Fast Removal of Non-Uniform Camera Shake

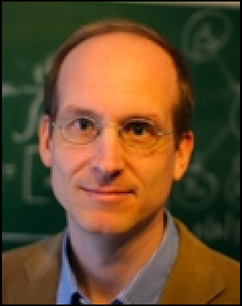
Michael Hirsch, Christian J. Schuler,
Stefan Harmeling, Bernhard Schölkopf

Max Planck Institute for Intelligent Systems, Tübingen, Germany

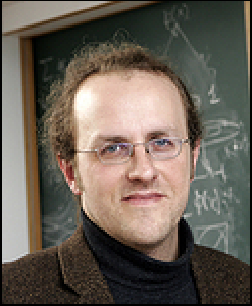
ICCV 2011 Barcelona

Joint work with ...

Christian J. Schuler



Stefan Harmeling



Bernhard Schölkopf

What is Camera Shake?

What problem do we want to solve?

Outline

1. Related Work
2. Fast Forward Model
3. Algorithm for Single Image Blind Deblurring
4. Results
5. Limitations
6. Conclusions

Where the story begins ...

- **Fergus et al. (SIGGRAPH'06): Removing camera shake from a single image**

Blurry Photo

Sharp Image

$$y \approx F \qquad x$$

Point Spread Function (PSF):

A fully describes the optical system. System response to a light point.

Where the story begins ...

- **Fergus et al. (SIGGRAPH'06): Removing camera shake from a single image**

Blurry Photo | PSF | Sharp Image

 $\approx$  $*$ 

$$y \approx f * x$$

Point Spread Function (PSF):

A fully describes the optical system. System response to a light point.

Common assumption:

$$F \neq F(s,t) \Rightarrow y = Fx = f * x$$

Where the story begins ...

- **Fergus et al. (SIGGRAPH'06): Removing camera shake from a single image**

Blurry Photo		PSF		Sharp Image
	$\approx$		$*$	
y	$\approx$	f	$*$	x

Task of Blind Deconvolution:

Given blurry image y recover x without knowing f.

Where the story begins ...

- **Fergus et al. (SIGGRAPH'06): Removing camera shake from a single image**

Blurry Photo | PSF | Sharp Image

$\approx$

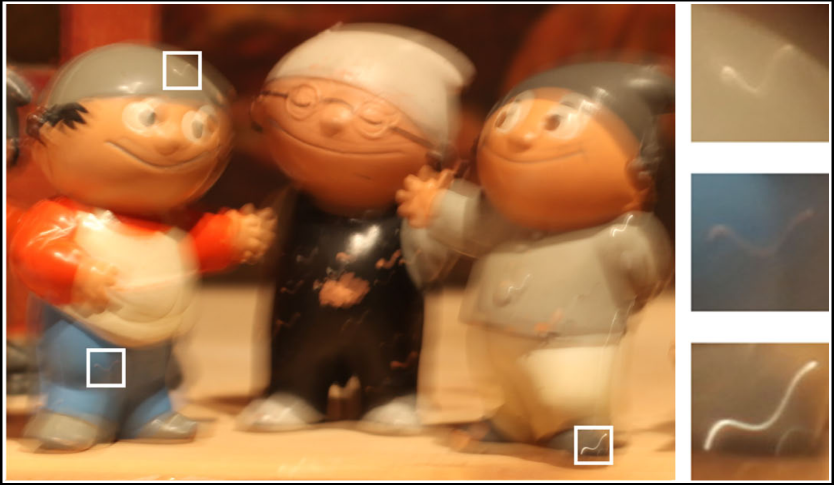
$*$

$$y \approx f * x$$

- **Cho and Lee (SIGGRAPH ASIA'09): Fast Motion Deblurring**
- **Xu and Jia (ECCV 2010): Two-Phase Kernel Estimation for Robust Motion Deblurring**
- **Levin et al. (CVPR 2009): Understanding and Evaluating Blind Deconvolution Algorithms**

What is Camera Shake?

Let's have a closer look...

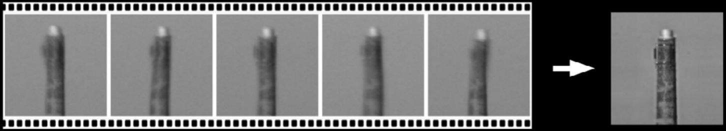


Spatially-Varying Convolution

- **Hirsch et al. (CVPR 2010): Efficient Filter Flow Framework**

$$y = Z_y^{\mathsf{T}} \sum_{r=0}^{p-1} C_r^{\mathsf{T}} F^{\mathsf{H}} \operatorname{Diag}(F Z_a a^{(r)}) F \operatorname{Diag}(w^{(r)}) C_r \, x,$$

- Multiframe Blind Deconvolution

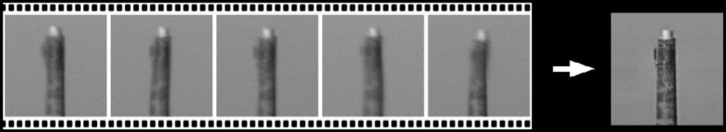


Spatially-Varying Convolution

- **Hirsch et al. (CVPR 2010): Efficient Filter Flow Framework**

$$y = Z_y^{\mathsf{T}} \sum_{r=0}^{p-1} C_r^{\mathsf{T}} F^{\mathsf{H}} \, \mathrm{Diag}(F Z_a a^{(r)}) F \, \mathrm{Diag}(w^{(r)}) C_r \, x,$$

- Multiframe Blind Deconvolution

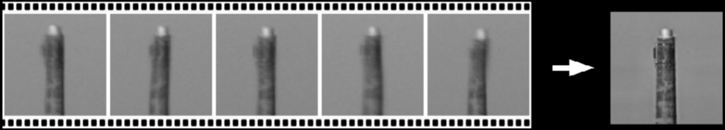


Spatially-Varying Convolution

- **Hirsch et al. (CVPR 2010): Efficient Filter Flow Framework**

$$y = Z_y^{\mathsf{T}} \sum_{r=0}^{p-1} C_r^{\mathsf{T}} F^{\mathsf{H}} \operatorname{Diag}(F Z_a a^{(r)}) F \operatorname{Diag}(w^{(r)}) C_r \, x,$$

- Multiframe Blind Deconvolution

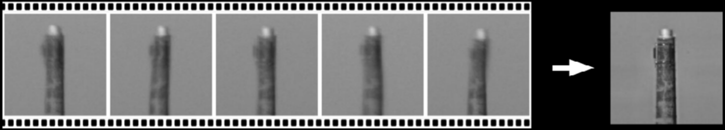


Spatially-Varying Convolution

- **Hirsch et al. (CVPR 2010): Efficient Filter Flow Framework**

$$y = Z_y^{\mathsf{T}} \sum_{r=0}^{p-1} C_r^{\mathsf{T}} F^{\mathsf{H}} \operatorname{Diag}(F Z_a a^{(r)}) F \operatorname{Diag}(w^{(r)}) C_r \, x,$$

- Multiframe Blind Deconvolution

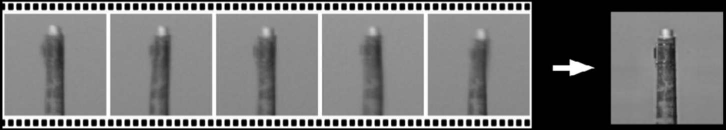


Spatially-Varying Convolution

- **Hirsch et al. (CVPR 2010): Efficient Filter Flow Framework**

$$y = Z_y^{\mathsf{T}} \sum_{r=0}^{p-1} C_r^{\mathsf{T}} F^{\mathsf{H}} \, \mathrm{Diag}(F Z_a a^{(r)}) F \, \mathrm{Diag}(w^{(r)}) C_r \, x,$$

- Multiframe Blind Deconvolution

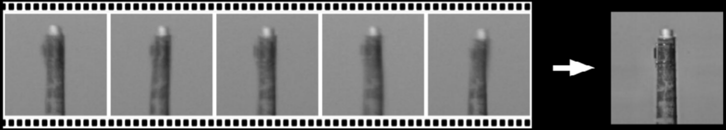


Spatially-Varying Convolution

- **Hirsch et al. (CVPR 2010): Efficient Filter Flow Framework**

$$y = Z_y^{\mathsf{T}} \sum_{r=0}^{p-1} C_r^{\mathsf{T}} F^{\mathsf{H}} \, \mathrm{Diag}(F Z_a a^{(r)}) F \, \mathrm{Diag}(w^{(r)}) C_r \, x,$$

- Multiframe Blind Deconvolution

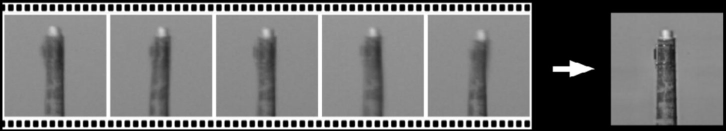


Spatially-Varying Convolution

- **Hirsch et al. (CVPR 2010): Efficient Filter Flow Framework**

$$y = \underbrace{Z_y^{\mathsf{T}} \sum_{r=0}^{p-1} C_r^{\mathsf{T}} F^{\mathsf{H}} \operatorname{Diag}(F Z_a a^{(r)}) F \operatorname{Diag}(w^{(r)}) C_r}_{A} \, x,$$

- Multiframe Blind Deconvolution

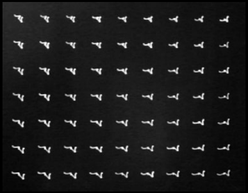


Spatially-Varying Convolution

- **Hirsch et al. (CVPR 2010): Efficient Filter Flow Framework**

$$y = Z_y^{\mathsf{T}} \sum_{r=0}^{p-1} C_r^{\mathsf{T}} F^{\mathsf{H}} \,\mathrm{Diag}(F Z_a a^{(r)}) F \,\mathrm{Diag}(w^{(r)}) C_r \, x,$$

- **Harmeling et al. (NIPS'10): Space-Variant Single-Image Blind Deconvolution for Removing Camera Shake**



hand shaked photo of grid

artificially blurred grid

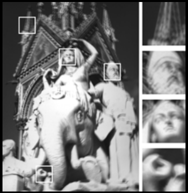
PSFs used for artificial blur

Spatially-Varying Convolution

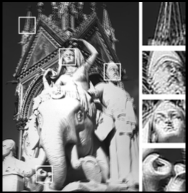
- **Hirsch et al. (CVPR 2010): Efficient Filter Flow Framework**

$$y = Z_y^{\mathsf{T}} \sum_{r=0}^{p-1} C_r^{\mathsf{T}} F^{\mathsf{H}} \operatorname{Diag}(F Z_a a^{(r)}) F \operatorname{Diag}(w^{(r)}) C_r \, x,$$

- **Harmeling et al. (NIPS'10): Space-Variant Single-Image Blind Deconvolution for Removing Camera Shake**



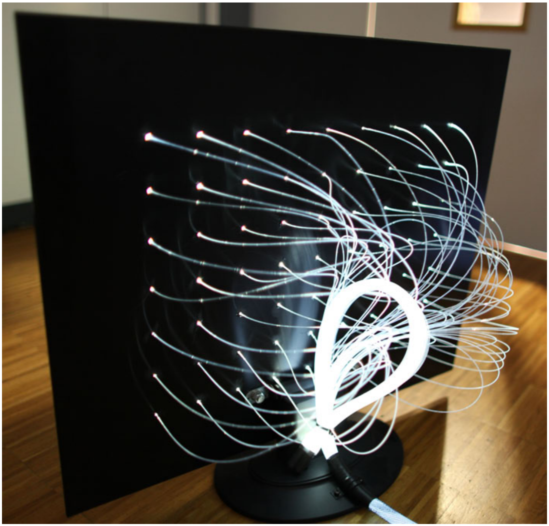
Blurred Image

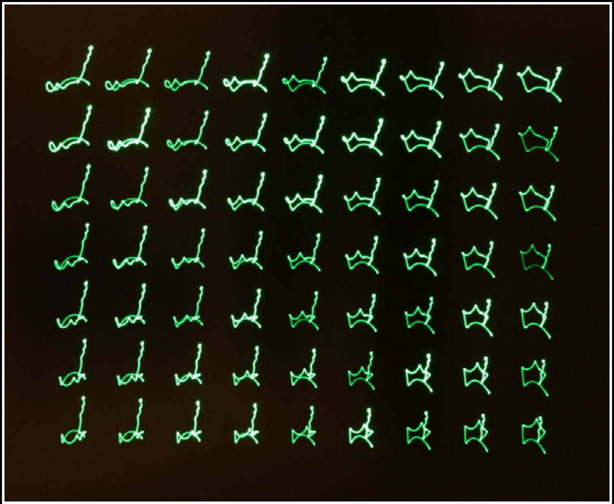


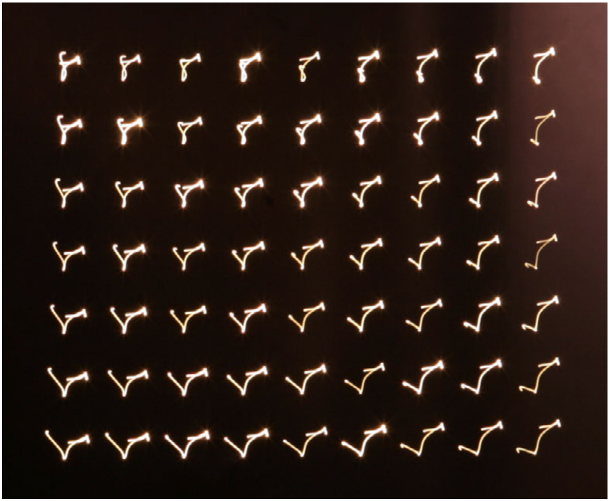
Cho and Lee

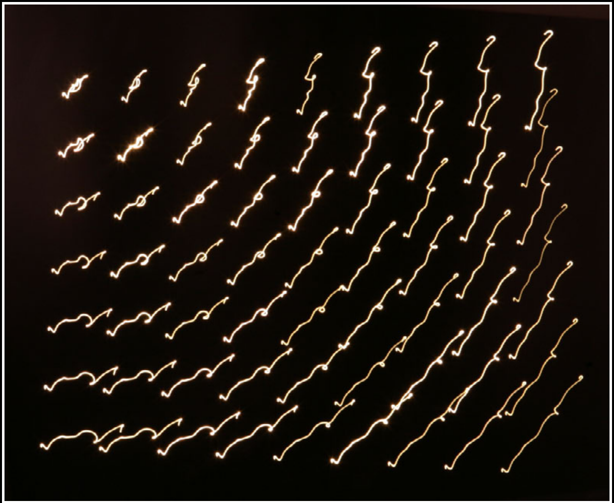
Harmeling et al.

What is Camera Shake really?



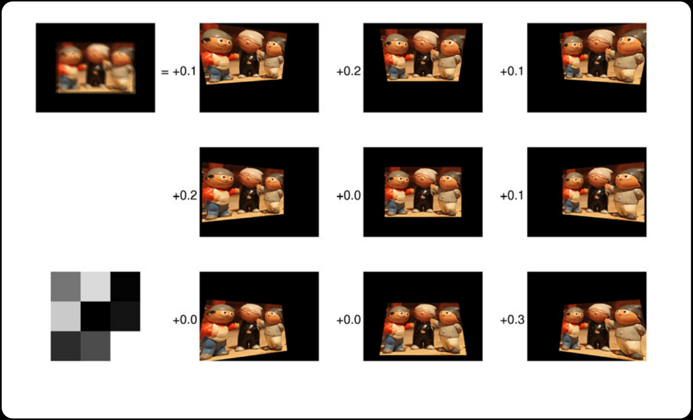






Projective Motion Path Blur Model

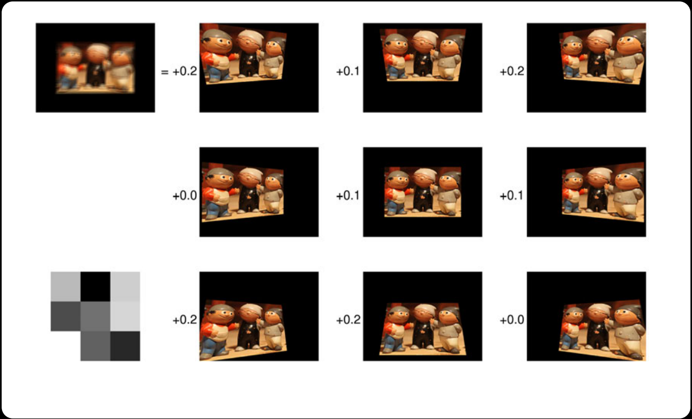
- **Tai et al. (Tech Report'09): Richardson-Lucy Deblurring for Scenes under Projective Motion Path**
- **Whyte et al. (CVPR'10): Non-uniform Deblurring for Shaken Images**
- **Gupta et al. (ECCV'10): Single Image Deblurring using Motion Density Functions**



Modified from Oliver Whyte's CVPR 2010 talk slides.

Projective Motion Path Blur Model

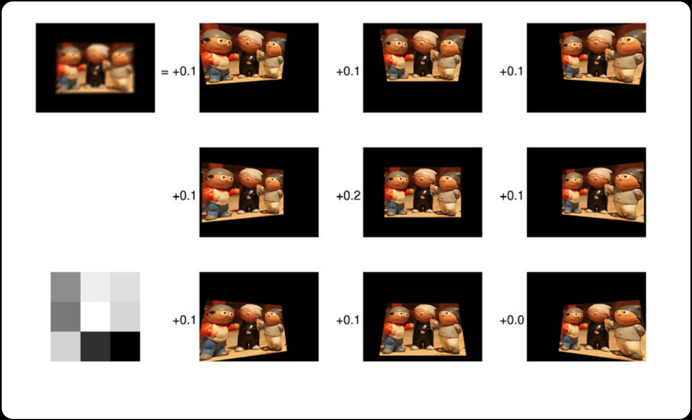
- Tai et al. (Tech Report'09): Richardson-Lucy Deblurring for Scenes under Projective Motion Path
- Whyte et al. (CVPR'10): Non-uniform Deblurring for Shaken Images
- Gupta et al. (ECCV'10): Single Image Deblurring using Motion Density Functions



Modified from Oliver Whyte's CVPR 2010 talk slides.

Projective Motion Path Blur Model

- Tai et al. (Tech Report'09): Richardson-Lucy Deblurring for Scenes under Projective Motion Path
- Whyte et al. (CVPR'10): Non-uniform Deblurring for Shaken Images
- Gupta et al. (ECCV'10): Single Image Deblurring using Motion Density Functions

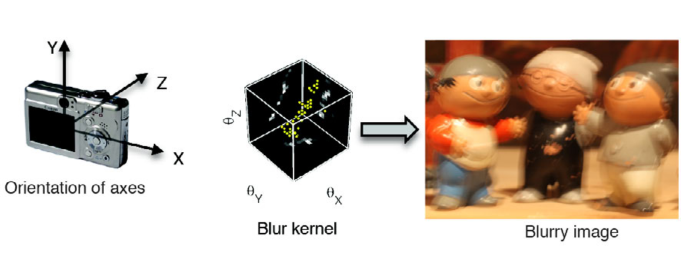


Modified from Oliver Whyte's CVPR 2010 talk slides.

Projective Motion Path Blur Model

- **Oliver Whyte et al. (CVPR 2010): Non-uniform Deblurring for Shaken Images**

Restrict allowed motion to rotations only and take a discrete set of camera orientations. Inference is performed on the weighting vector.



Modified from Oliver Whyte's CVPR 2010 talk slides using plot_nonuni_kernel.m.

Pros:

- Camera motion constraint is inherent in the imaging model, i.e. only physically plausible camera motions are allowed.

Cons:

- Computationally expensive, e.g. 1200x1600 pixels $\approx$ 180s.
- For comparison: our space-varying filtering $\approx$ 1.5s @CPU, 0.01s @GPU.

