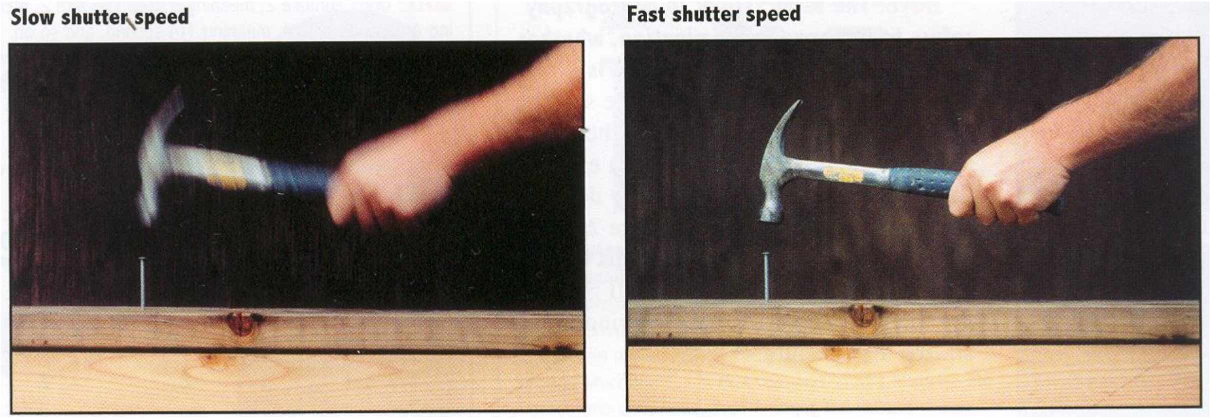
Computational Cameras

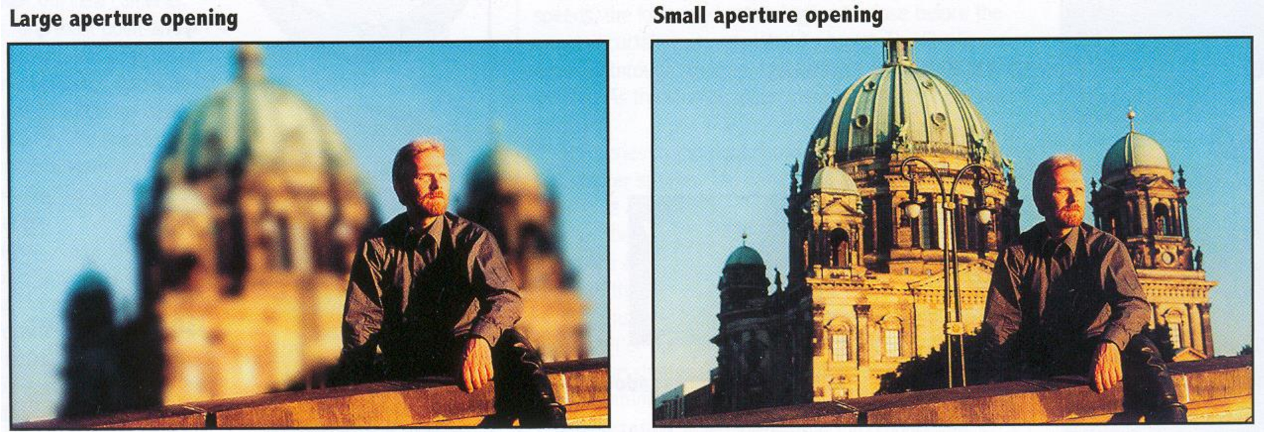
Anat Levin
Department of Electrical Engineering,
Technion

Blurring problems in imaging

- Motion blur
  - Flutter shutter
  - Motion invariant photography

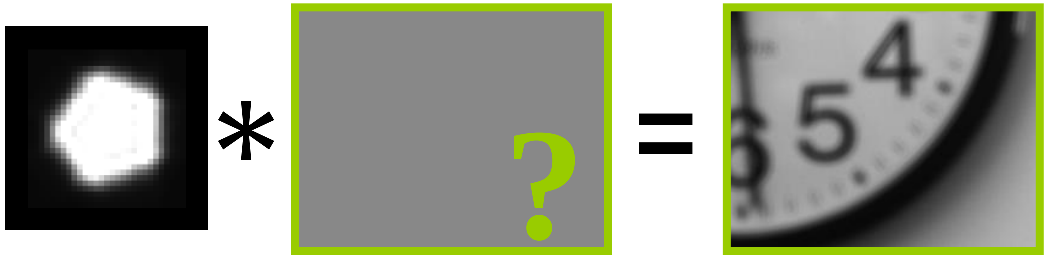


- Defocus blur
  - Coded aperture
  - Lattice focal
  - Flexible depth of field
  - Wavefront coding



