data structure is the priority queue.

Good: Priority queues are significant components of the data structures needed for many different applications.

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.

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?



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

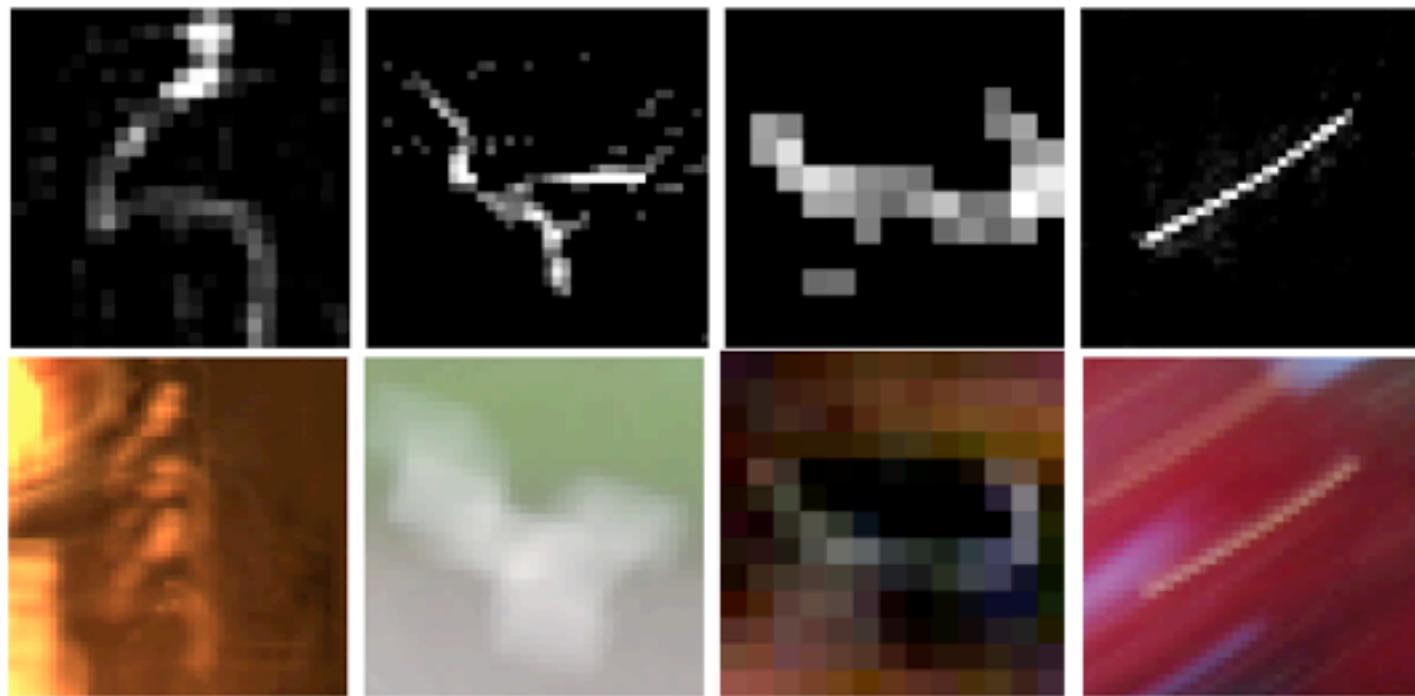


Figure 10: *Top row*: Inferred blur kernels from four real images (the cafe, fountain and family scenes plus another image not shown). *Bottom row*: Patches extracted from these scenes where the true kernel has been revealed. In the cafe image, two lights give a dual image of the kernel. In the fountain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