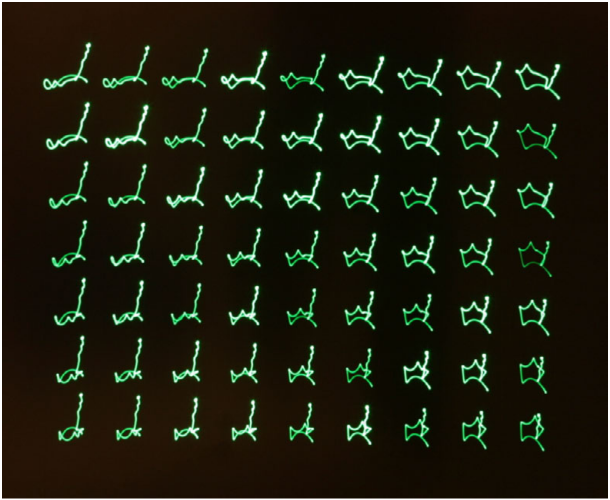
Question

How can we combine both approaches in a unified framework?

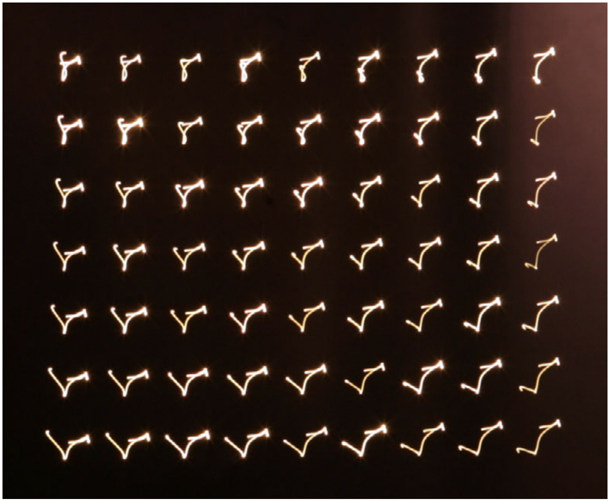
Forward Model

Camera Motion Constraint Efficient Filter Flow

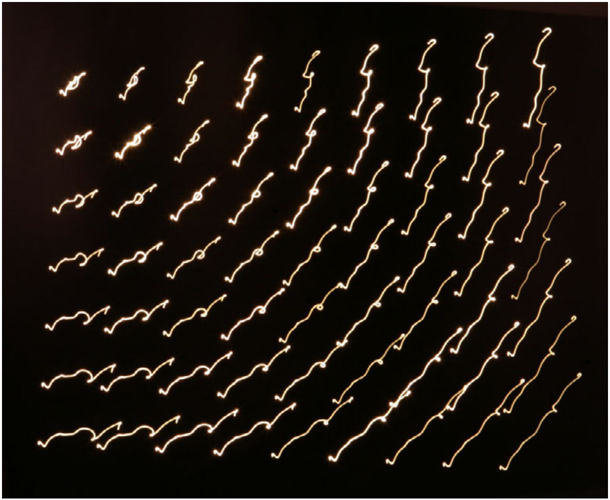
Inception - How the idea was born...



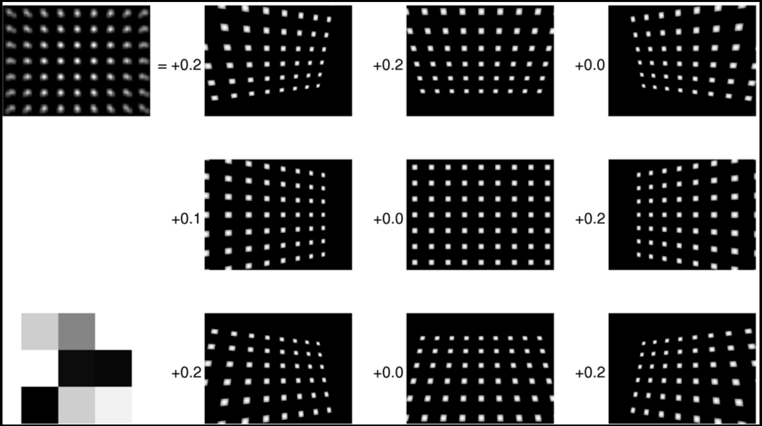
Inception - How the idea was born...



Inception - How the idea was born...

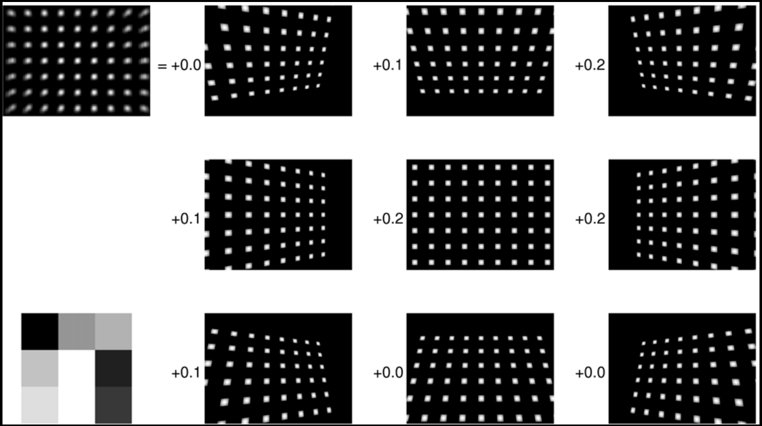


Looking how a Point Grid Transforms



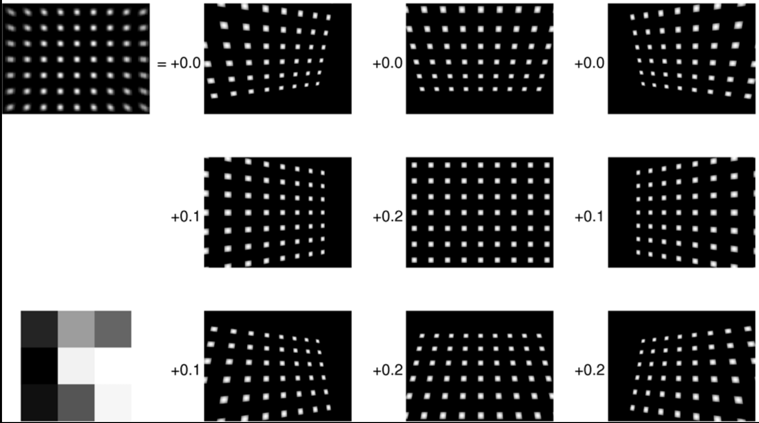
For EFF we need to know the PSF. As the PSF is the system's response to a point source, let's see how a grid of dots transforms.

Looking how a Point Grid Transforms



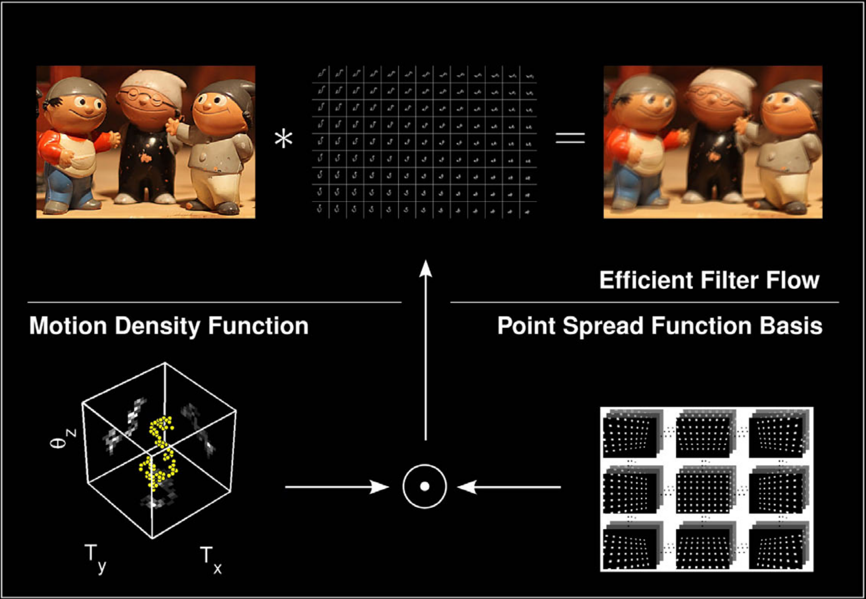
Transformed point grid depicts the response of the optical system to point sources, i.e. it encapsulates the space-varying PSF at abitrary positions.

Looking how a Point Grid Transforms

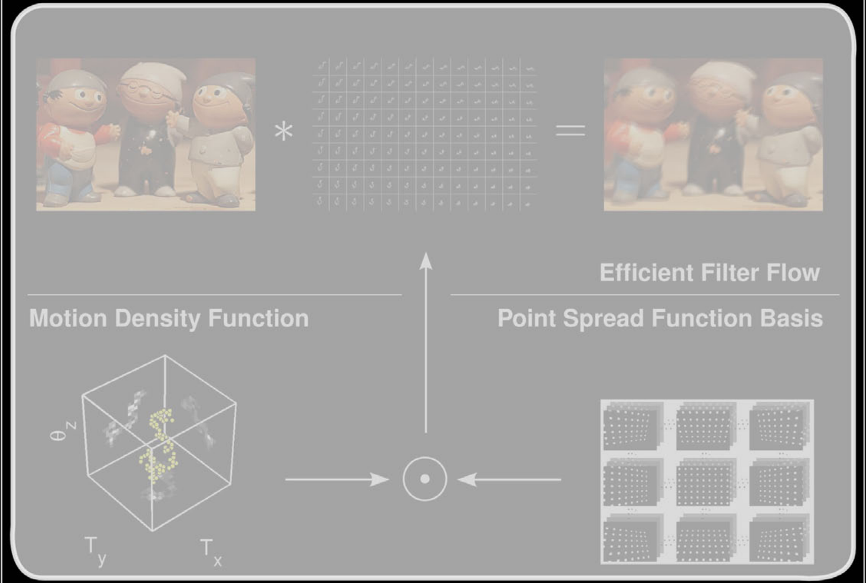


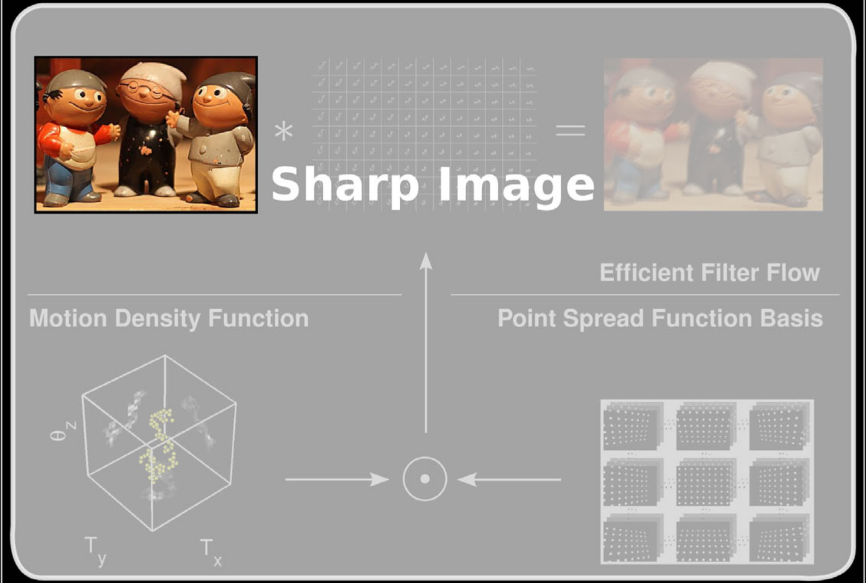
Transformed point grid depicts the response of the optical system to point sources, i.e. it encapsulates the space-varying PSF at abitrary positions.

Camera Motion Constraint Efficient Filter Flow



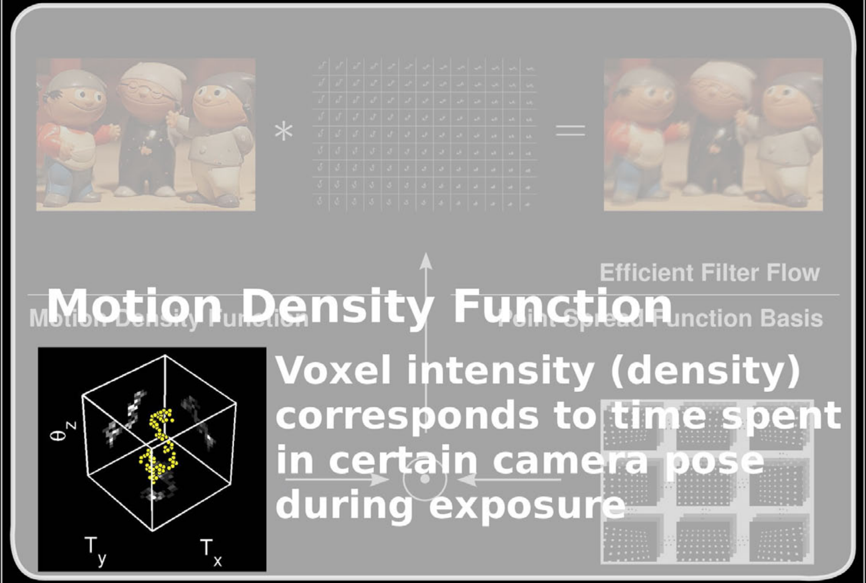
Camera Motion Constraint Efficient Filter Flow



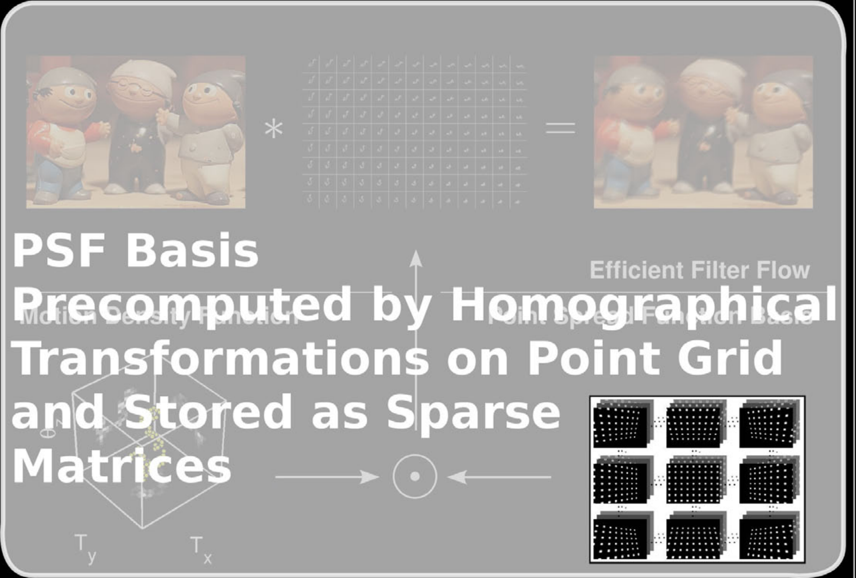

*
=
Sharp Image
Efficient Filter Flow
Motion Density Function
Point Spread Function Basis
θ_z
T_y
T_x

Motion Density Function

Voxel intensity (density) corresponds to time spent in certain camera pose during exposure

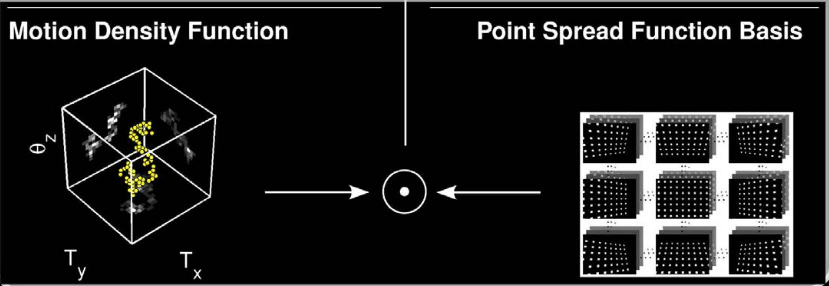


PSF Basis Precomputed by Homographical Transformations on Point Grid and Stored as Sparse Matrices

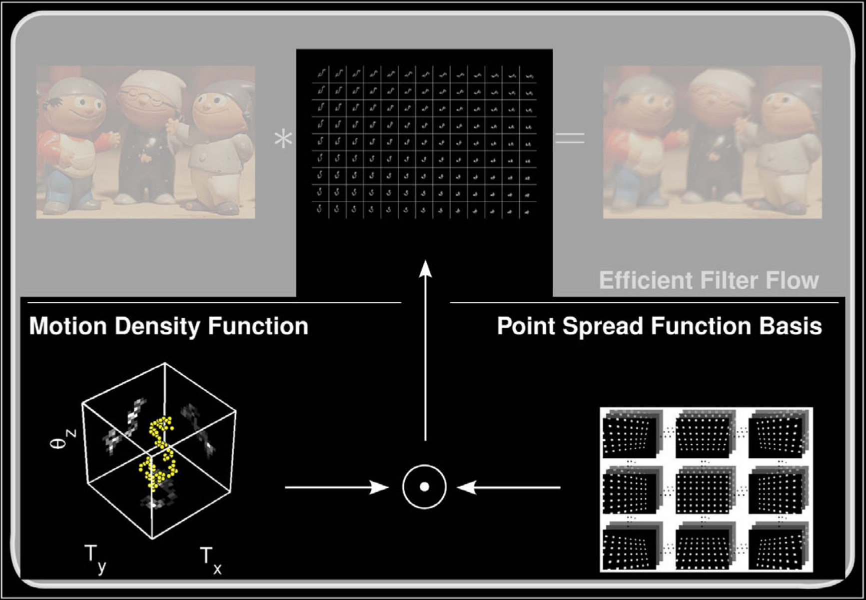


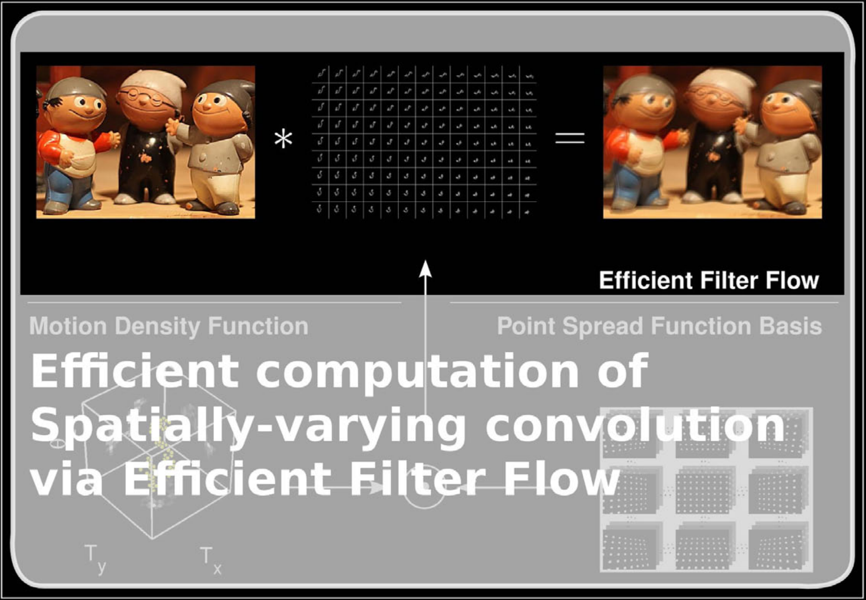
Computation of Non-uniform PSF as Weighted Sum of PSF Basis Frames via Sparse Matrix-Vector Multiplication