Blurring and deblurring

$$y=k*x$$



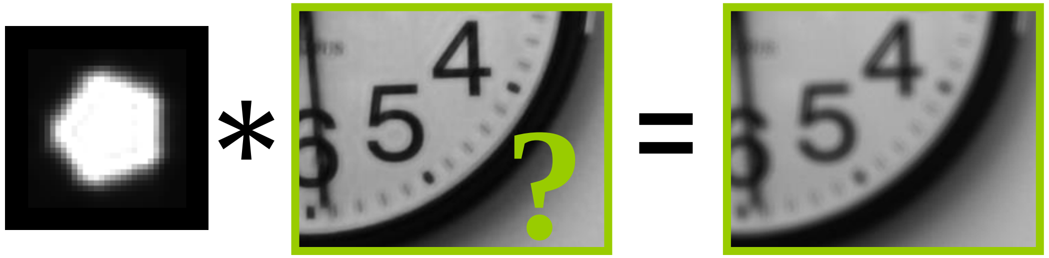
Deblurring is hard:

- **Need to know convolution kernel**
- **Deconvolution is ill posed**

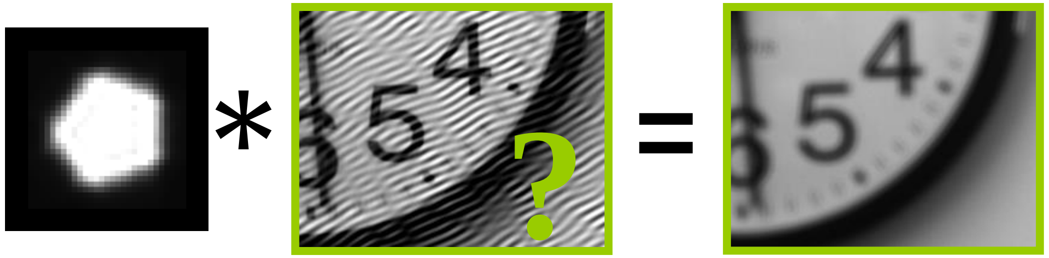
Deconvolution is ill posed

$$k*x=y$$

Solution 1:



Solution 2:



Convolution and deconvolution in Frequency domain

Primal domain: convolution $\mathbf{y=k*x}$

Frequencey domain: multiplication $\boldsymbol{Y_w = K_w \cdot X_w}$

Deconvolution in Frequency domain: division

$$\widehat{\boldsymbol{X_w}} = \frac{\boldsymbol{Y_w}}{\boldsymbol{K_w}}$$

Division by zero

Deconvolution and noise amplification

Primal domain: convolution $\quad \mathbf{y=k*x} \quad \color{red}{\mathbf{+n}}$

Frequencey domain: multiplication $\quad \boldsymbol{Y_w = K_w \cdot X_w} \quad \color{red}{\boldsymbol{+N_w}}$