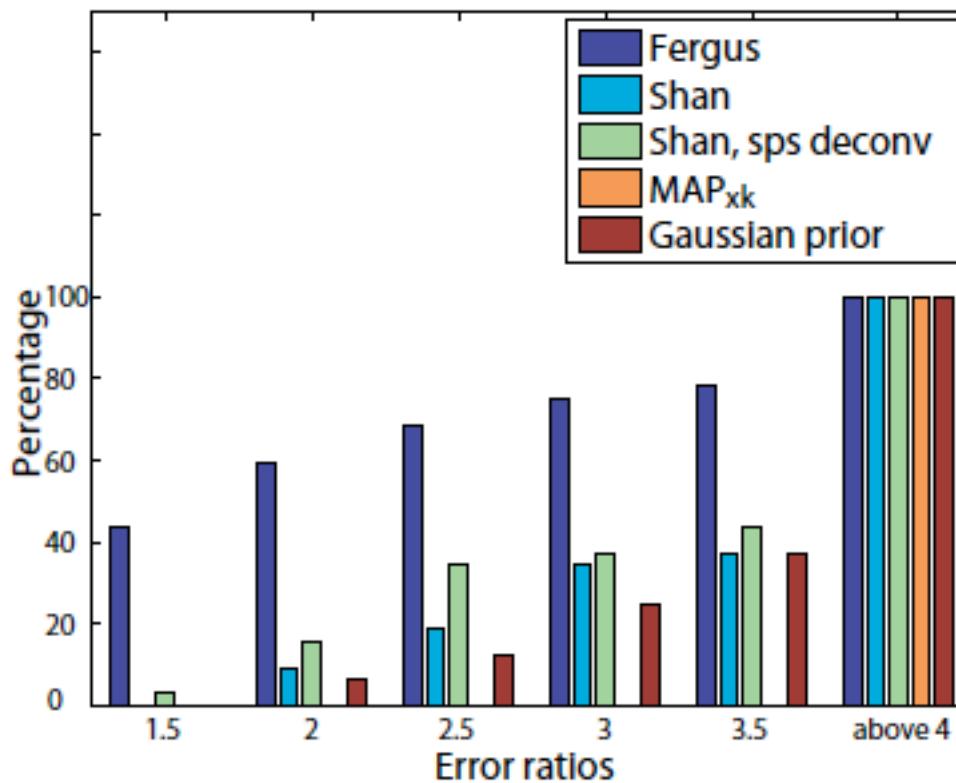


Figure 9. Evaluation results: Cumulative histogram of the deconvolution error ratio across test examples.

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 \mathbf{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

Bartleby.com
Great Books Online

Reference Verse Fiction Nonfiction

Search: Strunk's Style Go

Home | Subjects | Titles | Authors Encyclopedia | Dictionary | Thesaurus | Quotations | English Usage

Reference > Usage > William Strunk, Jr. > The Elements of Style



Make definite assertions. Avoid tame, colorless, hesitating, non-committal language.—[Rule 12](#)

William
Strunk, Jr.

The Elements of Style

William Strunk, Jr.

Asserting that one must first know the rules to break them, this classic reference book is a must-have for any student and conscientious writer. Intended for use in which the practice of composition is combined with the study of literature, it gives in brief space the principal requirements of plain English style and concentrates attention on the rules of usage and principles of composition most commonly violated.

Search:

All English Usage Go

CONTENTS

[Bibliographic Record](#) [Frontmatter](#)

ITHACA, N.Y.: W.P. HUMPHREY, 1918
NEW YORK: BARTLEBY.COM, 1999

I. [INTRODUCTORY](#)

<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.

