Removing Camera Shake from a Single Photograph
Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T. Roweis, and William T. Freeman
SIGGRAPH 2006

Single Image Motion Deblurring Using Transparency
Jiaya Jia
CVPR 2007
Motivation

Camera Motion
Removing Camera Shake from a Single Photograph

Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T. Roweis, and William T. Freeman

SIGGRAPH 2006
Camera Motion: Hardware Solution

Commercial stabilization

Ben-Ezra and Nayar PAMI 2004
Camera Motion

\[ I = L \ast f + n \]

- \( I \) is the degraded image
- \( L \) is the latent (unblurred) image
- \( f \) can be Point Spread Function or motion Kernel
The Problem

• Many pairs of $L$ and $f$ can be convolved resulting in the same $I$!

• On the other hand, the distribution of gradients in different natural images are similar across very different images!

[Field D., “What is the goal of sensory coding?” Neural Computation 6, 559-601, 1994]
Gradients Distribution
Gradients Distribution
The parametric model is built with mixture of Gaussians.
Estimation of the Blur Kernel

\[ P = L_p \ast K + n \]

But the statistics we have access to are on the gradients so:

\[ L_p, P \rightarrow \nabla L_p, \nabla P \]

Likelihood term

Mixture of zero-mean Gaussians

Encourages sparsity and positiveness of the kernel
Estimation of the Blur Kernel

The straightforward MAP solution to the problem fails so they use a Variational Bayesian approach.

Algorithm Outline

• User input:
  – Manual selection of a patch rich in edge structure
  – User input of the maximum size of the blur kernel
  – Initial estimate of the kernel (horizontal vs. vertical)

• Estimation of the blur kernel with the variational Bayes

• Multi-scaling

• Deconvolve with standard deconvolution algorithm (Lucy-Richardson)
Results
Results
Results
Single Image Motion Deblurring Using Transparency

Jiaya Jia
CVPR 2007
Motivation

Object Motion
(and camera motion)
Object Motion: Hardware Solution

Raskar et al. SIGGRAPH 2006
What happens when a solid, non-noisy, 1-D object is blurred?

IDEA:
Information about the blur can be extracted from the transparencies!
1-D Object Blur

The convolution can be seen as applied to the alpha matte!

\[ \alpha^i = \alpha^o \otimes f \]

\[ f = [f_{-n}, \ldots, f_0, \ldots, f_n]^T \]

\[ f_j = \begin{cases} 
\alpha^i_{j+n} - \alpha^i_{j+n-1} & \text{for } -n < j \leq n \\
\alpha^i_0 & \text{for } j = -n 
\end{cases} \]
The bands can be shown to be a close upper bound for the filter (most other methods require user interaction).
In this case the shape is unknown so we need to calculate the alpha matte together with $f$!

$$P(f, \alpha^o | \alpha^i) \propto P(\alpha^i | f, \alpha^o) P(\alpha^o) P(f)$$

- **Likelihood term**
- **Two terms**:
  - Alpha matte is binary
  - Neighboring pixels belong to the same region (either object or non-object)
- **Uniform distribution**
2-D Object Blur

- Solution of the MAP problem provides both the un-blurred alpha matte *and* the blur filter
- Solution is found by iterative optimization:
  - Conjugate gradient
  - Refinement with Belief Propagation
2-D Object Blur
Natural Images
Natural Images

- For a blurred object on a partially occluded background the transparency map is unique.
- For patches on natural images what is background?
- They prove mathematically that the identification of background does not affect the filter estimation!
Implementation

- For moving objects: user-drawn strokes to determine foreground and background
- For camera motion: user-selected patches around boundaries
- Calculate the transparency map with standard algorithms
- Estimate blur filter and deconvolve with standard deconvolution algorithm (Lucy-Richardson)
Results
Results
Results
Results
Results
Summary

• Fergus et al.
  – Use the finding that gradients distribution across natural images is heavy-tailed
  – Variational Bayes results in increased robustness

• Jia
  – Transparency carries information about blur
  – Is the MAP approach robust enough?
That's all!
Backup: Convolution

For the 1-D case:

\[(f * g)(\tau) = \sum_{n} f(n) \cdot g(\tau - n)\]

Point Spread Function PSF:

http://en.wikipedia.org/wiki/Point_spread_function
Backup: Alpha Matte

Pictures from Chang et al. “A Bayesian Approach to Digital Matting”, CVPR 2001
Backup: Maximum A Posteriori

\[ \hat{\theta}_{\text{ML}}(x) = \arg \max_{\theta} f(x|\theta) \]

\[ \theta \sim \frac{f(x|\theta) g(\theta)}{\int_{\Omega} f(x|\theta') g(\theta') d\theta'} \]

\[ \hat{\theta}_{\text{MAP}}(x) = \arg \max_{\theta} \frac{f(x|\theta) g(\theta)}{\int_{\Omega} f(x|\theta') g(\theta') d\theta'} = \arg \max_{\theta} f(x|\theta) g(\theta) \]

Formulas images from Wikipedia at: http://en.wikipedia.org/wiki/Maximum_a_posteriori
Backup: Terms Explained

\[ P(f, \alpha^o | \alpha^i) \propto P(\alpha^i | f, \alpha^o) P(\alpha^o) P(f) \]

\[ P(\alpha^i | f, \alpha^o) = \prod_{x,y} N(\alpha^i_{x,y} - \sum_{i,j} \alpha^o_{x-i,y-j} f_{i,j}; \sigma_1, \mu_1) \]

\[ P(\alpha^o) = \prod_{x,y} \exp(-\lambda \alpha^o_{x,y} | 1 - \alpha^o_{x,y}|) \prod_{(x,y), (x',y')} N(\alpha^o_{x',y'} - \alpha^o_{x,y}; \sigma_2, \mu_2) \]

Constrains that the unblurred alpha matte is a two-tone image

Alpha difference between neighboring pixels (8-neighborhood?)