Efficient Filter Flow



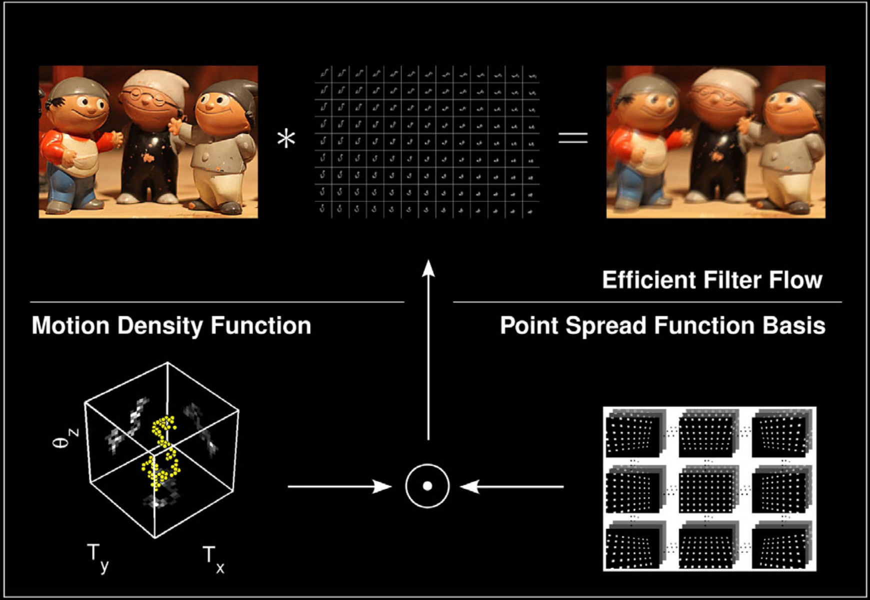
Camera Motion Constraint Efficient Filter Flow





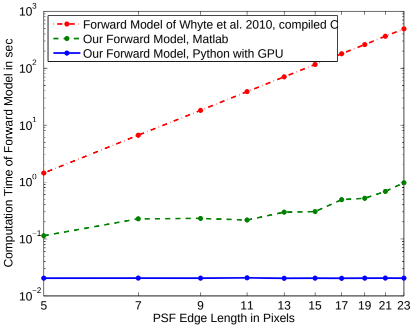
Efficient computation of Spatially-varying convolution via Efficient Filter Flow

Camera Motion Constraint Efficient Filter Flow

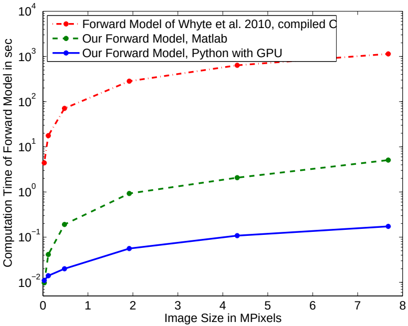


Discussion

Computation Time

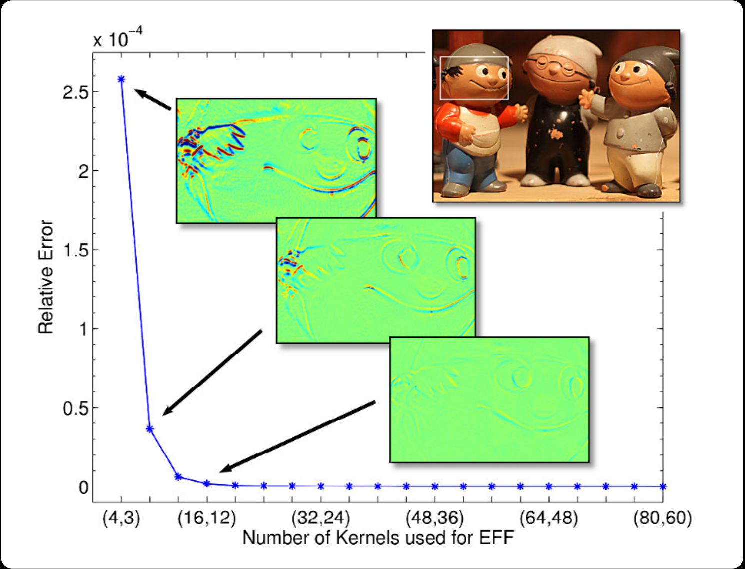


Dependence of PSF size

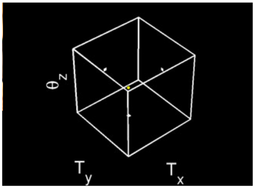


Dependence of image size

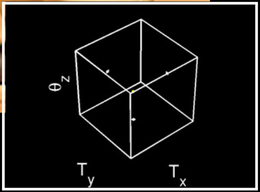
Accuracy


x 10^-4
Relative Error
2.5
2
1.5
1
0.5
0
(4,3)
(16,12)
(32,24)
(48,36)
(64,48)
(80,60)
Number of Kernels used for EFF