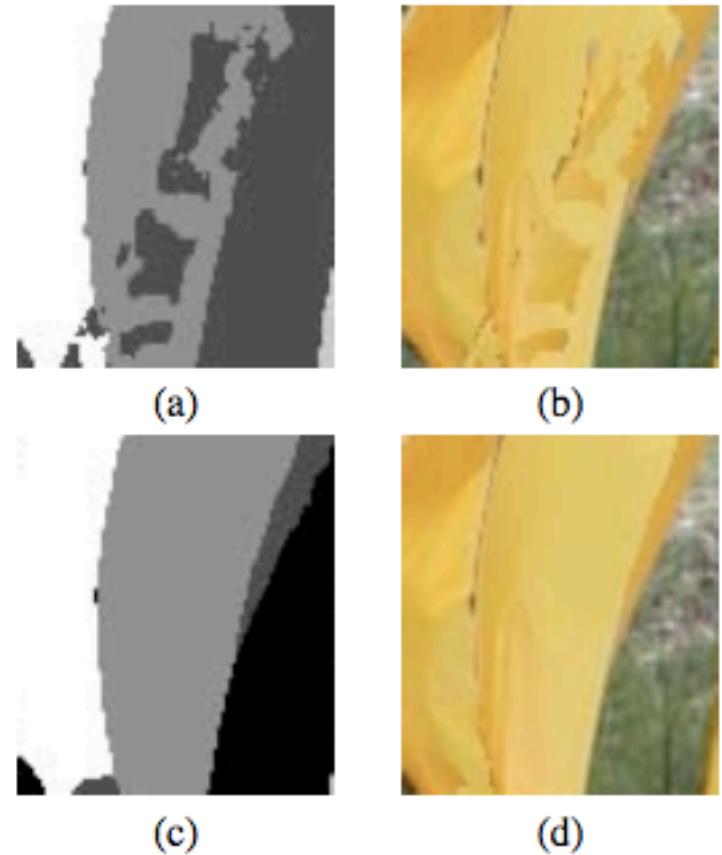
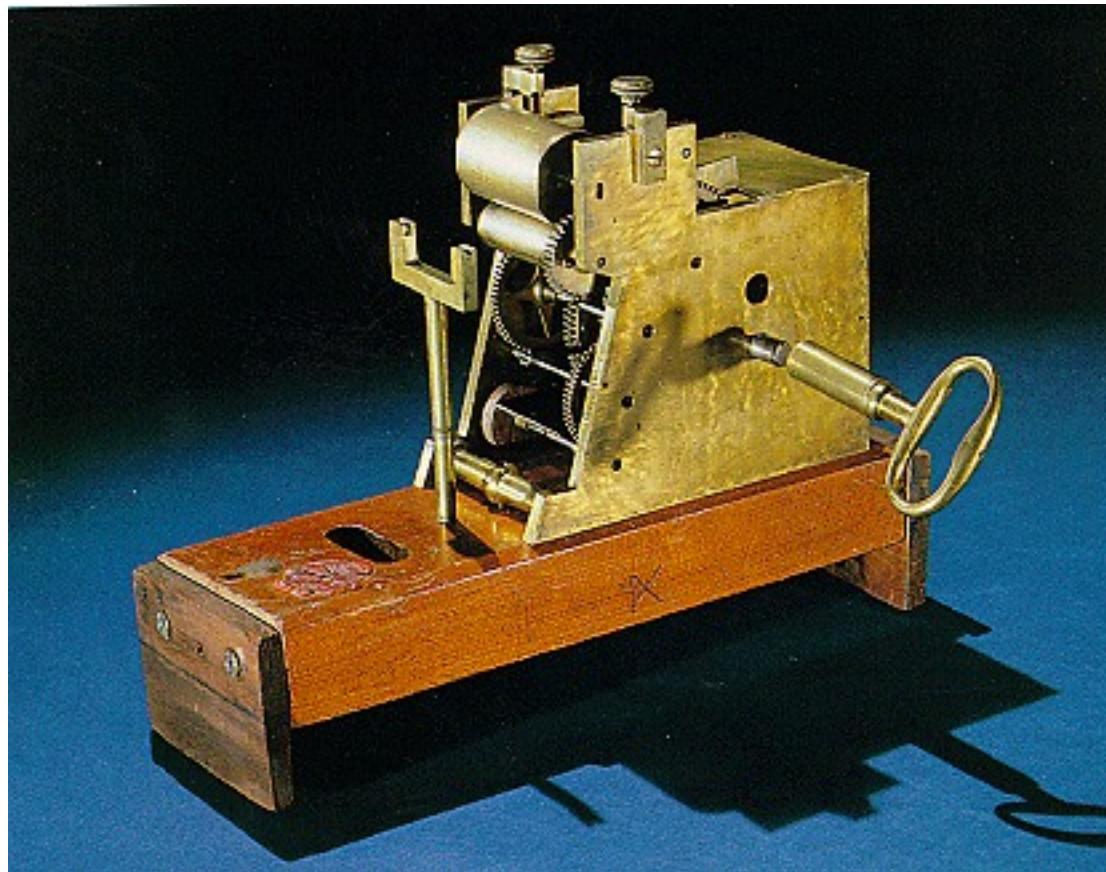