Deconvolution in Frequency domain: division

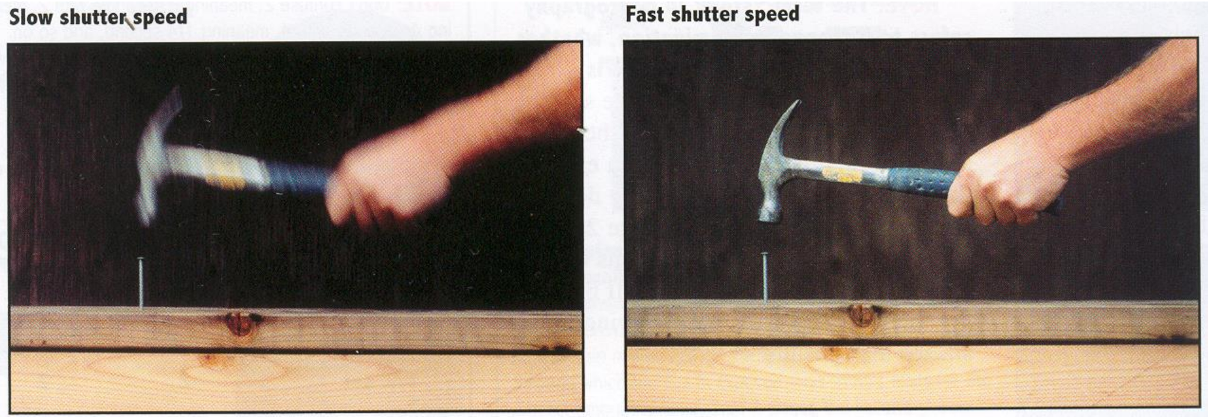
$$\widehat{\boldsymbol{X_w}} = \frac{\boldsymbol{Y_w}}{\boldsymbol{K_w}} \qquad \color{red}{+\frac{\boldsymbol{N_w}}{\boldsymbol{K_w}}}$$

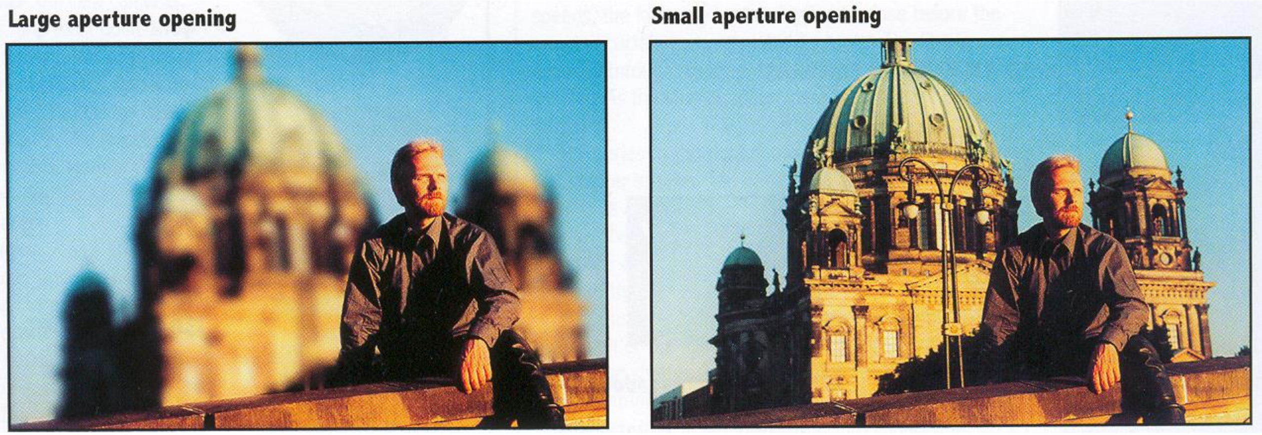
Division by zero

Noise contribution increases when $|\color{red}{K_w}|$ is small

Computational photography approaches to blurring problems

- Motion blur
  - Flutter shutter
  - Motion invariant photography
- Defocus blur
  - Coded aperture
  - Lattice focal
  - Flexible depth of field
  - Wavefront coding