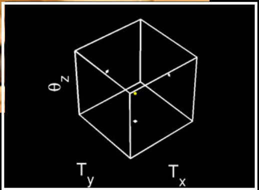
Expressivity



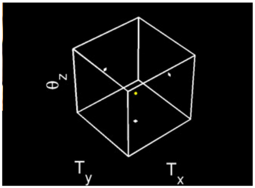
Vertical motion



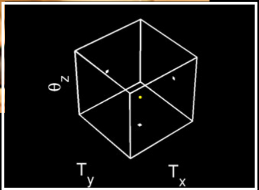
Vertical motion



Vertical motion

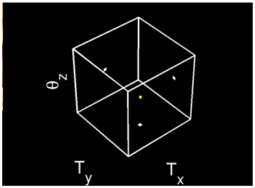


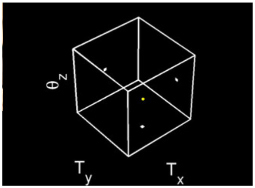
Vertical motion



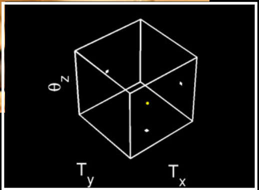
Vertical motion

Vertical motion

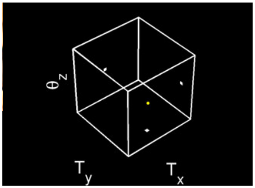




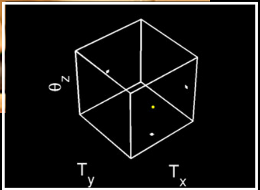
Vertical motion



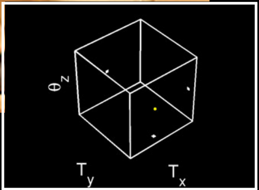
Vertical motion



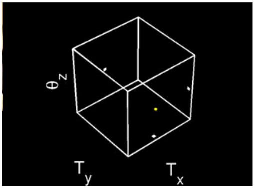
Vertical motion



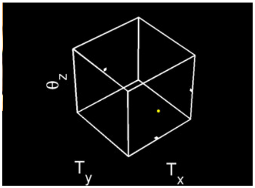
Vertical motion



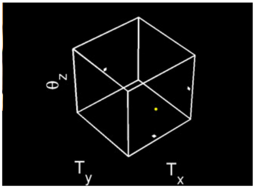
Vertical motion



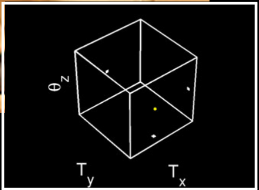
Vertical motion



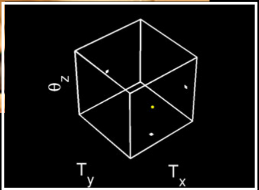
Vertical motion



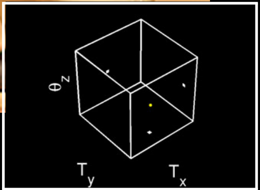
Vertical motion



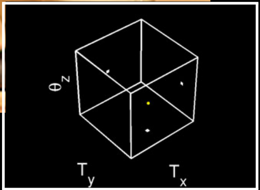
Vertical motion



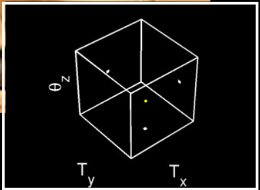
Vertical motion



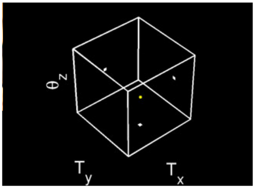
Vertical motion



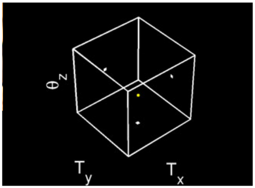
Vertical motion



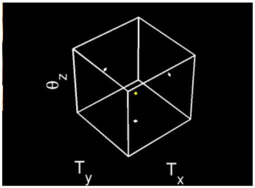
Vertical motion



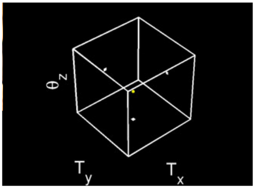
Vertical motion



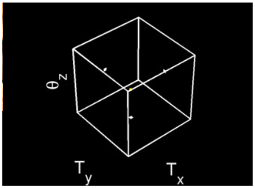
Vertical motion



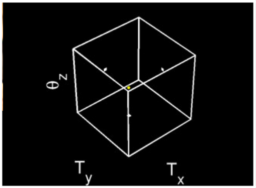
Vertical motion



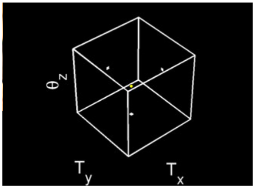
Vertical motion



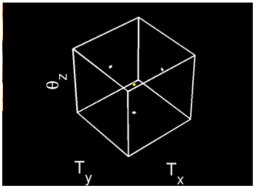
Vertical motion



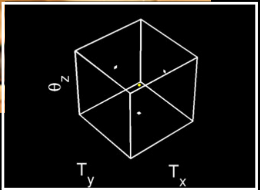
Horizontal motion



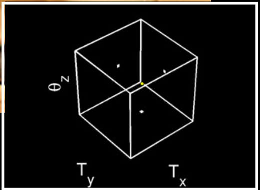
Horizontal motion



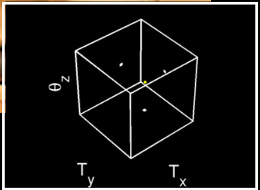
Horizontal motion



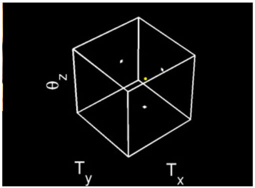
Horizontal motion



Horizontal motion

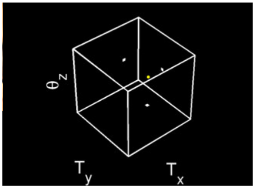


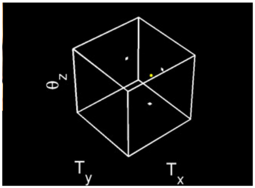
Horizontal motion



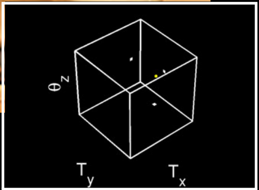
Horizontal motion

Horizontal motion

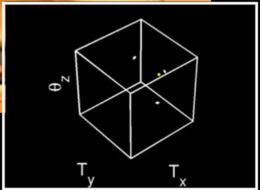




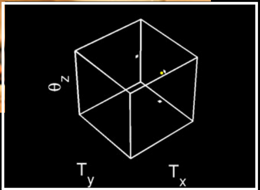
Horizontal motion



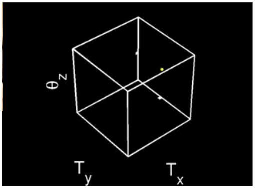
Horizontal motion



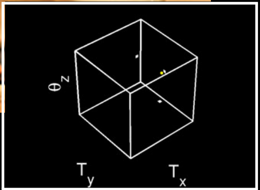
Horizontal motion



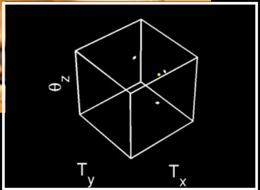
Horizontal motion



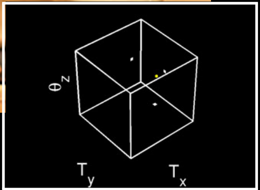
Horizontal motion



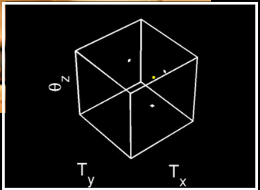
Horizontal motion



Horizontal motion

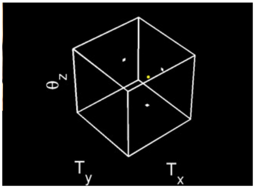


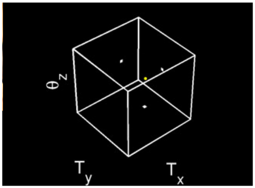
Horizontal motion



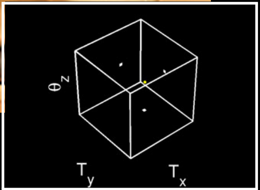
Horizontal motion

Horizontal motion

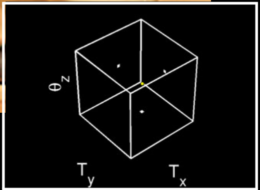




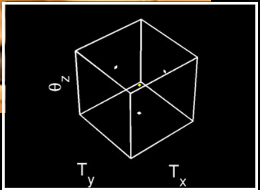
Horizontal motion



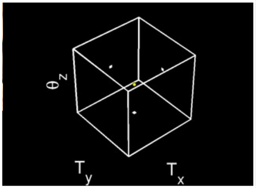
Horizontal motion



Horizontal motion

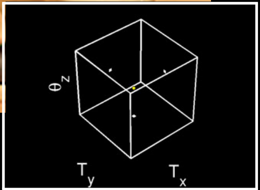


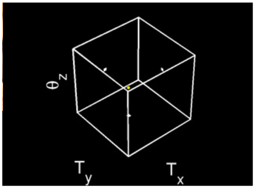
Horizontal motion



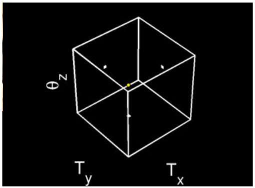
Horizontal motion

Horizontal motion

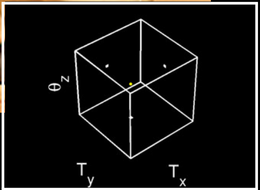




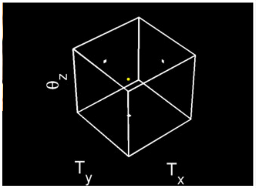
Rotational motion



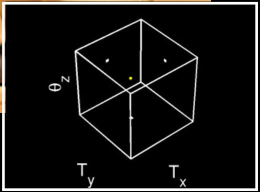
Rotational motion



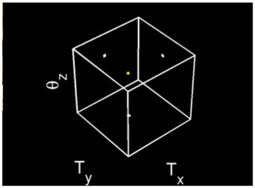
Rotational motion



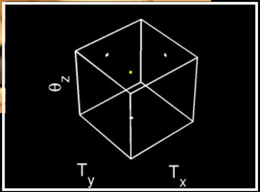
Rotational motion



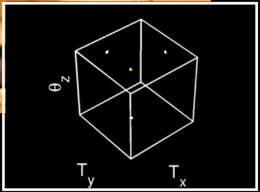
Rotational motion



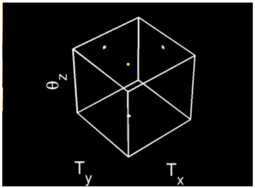
Rotational motion



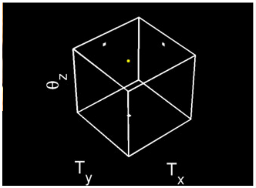
Rotational motion



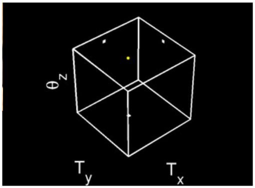
Rotational motion



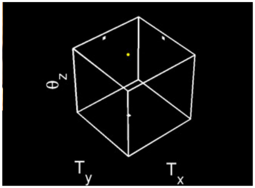
Rotational motion



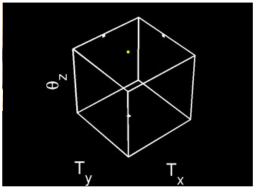
Rotational motion



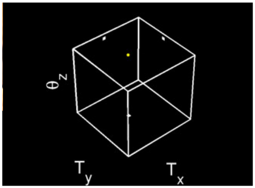
Rotational motion



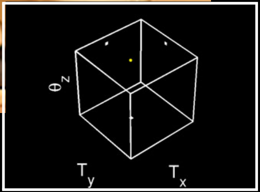
Rotational motion



Rotational motion

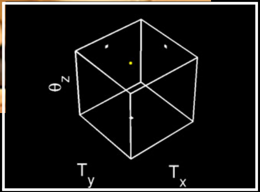


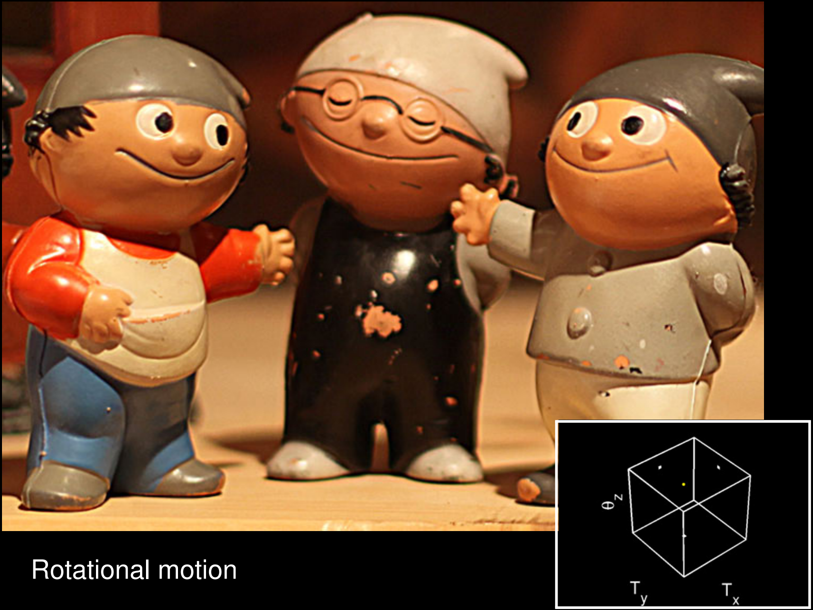
Rotational motion



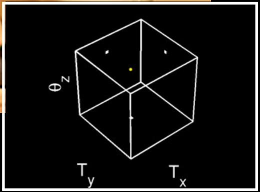
Rotational motion

Rotational motion

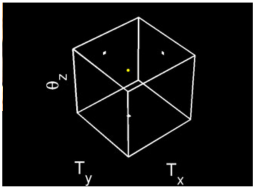




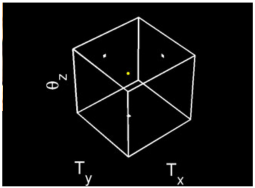
Rotational motion



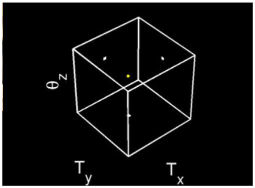
Rotational motion



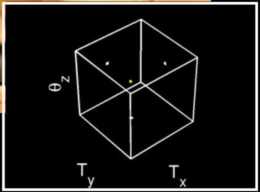
Rotational motion



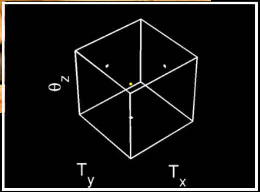
Rotational motion



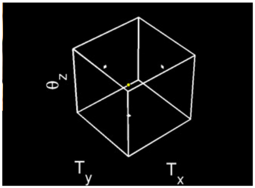
Rotational motion



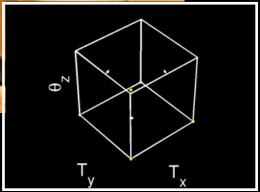
Rotational motion



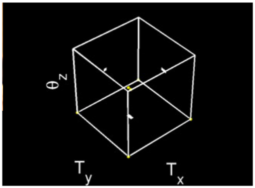
Rotational motion



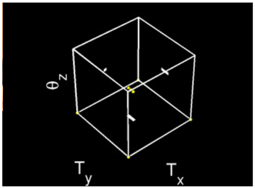
Rotational motion



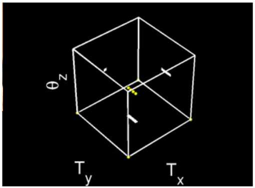
Vertical motion blur



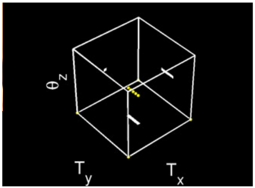
Vertical motion blur



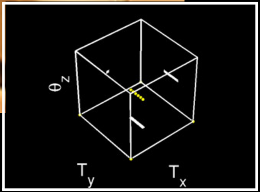
Vertical motion blur



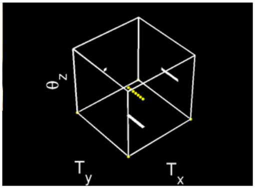
Vertical motion blur



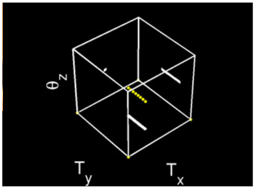
Vertical motion blur



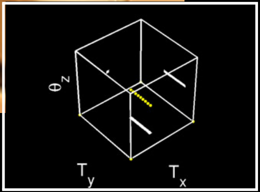
Vertical motion blur



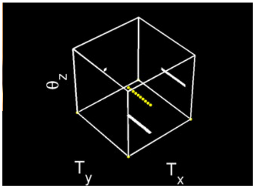
Vertical motion blur



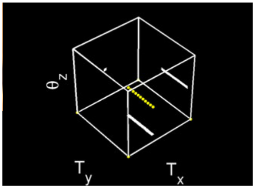
Vertical motion blur



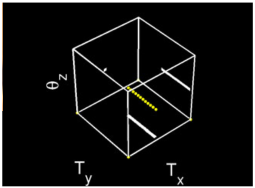
Vertical motion blur



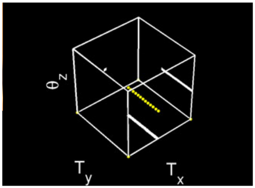
Vertical motion blur



Vertical motion blur

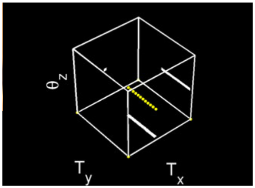


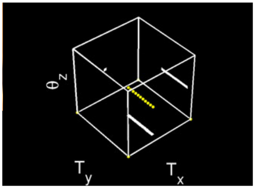
Vertical motion blur



Vertical motion blur

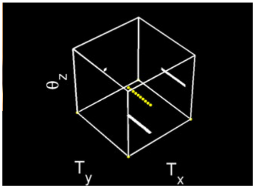
Vertical motion blur



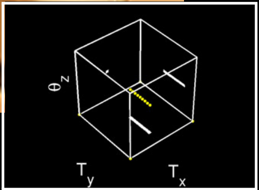


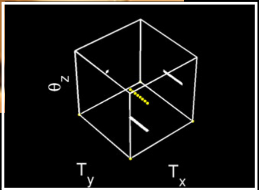
Vertical motion blur

Vertical motion blur



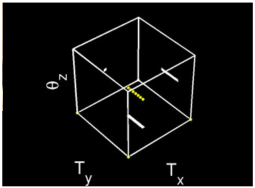
Vertical motion blur

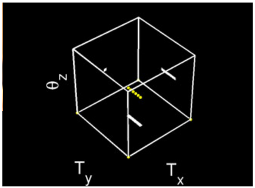




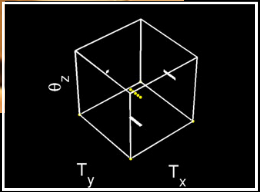
Vertical motion blur

Vertical motion blur

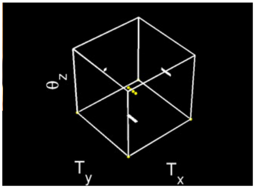