Figure 3: (a) Time-frame assignments for the front-most surface pixels, based on stereo depth measurements alone, without MRF processing. Grey level indicates the time-frame assignment at each pixel. (b) Shape-time image based on those assignments. (c) Most probable time-frame assignments, computed by 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?”)



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 graft the intensity variations are caused by the on which the paint was sprayed. Some h effects. (b) shows the same location a after an attempt was made to enforce a t over the rock. It is simple to see the unde in the image (a); we want to develop a cc do the same thing.

This problem has not yet been solve 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 sl 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. Freeman²

¹University of California, Berkeley

²Mitsubishi Electric Research Laboratories

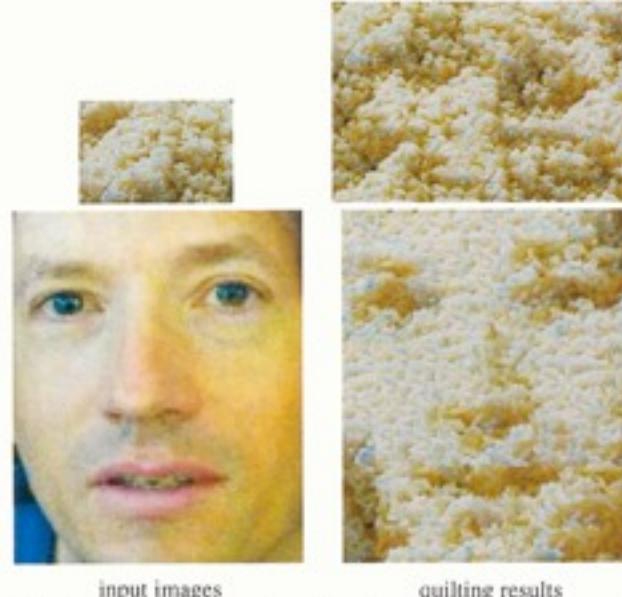
Abstract

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

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?

Moon paper issues

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?



Our title

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

Everything that matters, except for content

Carven von Bearnensquash
Department of Computer Science
University of Phoenix

bearensquash@live.com

Abstract

Peer reviews of conference paper submissions is an integral part of the research cycle, though it has unknown origins. For the computer vision community, this process has become significantly more difficult in recent years due to the volume of submissions. For example, the number of submissions to the CVPR conference has tripled in the last ten years. For this reason, the community has been forced to reach out to a less than ideal pool of reviewers, which unfortunately includes uninformed junior graduate students, disgruntled senior graduate students, and tenured faculty. In this work we take the simple intuition that the quality of a paper can be estimated by merely glancing through the general layout, and use this intuition to build a system that employs basic computer vision techniques to predict if the paper should be accepted or rejected. This system can then be used as a first cascade layer during the review process. Our results show that while rejecting 15% of “good papers”, we can cut down the number of “bad papers” by more than 50%, saving valuable time of reviewers. Finally, we fed this very paper into our system and are happy to report that it received a posterior probability of 88.4% of being “good”.