Flutter Shutter

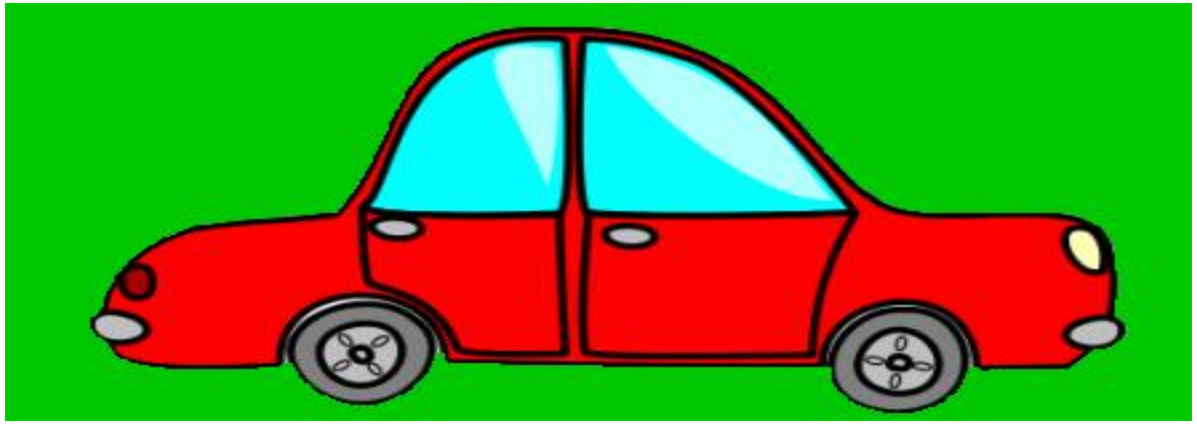
[Raskar et al. 2006]

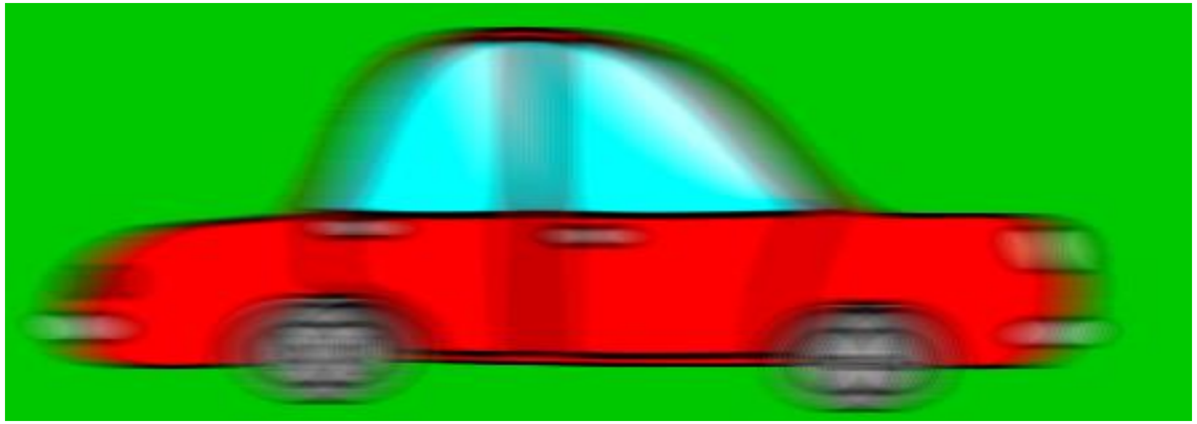
Engineer motion PSF (coding exposure time) so it becomes invertible!

[Raskar et al. 2006]