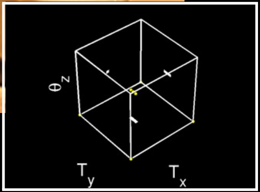
Vertical motion blur



Vertical motion blur

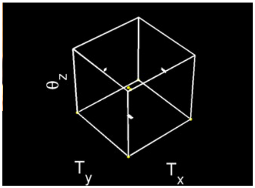


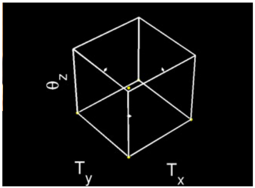
Vertical motion blur



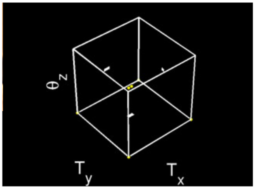
Vertical motion blur

Vertical motion blur

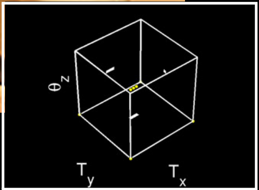




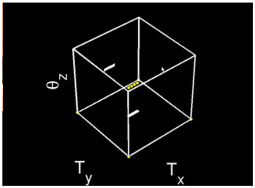
Horizontal motion blur



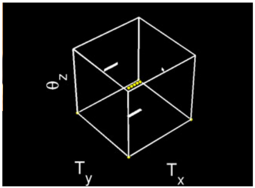
Horizontal motion blur



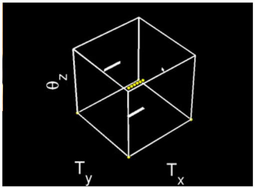
Horizontal motion blur



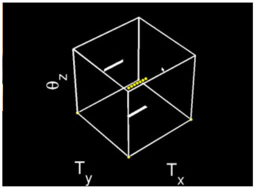
Horizontal motion blur



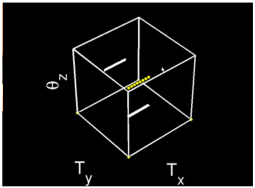
Horizontal motion blur



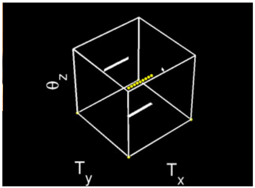
Horizontal motion blur



Horizontal motion blur

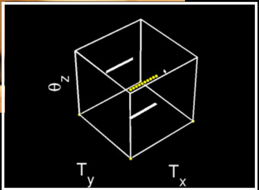


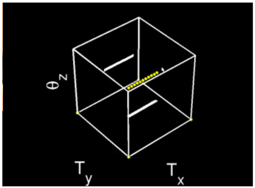
Horizontal motion blur



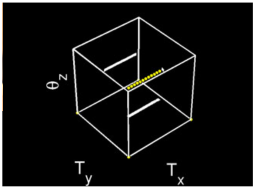
Horizontal motion blur

Horizontal motion blur

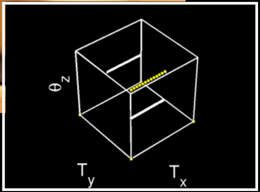




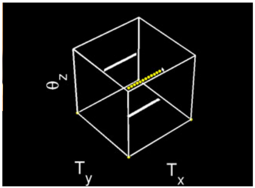
Horizontal motion blur



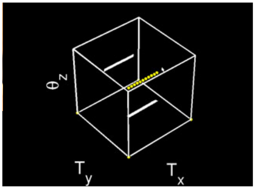
Horizontal motion blur



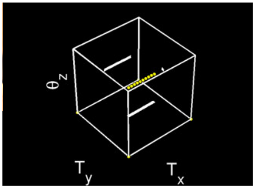
Horizontal motion blur



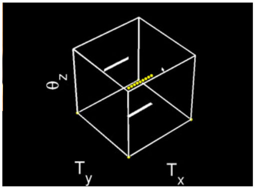
Horizontal motion blur



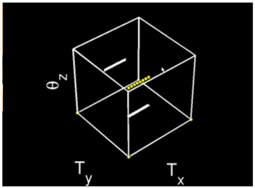
Horizontal motion blur



Horizontal motion blur

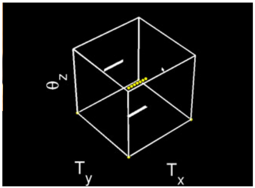


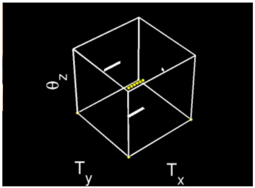
Horizontal motion blur



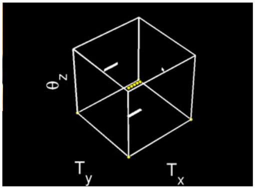
Horizontal motion blur

Horizontal motion blur

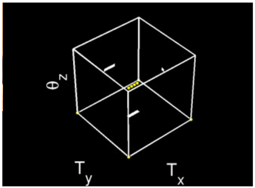




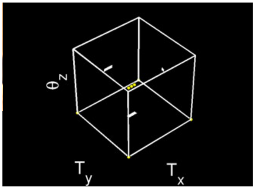
Horizontal motion blur



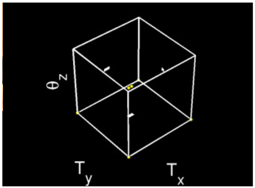
Horizontal motion blur



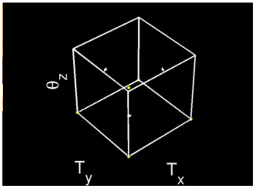
Horizontal motion blur



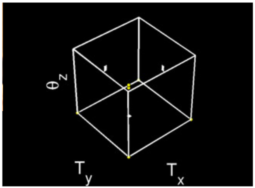
Horizontal motion blur



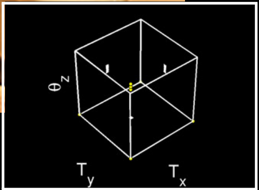
Horizontal motion blur



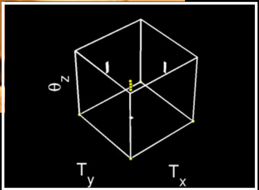
Rotational motion blur



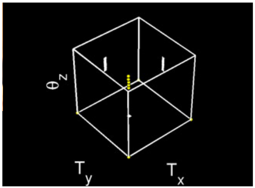
Rotational motion blur



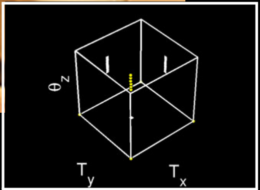
Rotational motion blur



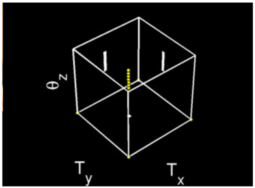
Rotational motion blur



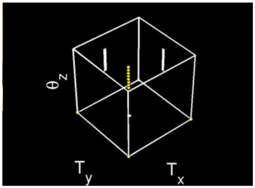
Rotational motion blur



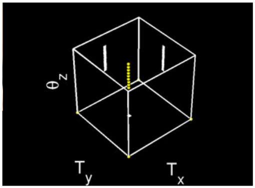
Rotational motion blur



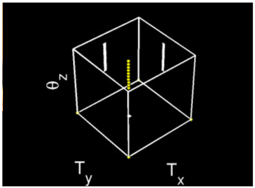
Rotational motion blur



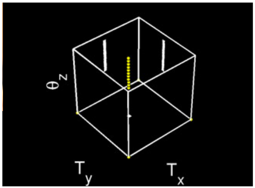
Rotational motion blur



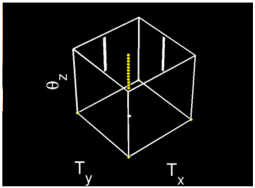
Rotational motion blur



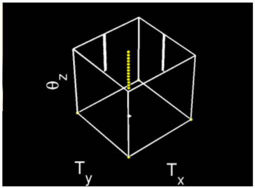
Rotational motion blur



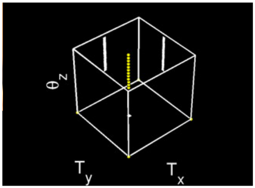
Rotational motion blur



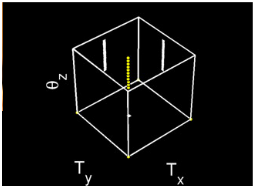
Rotational motion blur



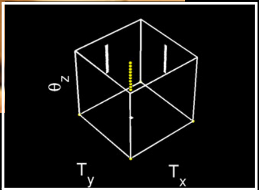
Rotational motion blur



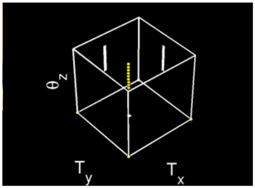
Rotational motion blur



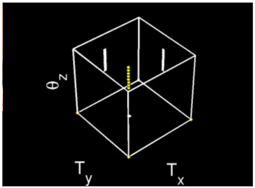
Rotational motion blur



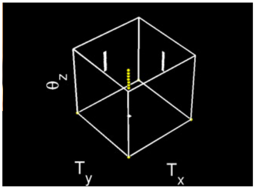
Rotational motion blur



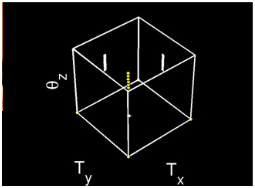
Rotational motion blur



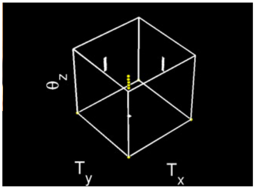
Rotational motion blur



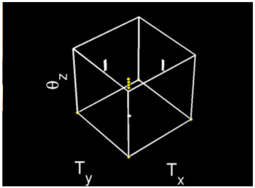
Rotational motion blur



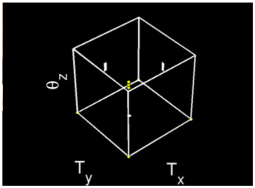
Rotational motion blur



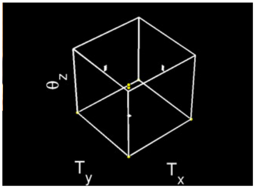
Rotational motion blur



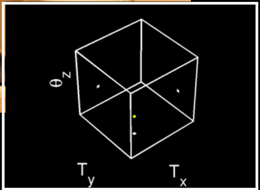
Rotational motion blur



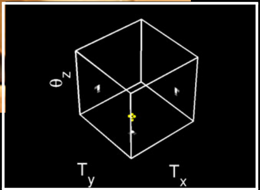
Rotational motion blur



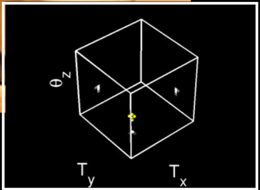
Rotational motion blur



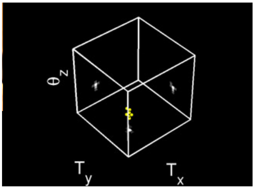
Camera motion blur



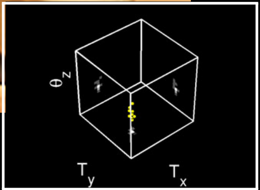
Camera motion blur



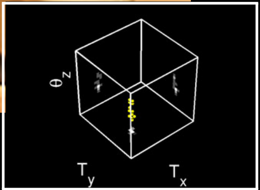
Camera motion blur



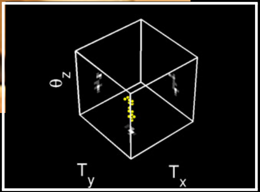
Camera motion blur



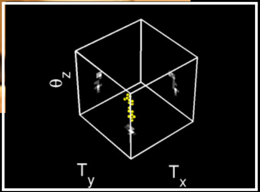
Camera motion blur



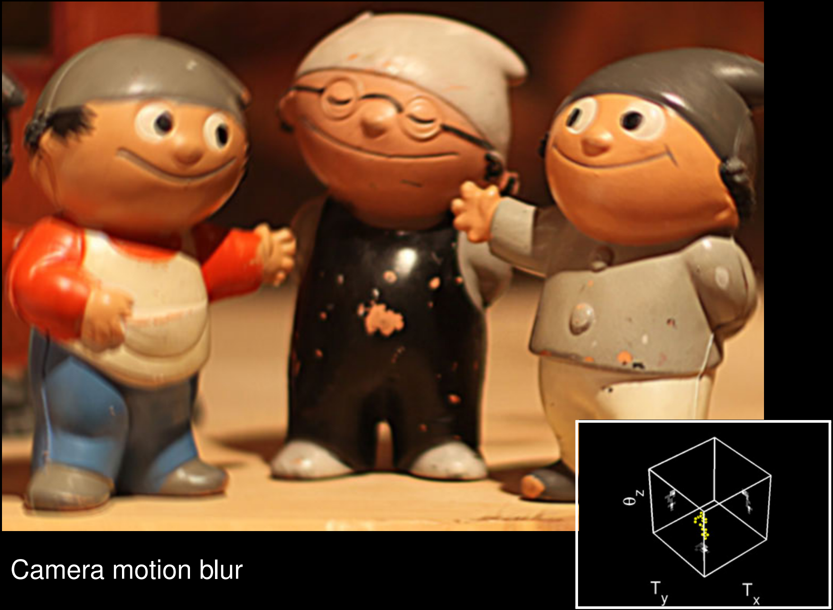
Camera motion blur



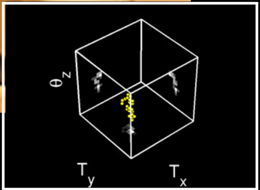
Camera motion blur



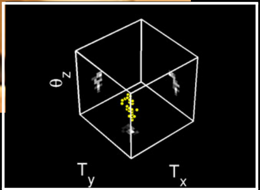
Camera motion blur



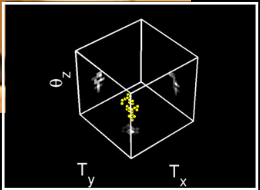
Camera motion blur



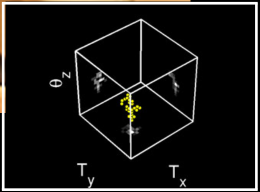
Camera motion blur



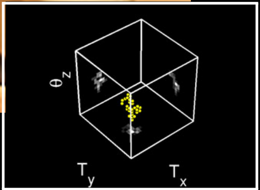
Camera motion blur



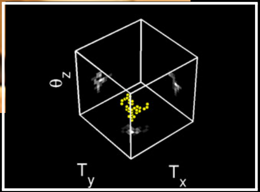
Camera motion blur



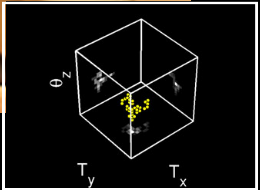
Camera motion blur



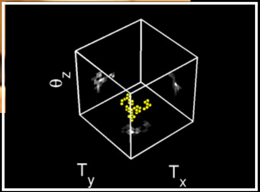
Camera motion blur



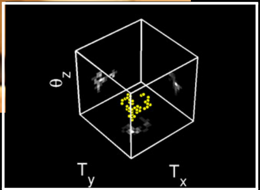
Camera motion blur



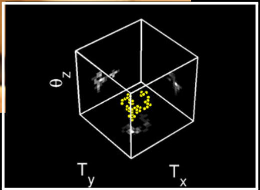
Camera motion blur



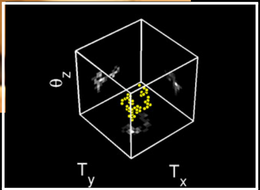
Camera motion blur



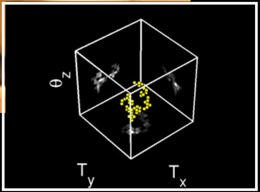
Camera motion blur



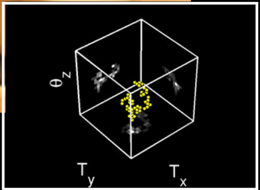
Camera motion blur



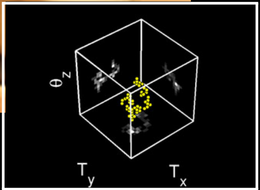
Camera motion blur



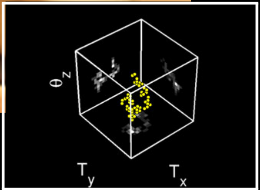
Camera motion blur



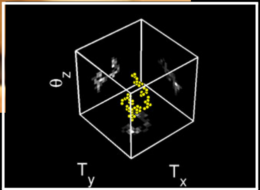
Camera motion blur



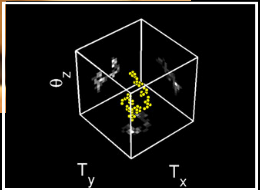
Camera motion blur



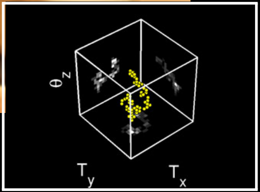
Camera motion blur



Camera motion blur

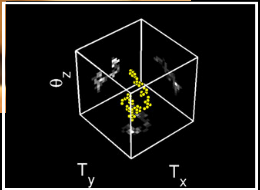


Camera motion blur

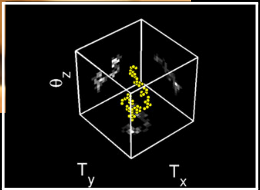


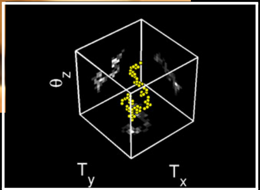
Camera motion blur

Camera motion blur



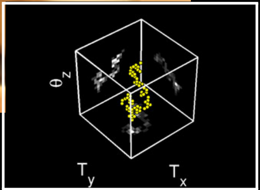
Camera motion blur

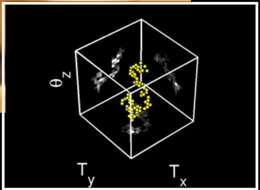




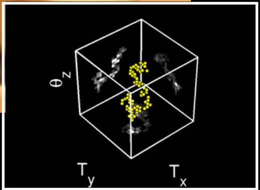
Camera motion blur

Camera motion blur

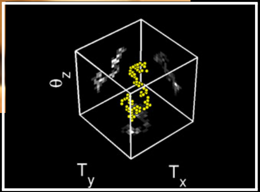




Camera motion blur



Camera motion blur



Camera motion blur

Single Image Blind Deblurring Algorithm

Overview of Single Image Blind Deblurring Algorithm

(i) **Step 1: Motion Blur Estimation:** We are iterating over four steps:

- **Prediction Step:** Apply shock and bilateral filtering to counter noise and emphasize edges (Cho and Lee (SIGGRAPH 2009))
- **Gradient Selection:** Select informative edges via gradient confidence map (rmap approach of Xu and Jia ECCV 2010)
- **Blur Parameter Estimation:** Update blur parameter by minimizing:

$$\min_{\mu} \left\|\partial g - m_S \odot \partial(\mu \diamond \tilde{f})\right\|_2^2 + \kappa \left\|\mu\right\|_2^2 + \lambda \left\|\partial \mu\right\|_2^2, \tag{1}$$

- **Latent Image Estimation:** Update latent image by minimizing:

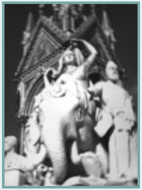
$$\min_{f} \left\|g - \mu \diamond f\right\|_2^2 + \nu \left\|\partial f\right\|_2^2 \tag{2}$$

(ii) **Step 2: Latent Image Estimation:**
Given the motion blur estimate we estimate the final deblurred image following Krishnan and Fergus (NIPS 2009):

$$\min_{f} \left\|g - \mu \diamond f\right\|_2^2 + \eta \left\|f\right\|_{2/3}^{2/3} \tag{3}$$

Step 1: Motion Blur Estimation

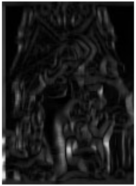
STEP 1: Blur Motion Estimation



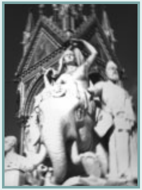
Blurred Image

Input

STEP 1: Blur Motion Estimation



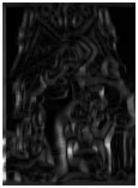
Rmap ⟵



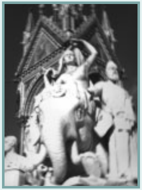
Blurred Image

SCALE 1 / ITERATION 1

STEP 1: Blur Motion Estimation



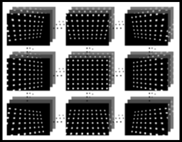
Rmap



Blurred Image

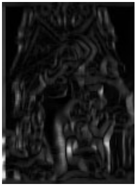
SCALE 1 / ITERATION 1

PSF Basis

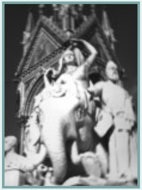


Step 1 : Blur Motion Estimation

Scale 1 / Iteration 1



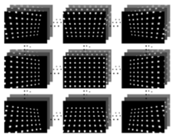
Rmap



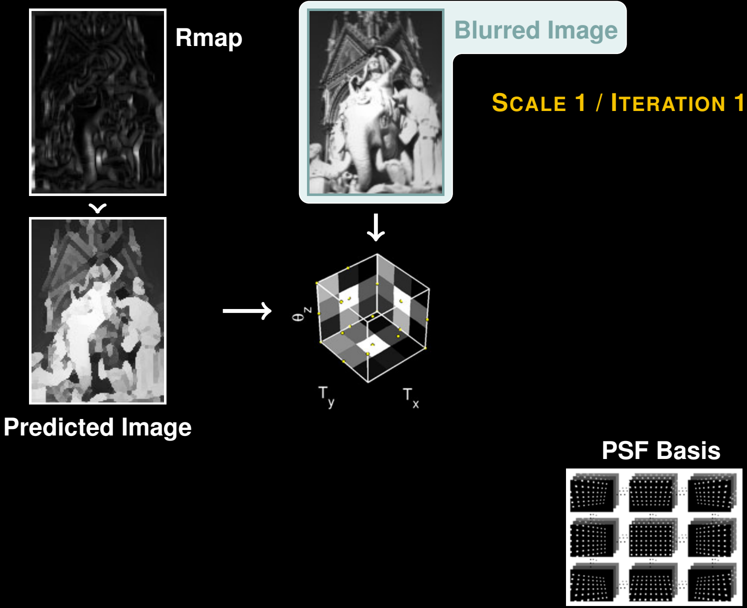
Blurred Image

Predicted Image

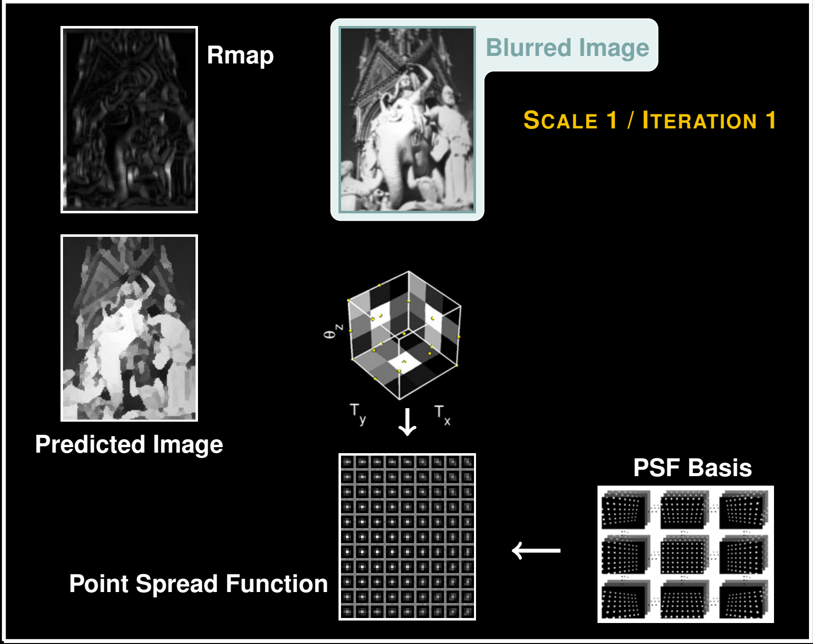
PSF Basis



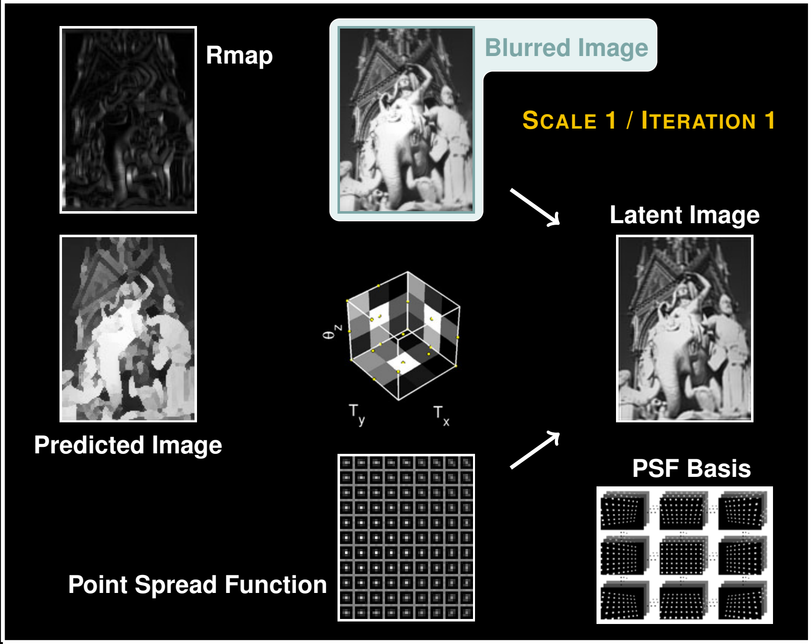
Step 1: Blur Motion Estimation



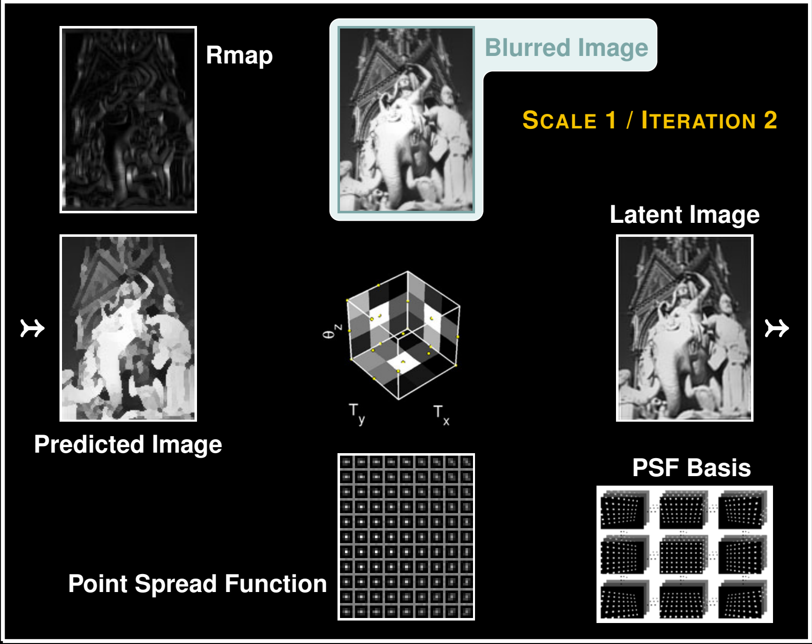
Step 1: Blur Motion Estimation



Step 1: Blur Motion Estimation

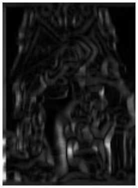


Step 1 : Blur Motion Estimation



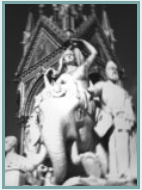
Step 1: Blur Motion Estimation

Scale 1 / Iteration 2

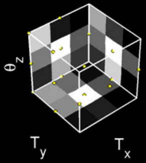


Rmap

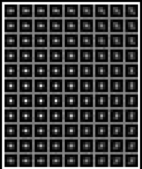
Predicted Image



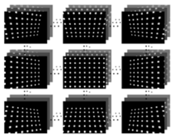
Blurred Image



Latent Image

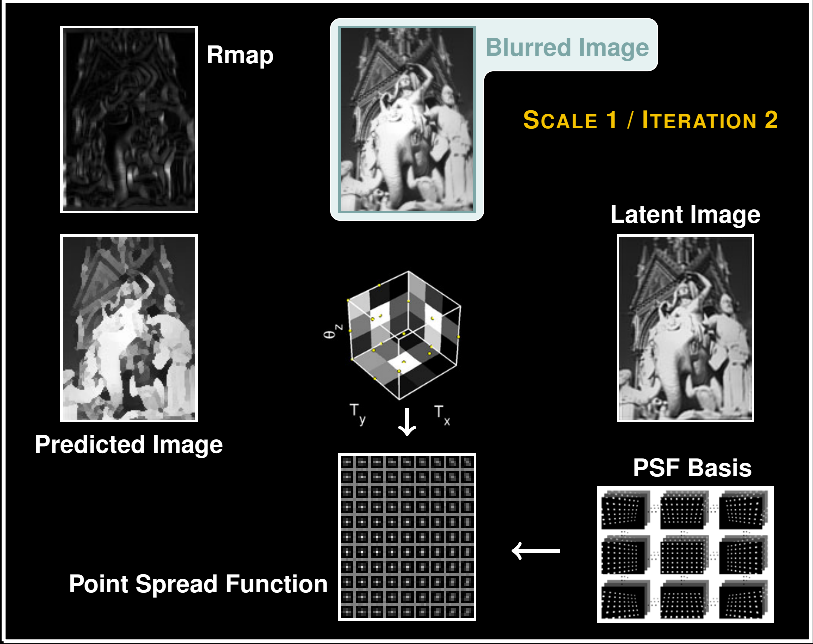


Point Spread Function



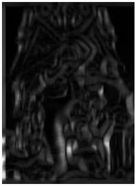
PSF Basis

Step 1: Blur Motion Estimation

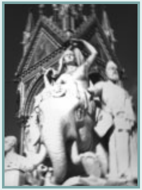


Step 1 : Blur Motion Estimation

Scale 1 / Iteration 2

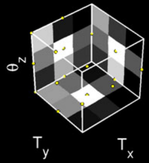


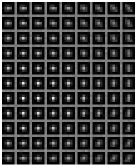
Rmap



Blurred Image

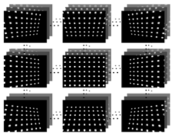
Predicted Image





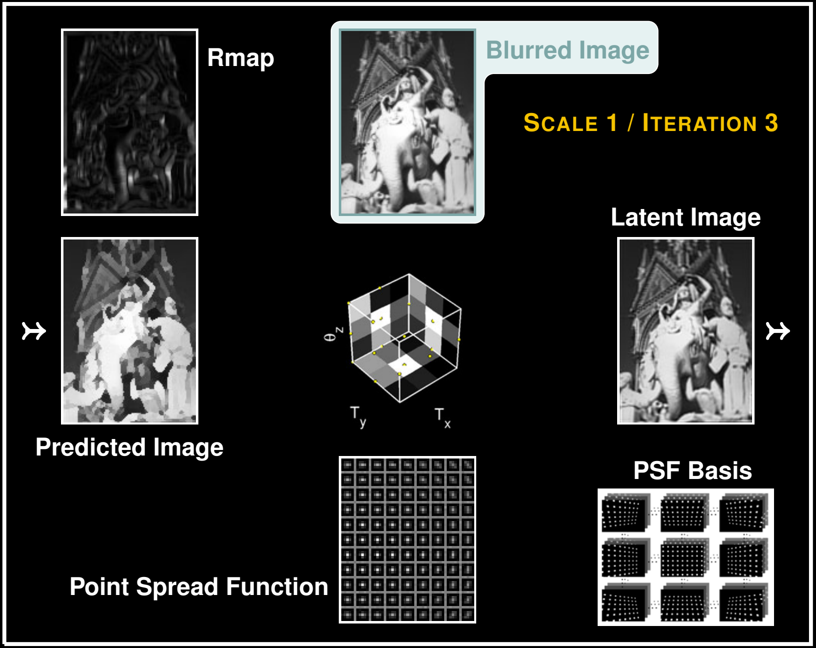
Point Spread Function

Latent Image

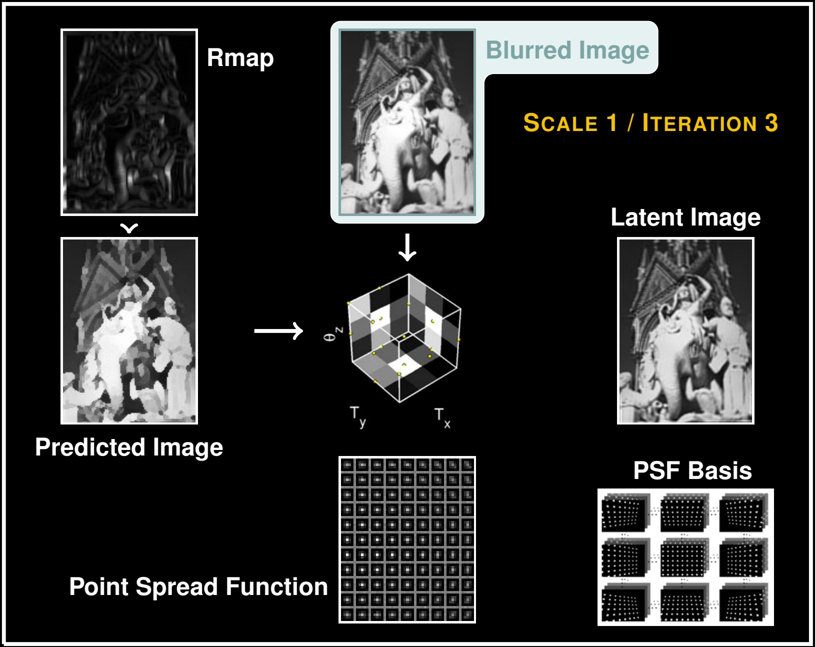


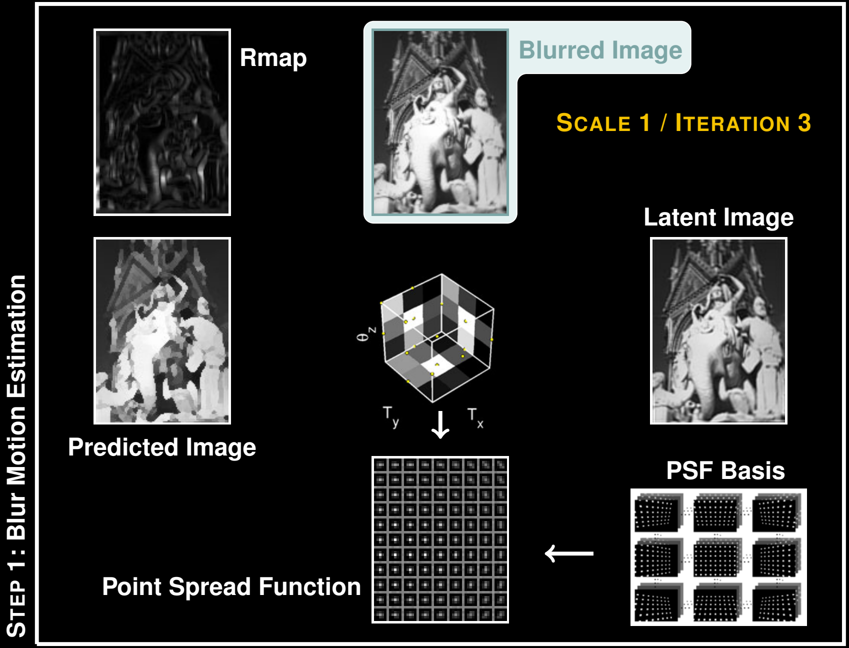
PSF Basis

Step 1: Blur Motion Estimation

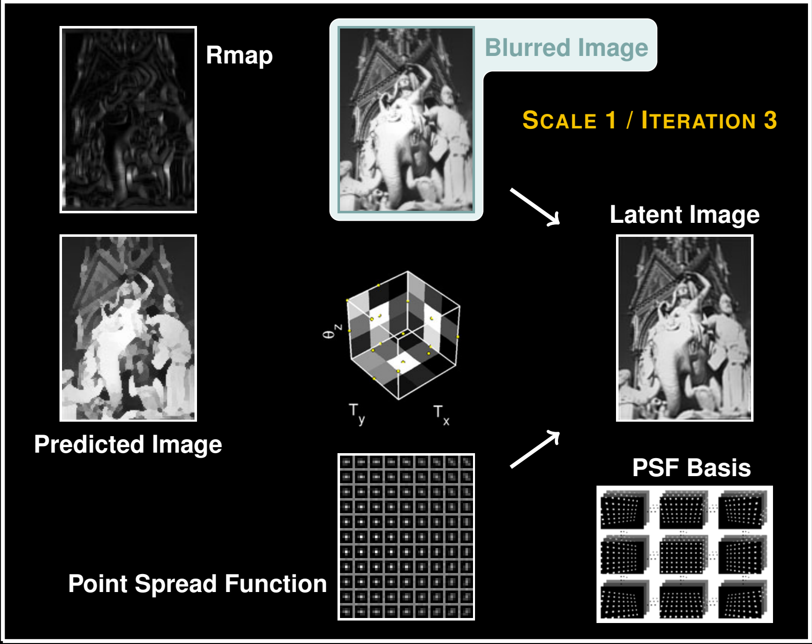


Step 1: Blur Motion Estimation



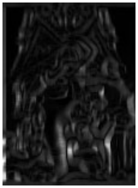

STEP 1: Blur Motion Estimation
Rmap
Blurred Image
SCALE 1 / ITERATION 3
Latent Image
Predicted Image
θ_z
T_y
T_x
PSF Basis
Point Spread Function