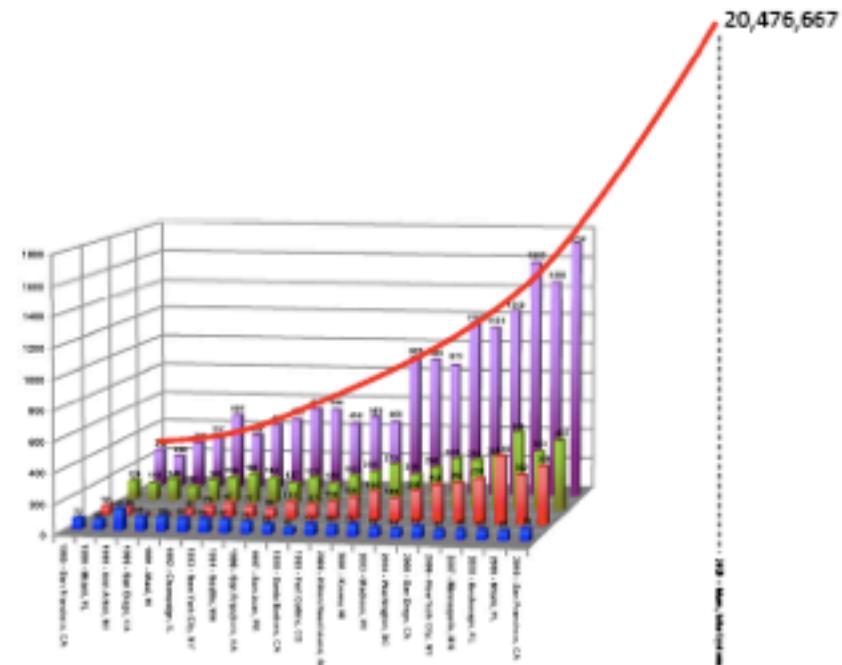


Figure 1. Paper submission trends. The number of submitted papers to CVPR, and other top tier computer vision conferences, is growing at an alarming rate. In this paper we propose an automated method of rejected sub-par papers, thereby reducing the burden on reviewers.

and tenured faculty. Although many excellent research pa-

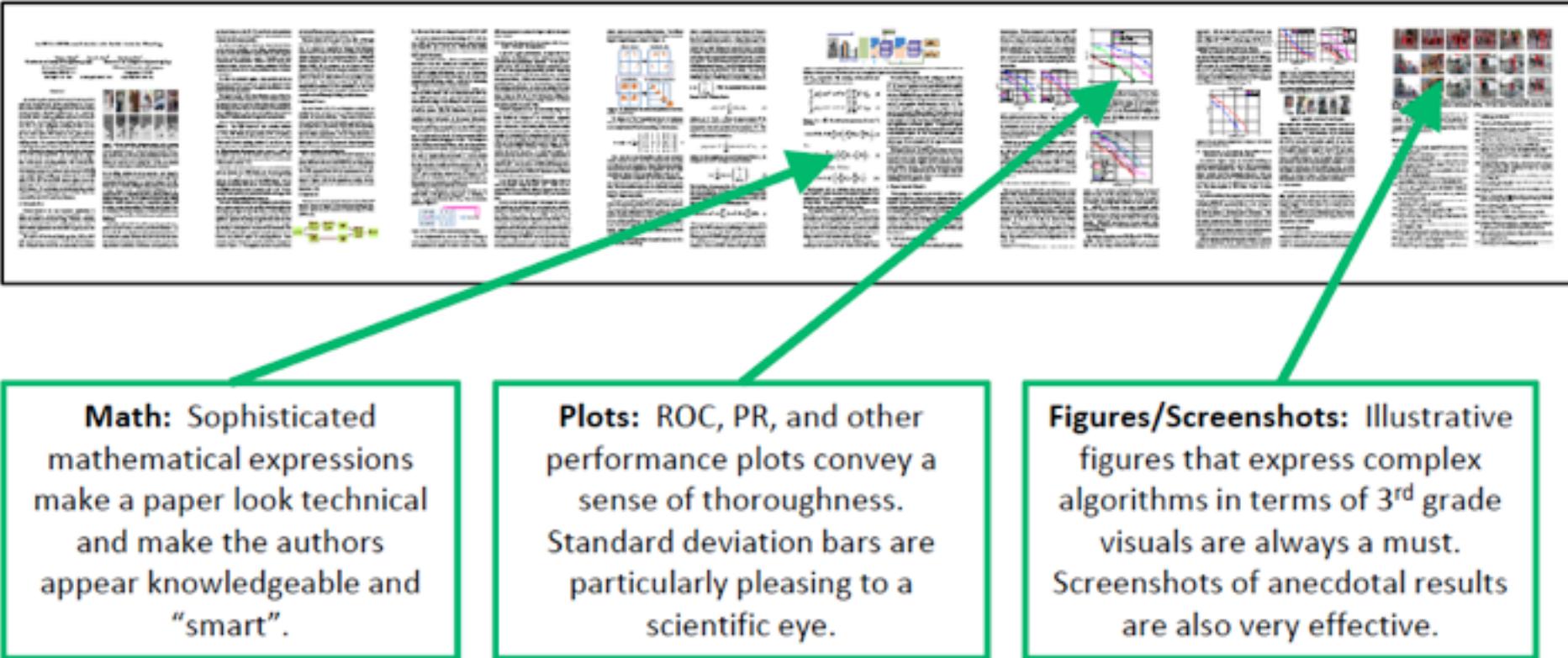


Figure 6. Characteristics of a “Good” paper.