Step 1: Blur Motion Estimation

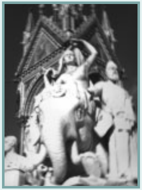


STEP 1: Blur Motion Estimation

Rmap

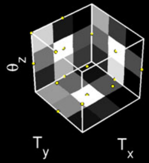


Blurred Image

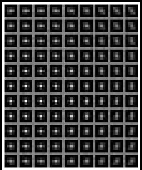


SCALE 1 / ITERATION 4

Predicted Image

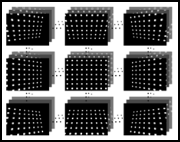


Latent Image

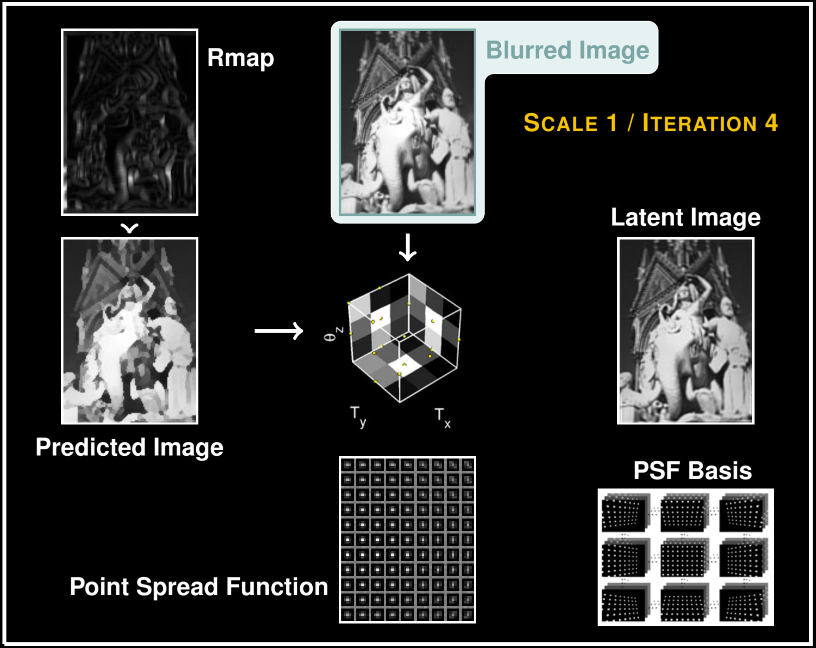


Point Spread Function

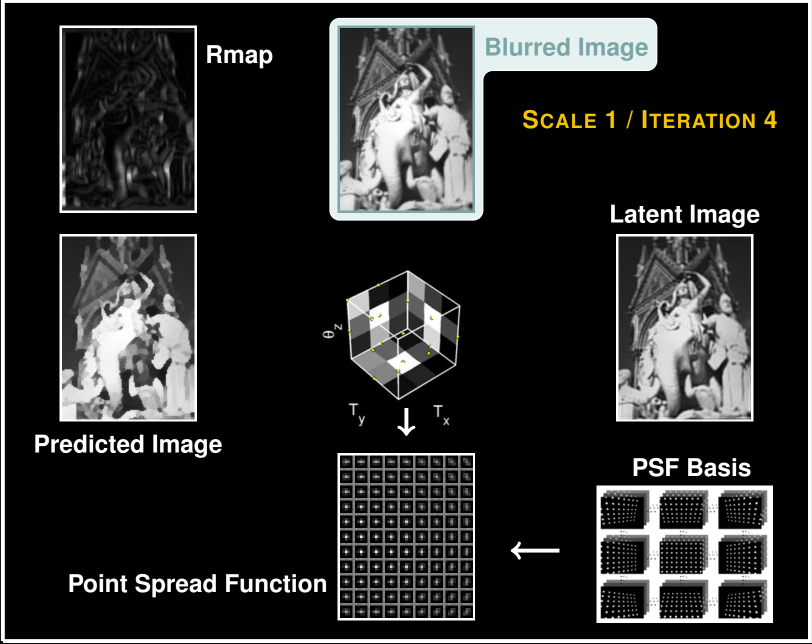
PSF Basis



STEP 1: Blur Motion Estimation

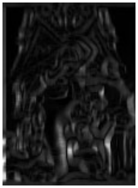


Step 1: Blur Motion Estimation

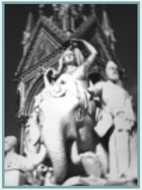


Step 1: Blur Motion Estimation

Scale 1 / Iteration 4

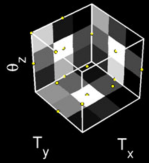


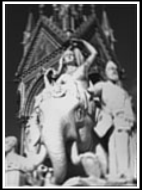
Rmap



Blurred Image

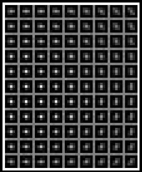
Predicted Image



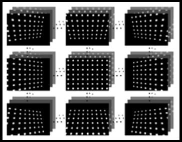


Latent Image

Point Spread Function

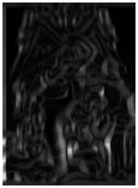


PSF Basis

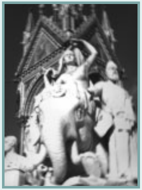


Step 1: Blur Motion Estimation

Scale 1 / Iteration 5

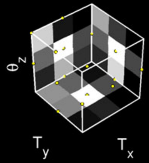


Rmap

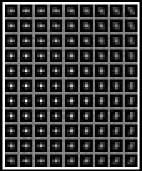


Blurred Image

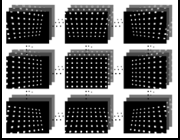
Predicted Image



Latent Image

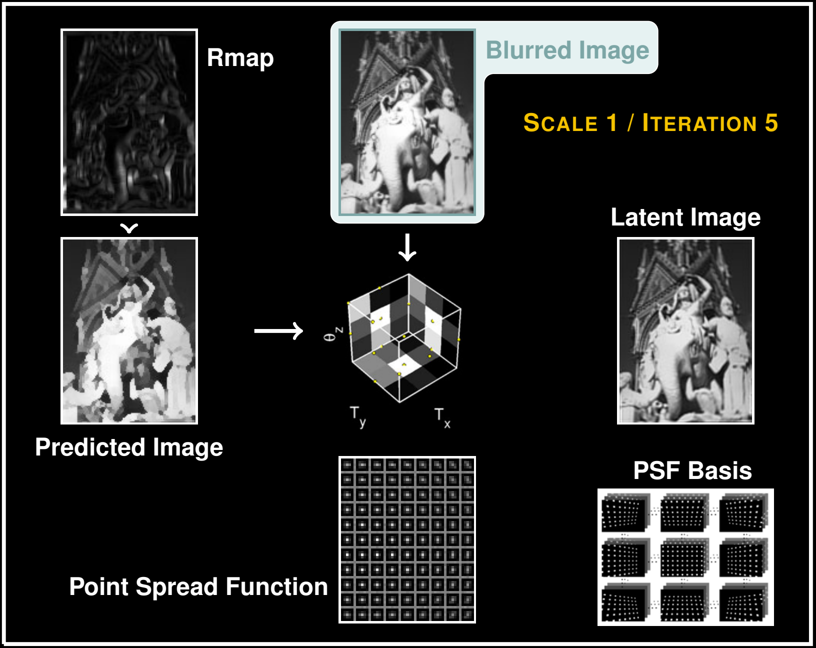


Point Spread Function

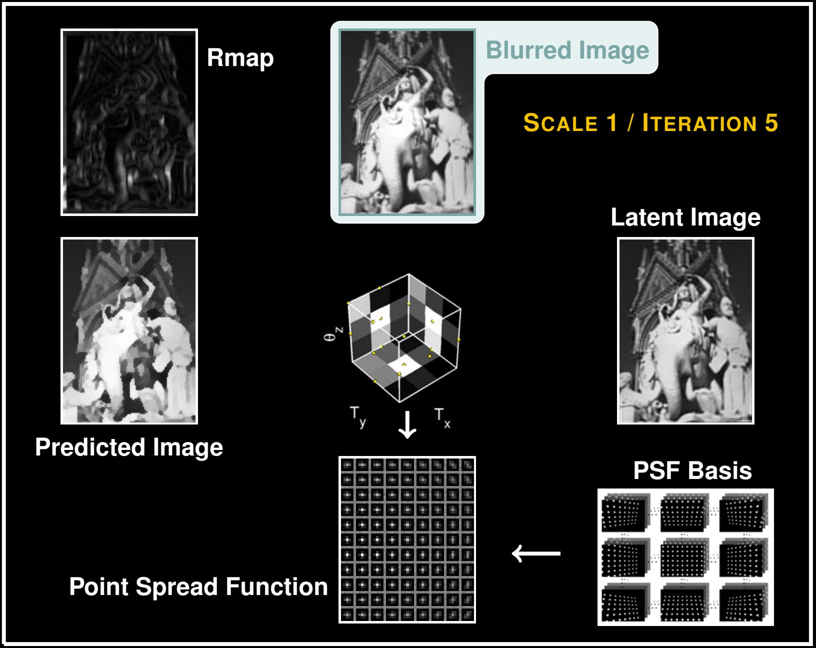


PSF Basis

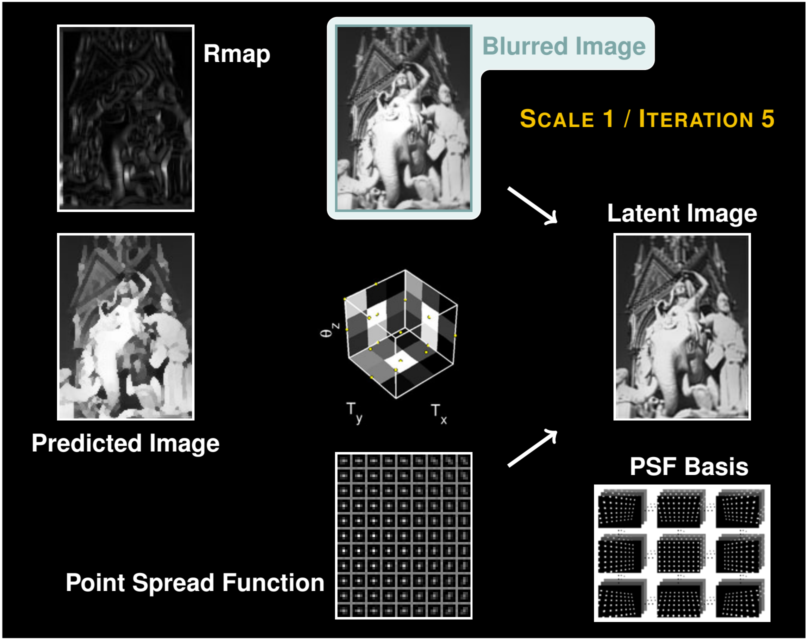
Step 1: Blur Motion Estimation



STEP 1: Blur Motion Estimation



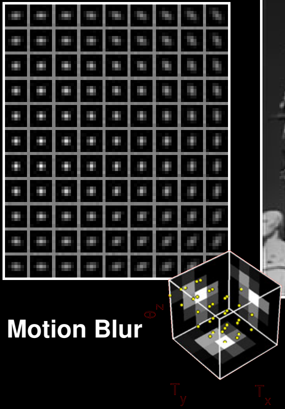
Step 1 : Blur Motion Estimation



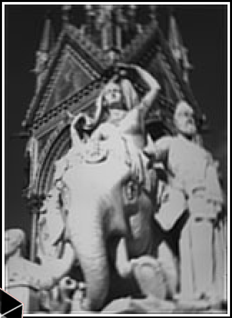
Step 1: Blur Motion Estimation

Scale 2 / Iteration 1

Predicted Image



Motion Blur

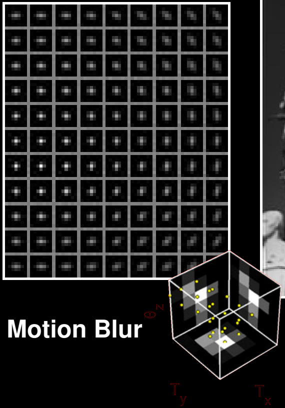


Latent Image

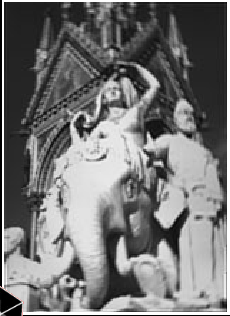
STEP 1: Blur Motion Estimation

SCALE 2 / ITERATION 2

Predicted Image



Motion Blur

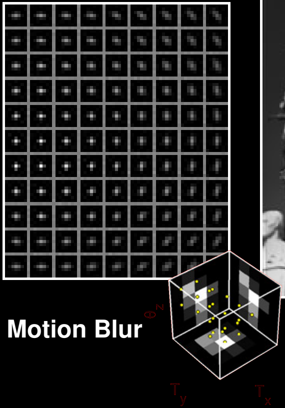


Latent Image

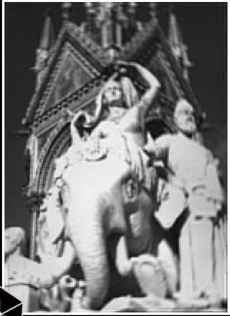
STEP 1: Blur Motion Estimation

SCALE 2 / ITERATION 3

Predicted Image



Motion Blur

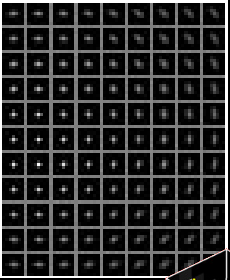


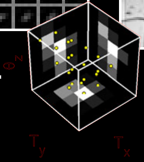
Latent Image

STEP 1: Blur Motion Estimation

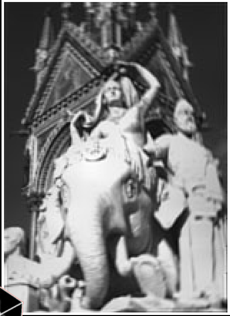
SCALE 2 / ITERATION 4

Predicted Image





Motion Blur

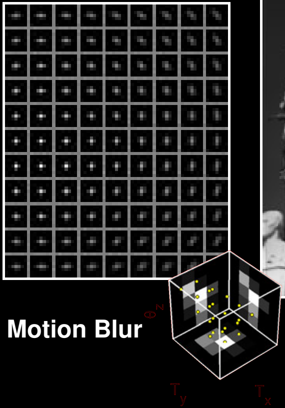


Latent Image

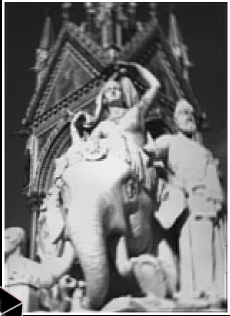
STEP 1: Blur Motion Estimation

SCALE 2 / ITERATION 5

Predicted Image



Motion Blur

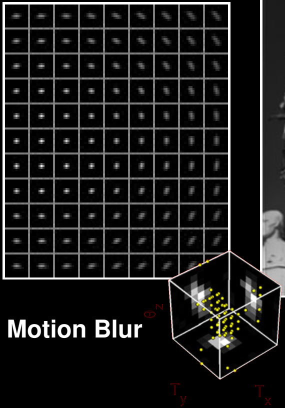


Latent Image

STEP 1: Blur Motion Estimation

SCALE 3 / ITERATION 1

Predicted Image



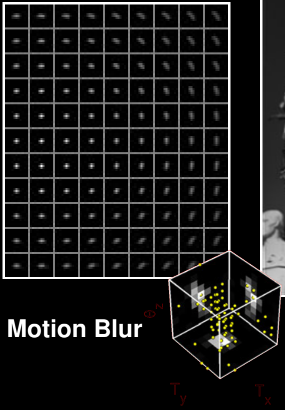
Motion Blur

Latent Image

STEP 1: Blur Motion Estimation

SCALE 3 / ITERATION 2

Predicted Image



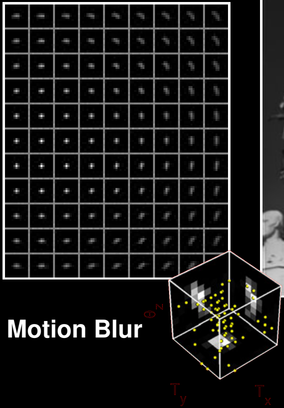
Motion Blur

Latent Image

STEP 1: Blur Motion Estimation

SCALE 3 / ITERATION 3

Predicted Image



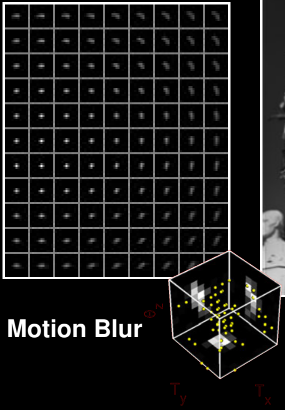
Motion Blur

Latent Image

STEP 1: Blur Motion Estimation

SCALE 3 / ITERATION 4

Predicted Image



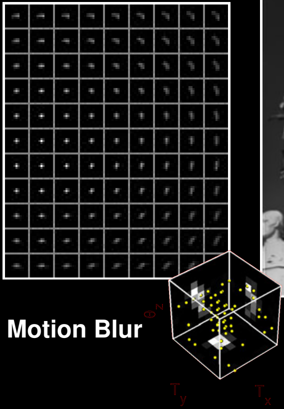
Motion Blur

Latent Image

STEP 1: Blur Motion Estimation

SCALE 3 / ITERATION 5

Predicted Image



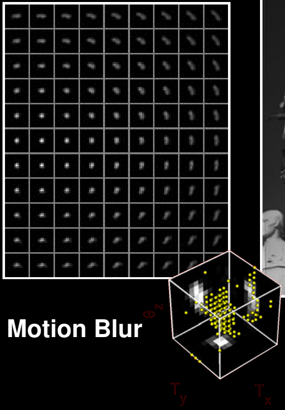
Motion Blur

Latent Image

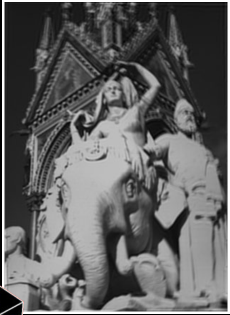
STEP 1 : Blur Motion Estimation

SCALE 4 / ITERATION 1

Predicted Image



Motion Blur

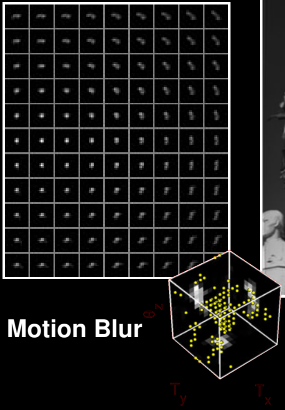


Latent Image

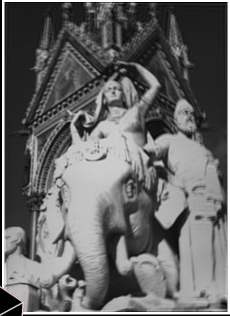
Step 1: Blur Motion Estimation

Scale 4 / Iteration 2

Predicted Image



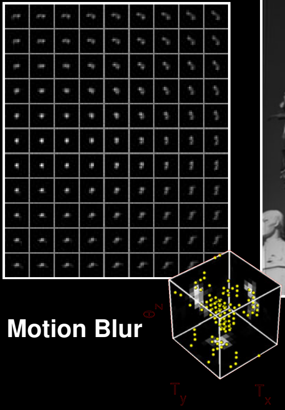
Motion Blur



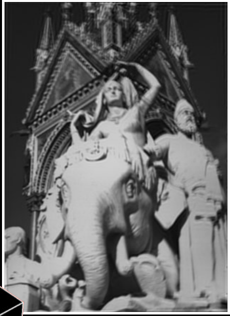
Latent Image

SCALE 4 / ITERATION 3

Predicted Image



Motion Blur



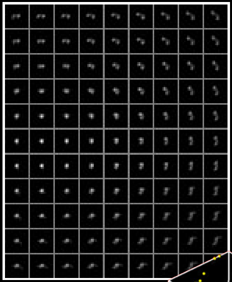
Latent Image

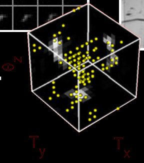
STEP 1 : Blur Motion Estimation

Step 1 : Blur Motion Estimation

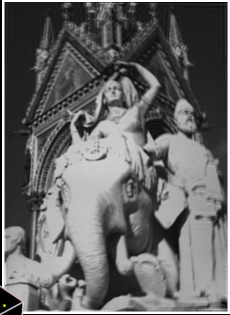
Scale 4 / Iteration 4

Predicted Image





Motion Blur

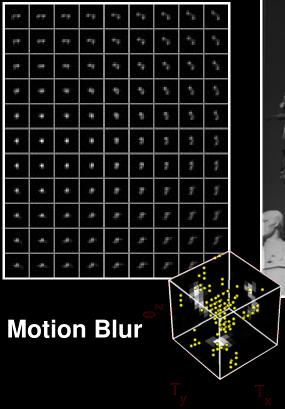


Latent Image

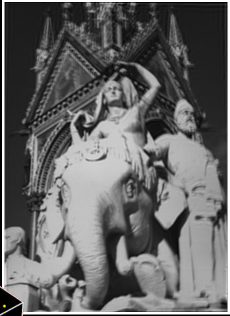
Step 1: Blur Motion Estimation

Scale 4 / Iteration 5

Predicted Image



Motion Blur

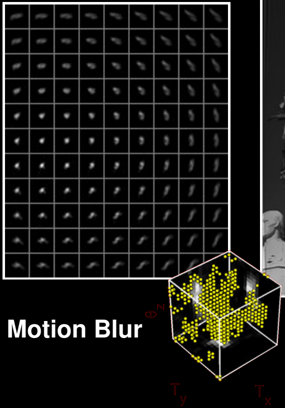


Latent Image

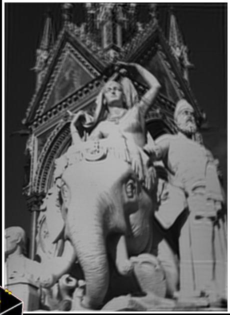
STEP 1: Blur Motion Estimation

SCALE 5 / ITERATION 1

Predicted Image



Motion Blur

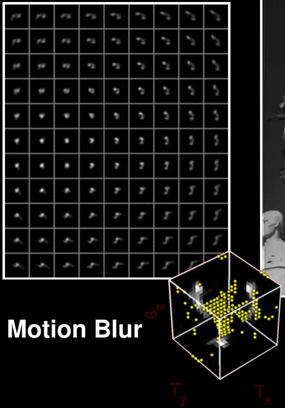


Latent Image

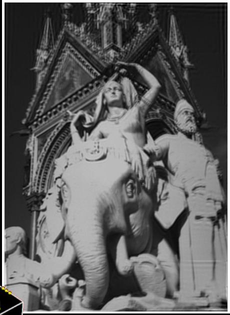
Step 1: Blur Motion Estimation

Scale 5 / Iteration 2

Predicted Image



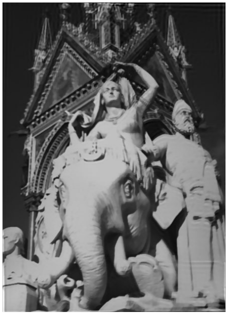
Motion Blur



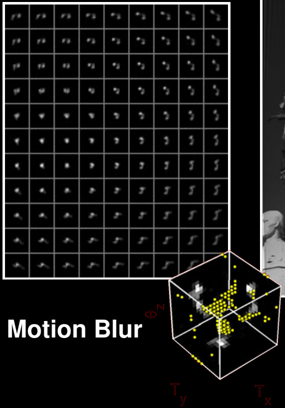
Latent Image

Step 1: Blur Motion Estimation

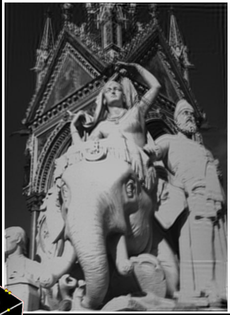
Scale 5 / Iteration 3



Predicted Image



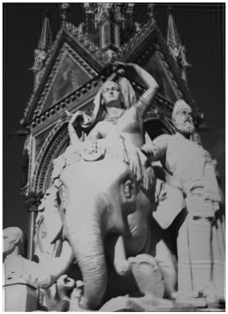
Motion Blur



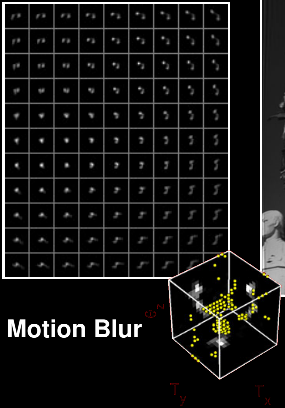
Latent Image

STEP 1: Blur Motion Estimation

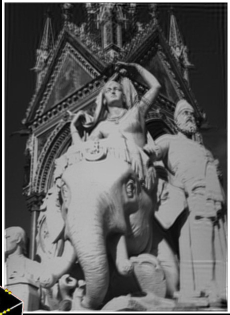
SCALE 5 / ITERATION 4



Predicted Image



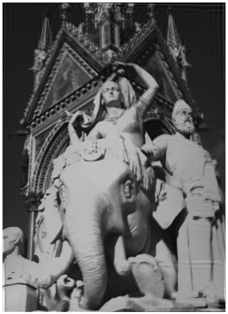
Motion Blur



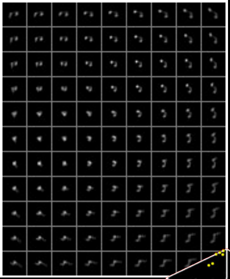
Latent Image

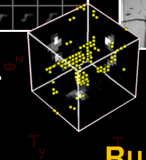
Step 1: Blur Motion Estimation

Scale 5 / Iteration 5

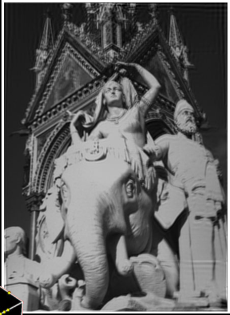


Predicted Image





Motion Blur

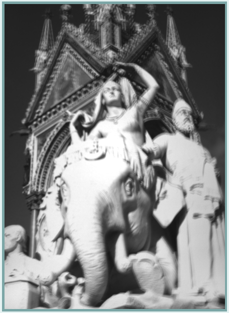


Latent Image

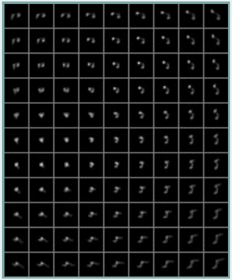
Runtime: 8.316s

Step 2: Latent Image Estimation

STEP 2: Latent Image Estimation

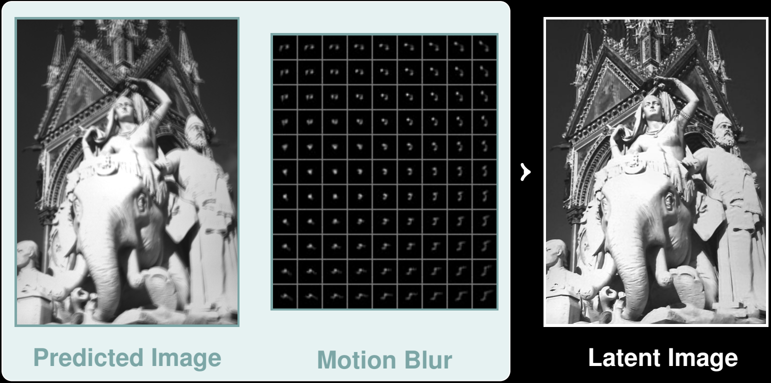


Predicted Image



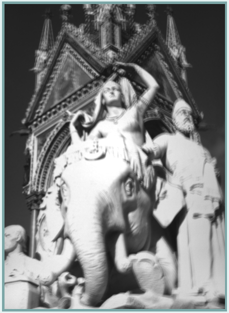
Motion Blur

STEP 2: Latent Image Estimation

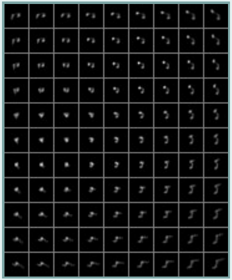


RUNTIME: 0.132s

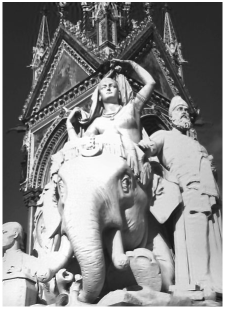
Step 2: Latent Image Estimation



Predicted Image



Motion Blur



Latent Image

Total Runtime: 8.454s

Comparison

Comparison

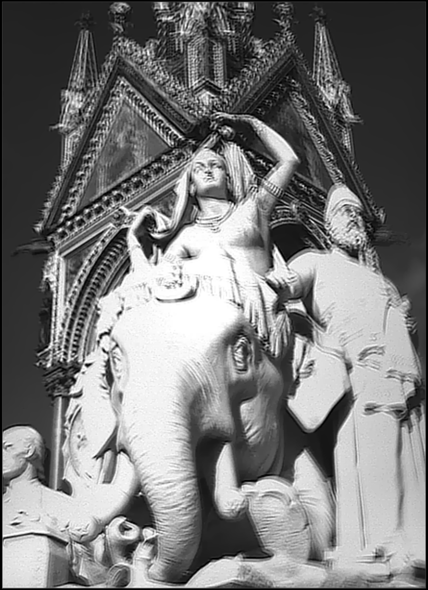
with [Harmeling et al. 2010 NIPS]

[Harmeling et al. 2010 NIPS] estimates the motion blur locally and doesn't enforce global consistency. This leads to unintentional local distortions as most noticeable in the elephant example. Harmeling et al. also provide ground truth data for the captured PSF. A comparison with our estimated PSF shows very close agreement and serves as an additional validation of our approach. We also show their results for [Cho and Lee 2009 Siggraph] who assume uniform blur.

Elephant

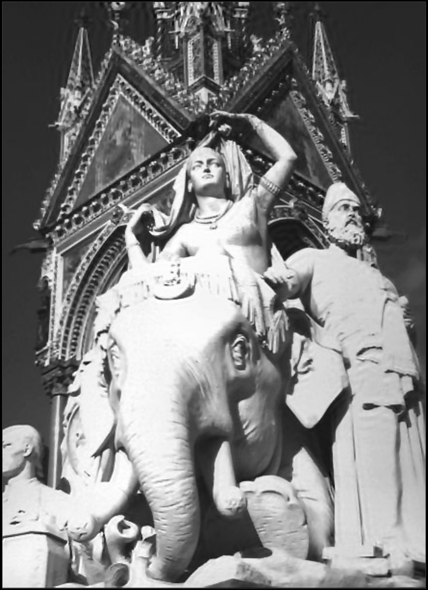
Image Size	441x611
PSF Size	19x19
Kernel Estimation	8.316 sec
Final deconvolution	0.132 sec
Total processing time	8.454 sec

Blurred image



[Cho and Lee 2009 Siggraph]

[Harmeling et al. 2010 NIPS]



Our approach

Blurred image

Comparison

with [Whyte et al. 2010 CVPR]

For comparison, we also show their results of the method of [Fergus et al. 2006 Siggraph] which assume uniform blur.

Pantheon

Image Size	366x274x2
PSF Size	15x15
Kernel Estimation	5.078 sec
Final deconvolution	0.07 sec
Total processing time	5.156 sec

Blurred image

[Fergus et al. 2006 Siggraph]

[Whyte et al. 2010 CVPR]

Our approach

Blurred image

Comparison

with [Gupta et al. 2010 ECCV]

We also show results for a state-of-the-art single image blind deblurring method which assumes uniform blur [Xu and Jia 2010 ECCV].

Books

Image Size	768x512x3
PSF Size	17x17
Kernel Estimation	10.3 sec
Final deconvolution	0.172 sec
Total processing time	10.482 sec

Blurred image

[Xu and Jia 2010 ECCV]

[Gupta et al. 2010 ECCV]

Our approach

Blurred image

Comparison

with [Joshi et al. 2010 Siggraph]

[Joshi et al. 2010 Siggraph] uses additional inertial measurement sensor data. In contrast, we use the blurry image only. We also show the result for a state-of-the-art single image blind deblurring method which assumes uniform blur [Xu and Jia 2010 ECCV].

Coke Cans

Image Size	1123x749x3
PSF Size	21x21
Kernel Estimation	12.776 sec
Final deconvolution	0.27 sec
Total processing time	13.07 sec

Blurred image

[Xu and Jia 2010 ECCV]

[Joshi et al. 2010 Siggraph] (using sensor data)

Our approach

Blurred image

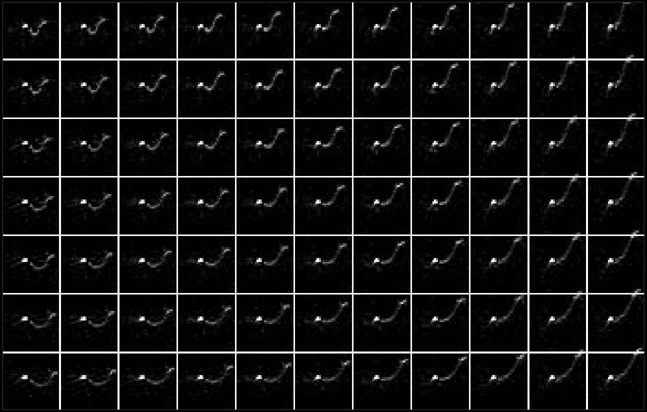
Limitations

Staturation

Our method suffers from saturated pixels regions as in these regions the linearity assumption of our forward model is violated. Although the PSF seems to be estimated correctly, the final deconvolution reveals severe artifacts.

Blurred image

Our approach



Estimated PSF

Objects in Motion

Our method assumes a static scene. The case where objects in the scene are moving (left train) is not covered by our approach.

Superimposed camera and object motion

Severe Blur

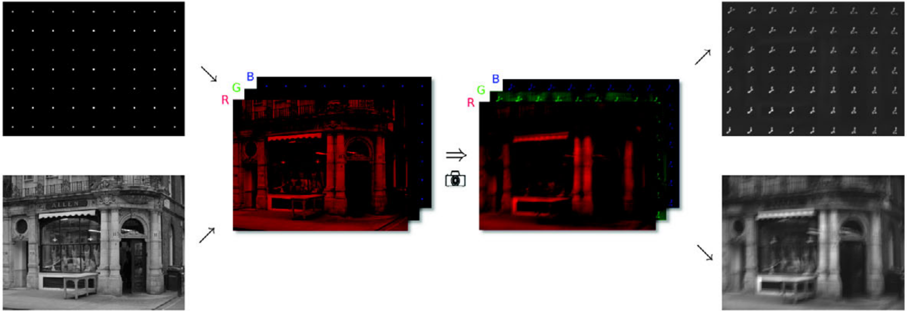
For large camera motions our approach becomes instable and fails to find a meaningful PSF.

Severe camera motion blur

Summary

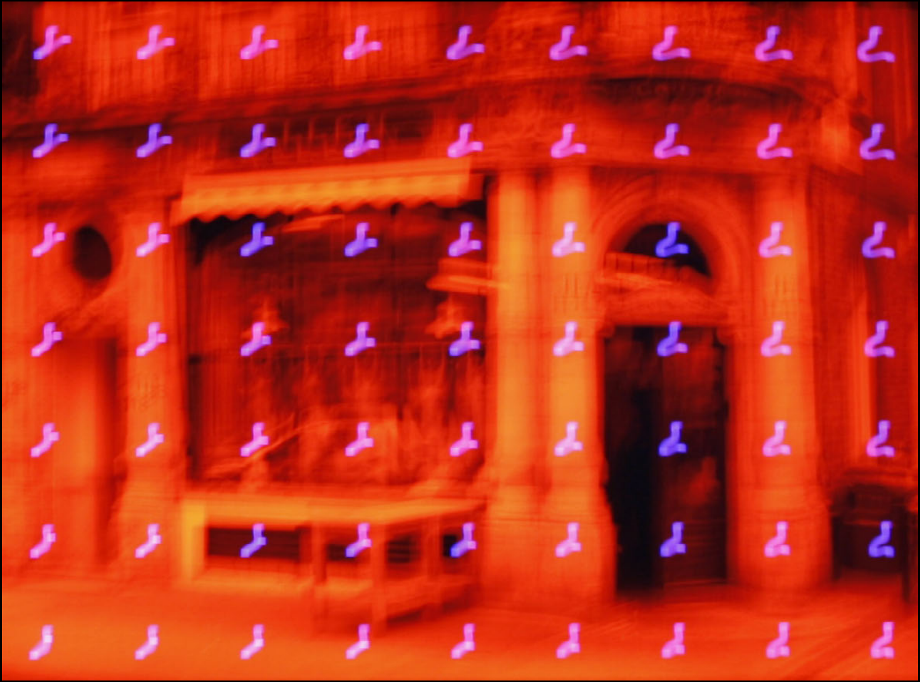
- We proposed a unified imaging model which combines the efficiency of the EFF and incorporates the global camera motion constraint of the Proejctive Motion Path Blur Model.
- Ideally suited for parallel computation ($\rightarrow$ GPU) and hence very efficient and fast.
- Improved and robust sharpening algorithm for a single blurred image with spatially varying blur.
- **Visit our poster tonight at poster stand 75**
- **Visit our webapge at `http://webdav.is.mpg.de/pixel/fast_removal_of_camera_shake`**

How to Visualize Camera Shake...