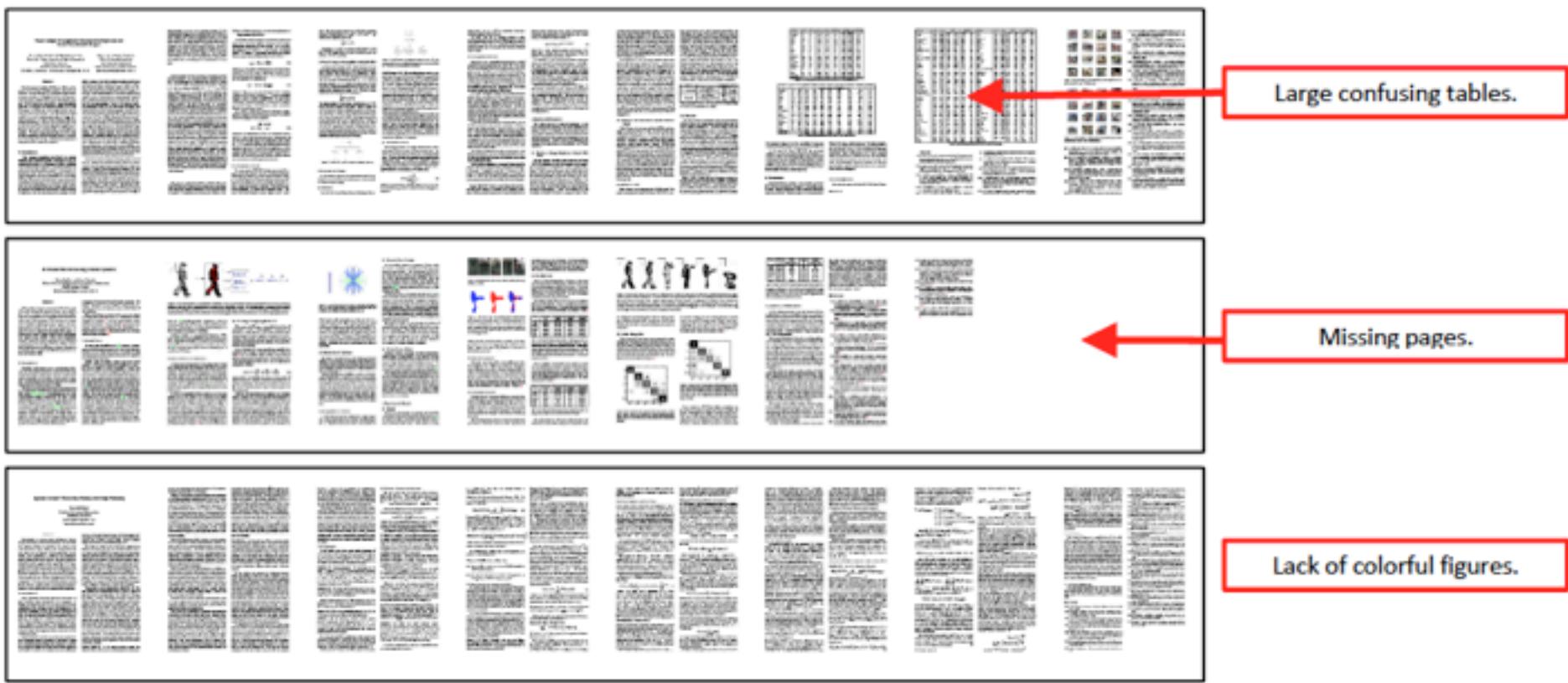


Figure 7. Characteristics of a “Bad” paper.

Though this does not affect our system directly, we have found that equations improve paper gestalt and thus increase the chances of this paper being accepted (*c.f.* Figure 6). While this is a good start, we believe that the amount of math in this paper is still not adequate (*c.f.* Figure 6). Therefore, purely for aesthetic purpose, we reproduce Maxwell's equations below [12]:

$$\oint_{\text{closed surface}} \vec{E} \cdot d\vec{A} = \frac{Q_{enc}}{\epsilon_0} \quad (4)$$

$$\oint_{\text{closed surface}} \vec{B} \cdot d\vec{A} = 0 \quad (5)$$

$$\oint \vec{E} \cdot d\vec{s} = -\frac{d\phi_B}{dt} \quad (6)$$

$$\oint \vec{B} \cdot d\vec{s} = \mu_0 \epsilon_0 \frac{d\phi_E}{dt} + \mu_0 i_{enc} \quad (7)$$

3.2. Features

63

Given an image of a paper, we need to compute a num-

Outline

- writing technical papers
- giving technical talks



Monday, May 2, 2011

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.



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

- 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.



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

- 2-5 minute talks
- 20 -30 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.

Homework assignment

- For the 4 minute talk you'll give next Weds, 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
- 20 -30 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 another talk...



Layer the talk

In general, during any set of technical talks, the audience is bored and tired. Few are paying careful attention.

You want to give the talk at several different layers simultaneously. In some places, you want to give the technical details, for those few people who might actually follow them. This talk at a technical level gives a “peek under the hood” to reassure people that there is, indeed, an engine there.

For the other people, you want to give a running high-level summary of what you’re talking about, so they can follow along even though they’re not getting the details. These also serve an organizational function, like section headings in a paper. “So, we’ve derived the update equations for the variational Bayes algorithm. Now let’s see what form those take for our deblurring problem.”

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.

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.

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

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.

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

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

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.

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

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

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

You tell the story at several different levels of detail

The main idea

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

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

Then more details or
equations fleshing that
next part out,

Ways to engage the audience

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?

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.

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.

demo

Ted Adelson

Ted Adelson

- “people like to see a good fight”

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:

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.erantis.com/events/denmark/aarhus/billeder/tallshipsrace-skibe-i-havn-728.jpg>

Monday, May 2, 2011



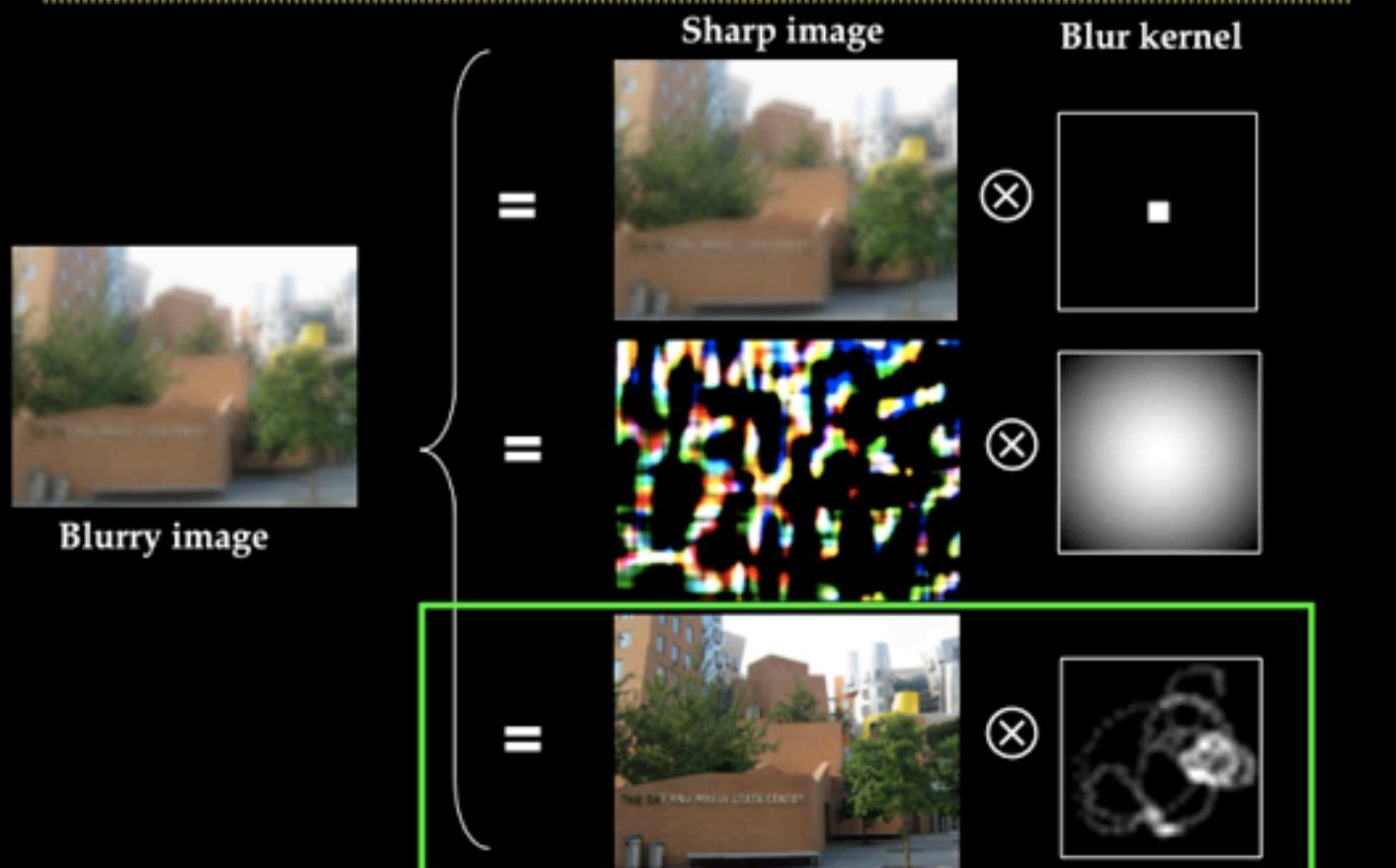
[http://www.flickr.com/photos/mnsomero/
2738807250/](http://www.flickr.com/photos/mnsomero/2738807250/)

Add dynamics to the talk



- 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



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.

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.