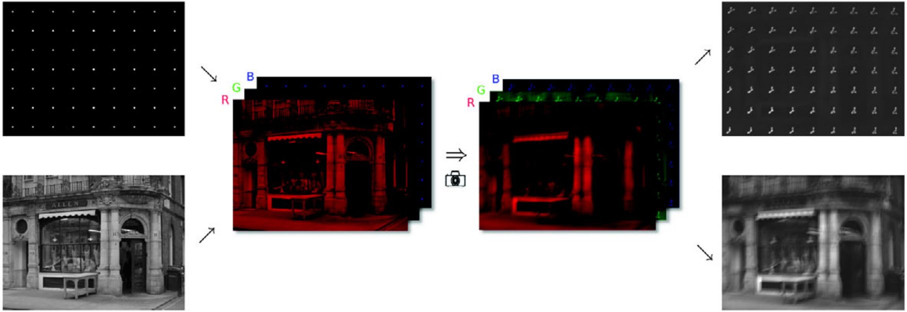


Experimental Setup of Harmeling et al. (NIPS 2010)


ALLEN
117
117
118
118



How to Visualize Camera Shake...

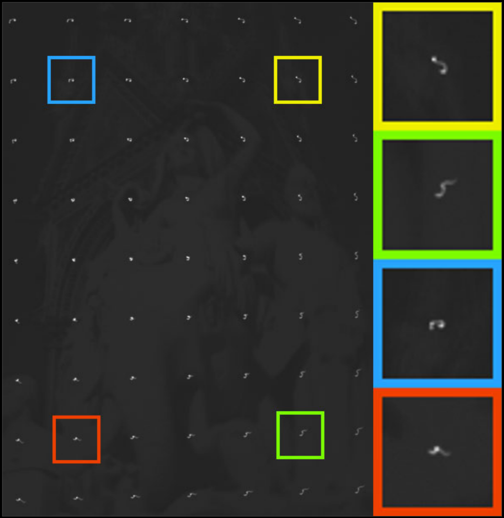


Experimental Setup of Harmeling et al. (NIPS 2010)

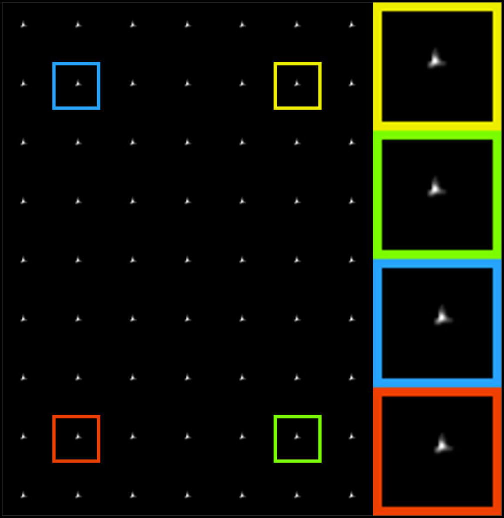
Elephant

PSF comparison

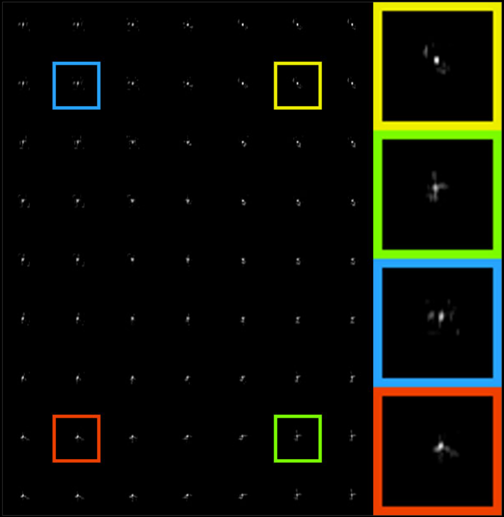
While the local PSF estimation of [Harmeling et al. 2010 NIPS] faces difficulties in regions with little edge information (as e.g. sky), our camera motion constrained EFF framework is able to use the textural information within the entire image to infer a globally consistent PSF. The benefit is not only apparent in the elephant example but also in the subsequent example of the vintage car (bottom left).



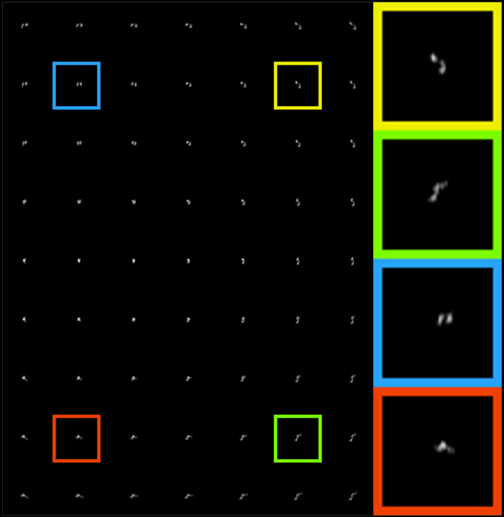
Ground truth



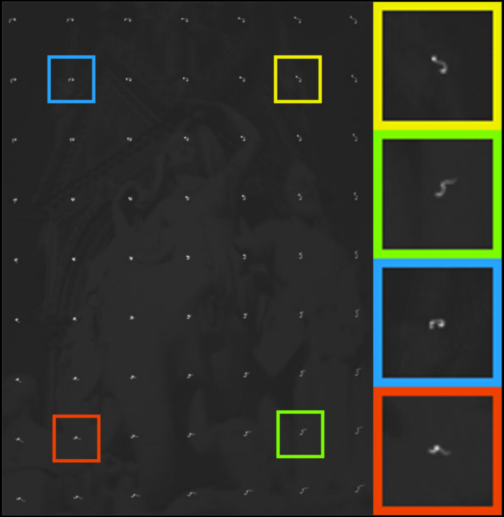
[Cho and Lee 2009 Siggraph]



[Harmeling et al. 2010 NIPS]



Our approach



Ground truth

Butcher Shop

Image Size	601x401
PSF Size	25x25
Kernel Estimation	13.638 sec
Final deconvolution	0.036 sec
Total processing time	13.686 sec

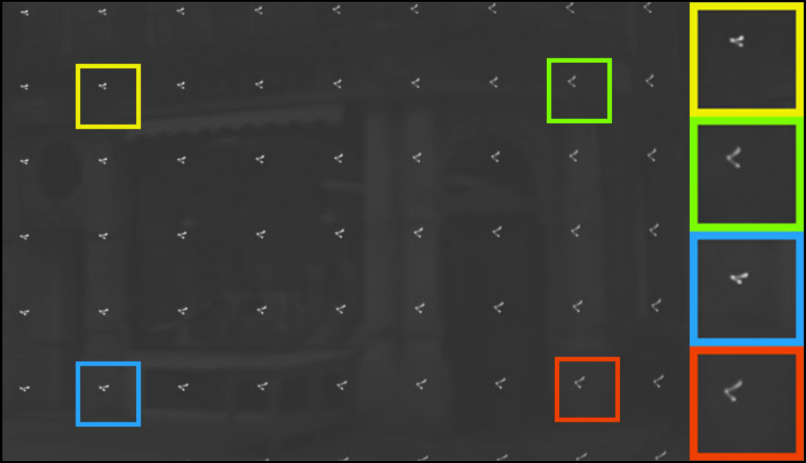
Blurred image

[Cho and Lee 2009 Siggraph]

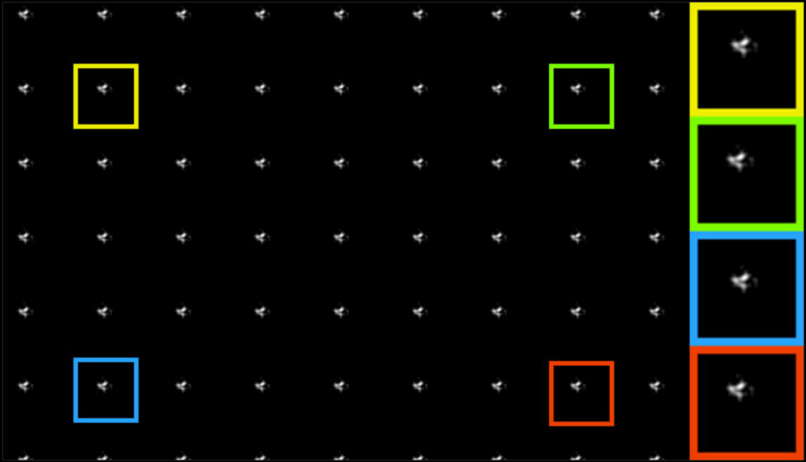
[Harmeling et al. 2010 NIPS]

Our approach

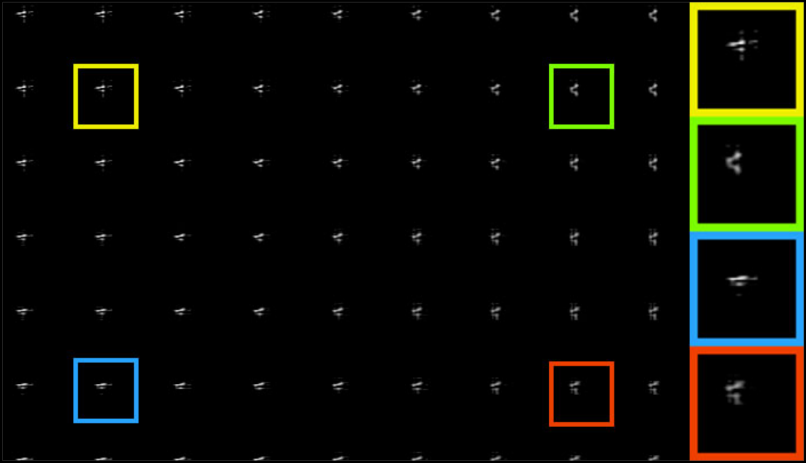
Blurred image



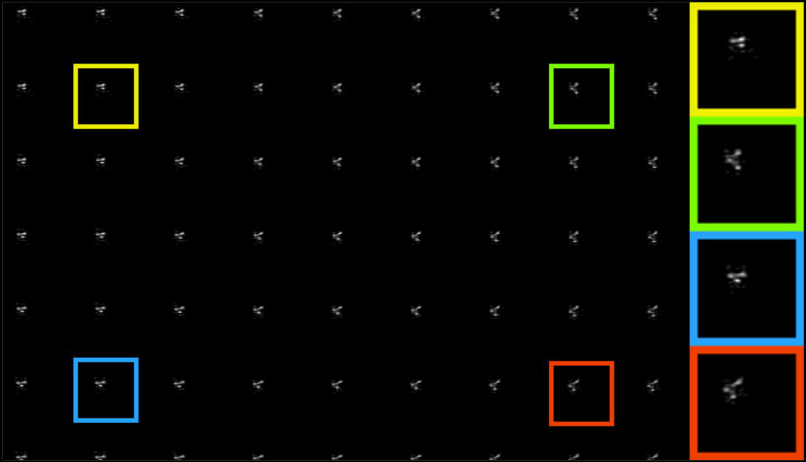
Ground truth



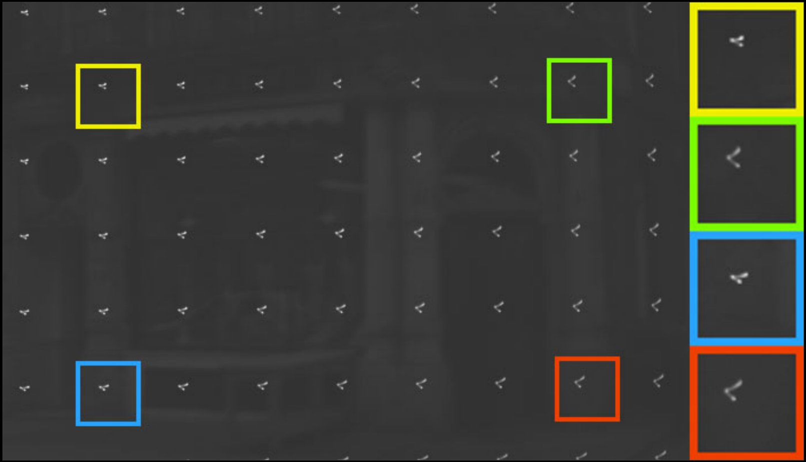
[Cho and Lee 2009 Siggraph]



[Harmeling et al. 2010 NIPS]



Our approach



Ground truth

Vintage Car

PSF comparison

Blurred image

[Cho and Lee 2009 Siggraph]

[Harmeling et al. 2010 NIPS]

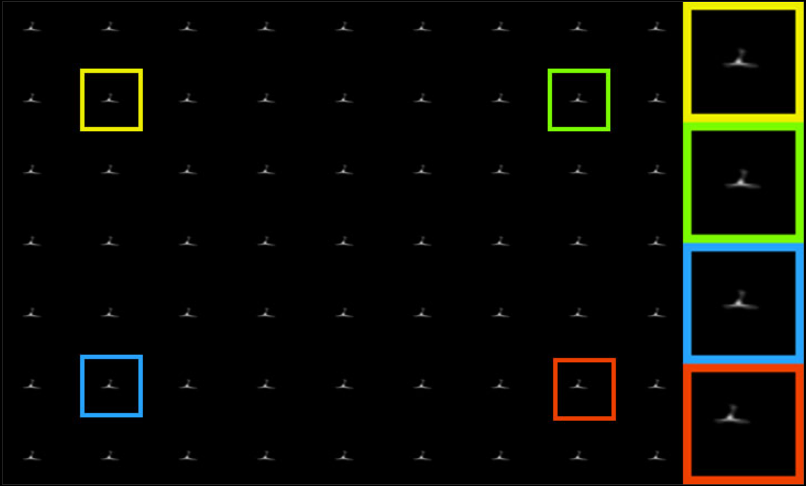
Our approach

Blurred image

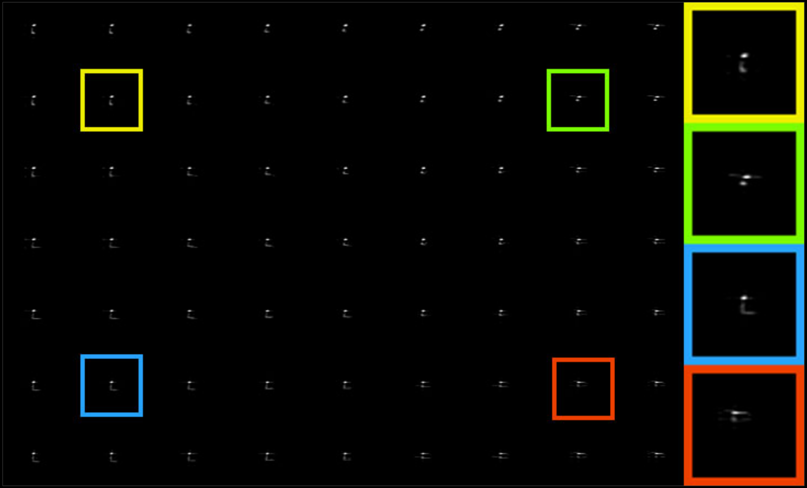
Vintage Car

Image Size	621x441
PSF Size	19x19
Kernel Estimation	8.306 sec
Final deconvolution	0.038 sec
Total processing time	8.354 sec

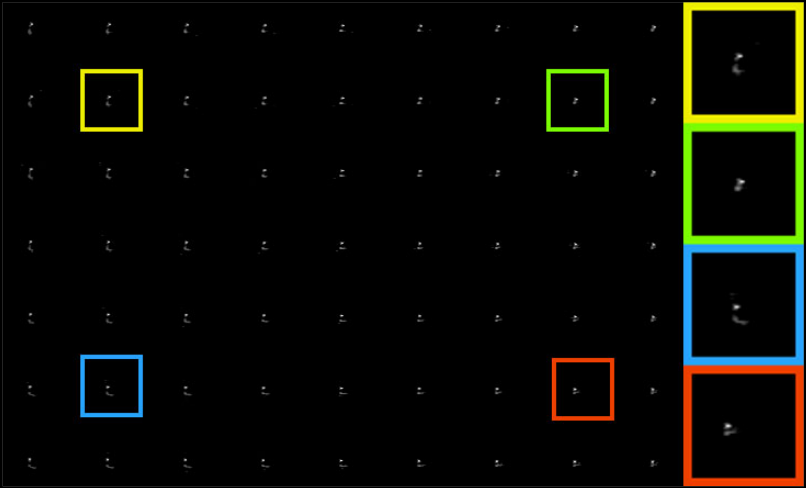
Ground truth



[Cho and Lee 2009 Siggraph]



[Harmeling et al. 2010 NIPS]



Our approach

Ground truth

Petrol Station Example

This example image was reported to be challenging due to the great variation of depth in the scene. The slight shift in our restoration stems from the translational invariance of the deblurring problem, i.e. a shift in the restored image can be compensated by a shift in the opposite direction of the corresponding PSF.

Image Size	679x406x3
PSF Size	19x19
Kernel Estimation	8.194 sec
Final deconvolution	0.132 sec
Total processing time	8.338 sec

Blurred image

[Xu and Jia 2010 ECCV]

[Gupta et al. 2010 ECCV]

Our approach

Blurred image

Notre Dame

Image Size	354x265x3
PSF Size	21x21
Kernel Estimation	7.388 sec
Final deconvolution	0.082 sec
Total processing time	7.474 sec

Blurred image

[Fergus et al. 2006 Siggraph]

[Whyte et al. 2010 CVPR]

Our approach

Blurred image

Statue

[Whyte et al. 2010 CVPR] takes a noisy/blurry image pair for image restoration. In contrast, we use the blurry image only. We also show the result for a state-of-the-art single image blind deblurring method which assumes uniform blur [Cho and Lee 2009 Siggraph].

Image Size	523x710x3
PSF Size	21x21
Kernel Estimation	12.434 sec
Final deconvolution	0.174 sec
Total processing time	12.618 sec

Blurred image

[Cho and Lee 2009 Siggraph]

[Whyte et al. 2010 CVPR]

Our approach

Blurred image

References

S. Cho and S. Lee.
Fast Motion Deblurring, Siggraph Asia 2009.

R. Fergus, B. Singh, A. Hertzmann and S.T. Roweis and W.T. Freeman.
Removing camera shake from a single image, Siggraph 2006.

A. Gupta, N. Joshi, L. Zitnick, M. Cohen and B. Curless.
Single Image Deblurring Using Motion Density Functions, ECCV, 2010.

S. Harmeling, M. Hirsch and B. Schölkopf.
Space-Variant Single-Image Blind Deconvolution for Removing Camera Shake, NIPS 2010.

A. Klöckner, N. Pinto, Y. Lee, B. Catanzaro, P. Ivanov and A. Fasih.
PyCUDA: GPU run-time code generation for high-performance computing, Performance Computing 2009.

N. Joshi, S.B. Kang, C.L. Zitnick and R. Szeliski.
Image deblurring using inertial measurement sensors, Siggraph 2010.

O. Whyte, J. Sivic, A. Zisserman and J. Ponce.
Non-uniform Deblurring for Shaken Images, CVPR, 2010.

L. Xu and J. Jia.
Two-Phase Kernel Estimation for Robust Motion Deblurring, ECCV 2010.