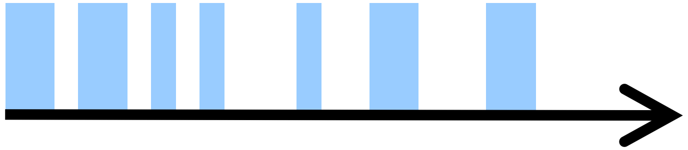
Traditional Camera

Shutter is OPEN

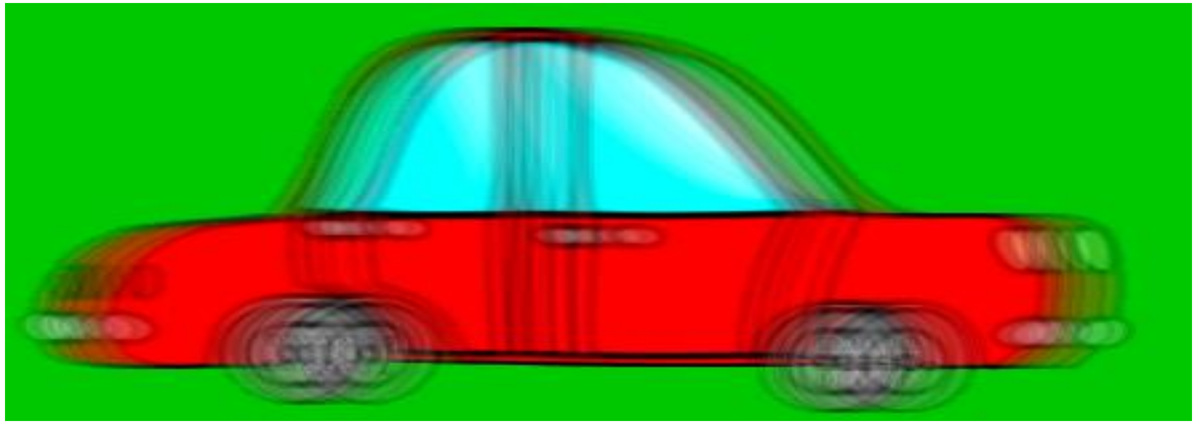




[Raskar et al. 2006]

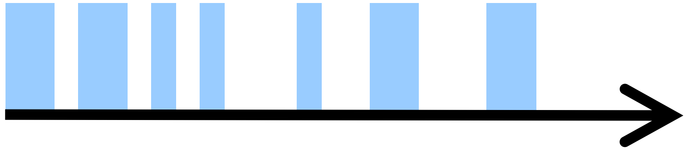


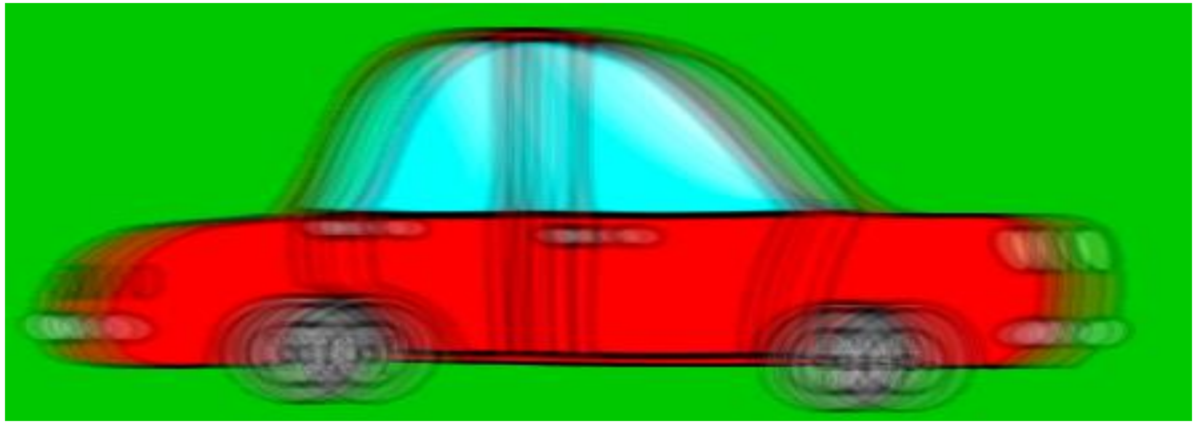
Flutter Shutter



[Raskar et al. 2006]

Shutter is OPEN and CLOSED

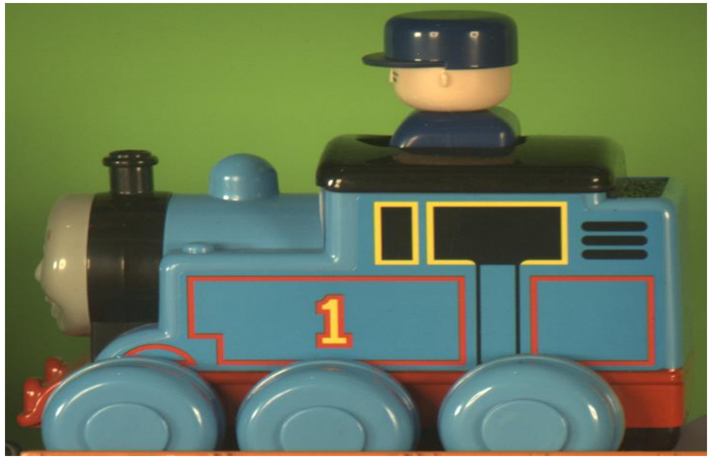




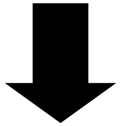
Lab Setup

[Raskar et al. 2006]

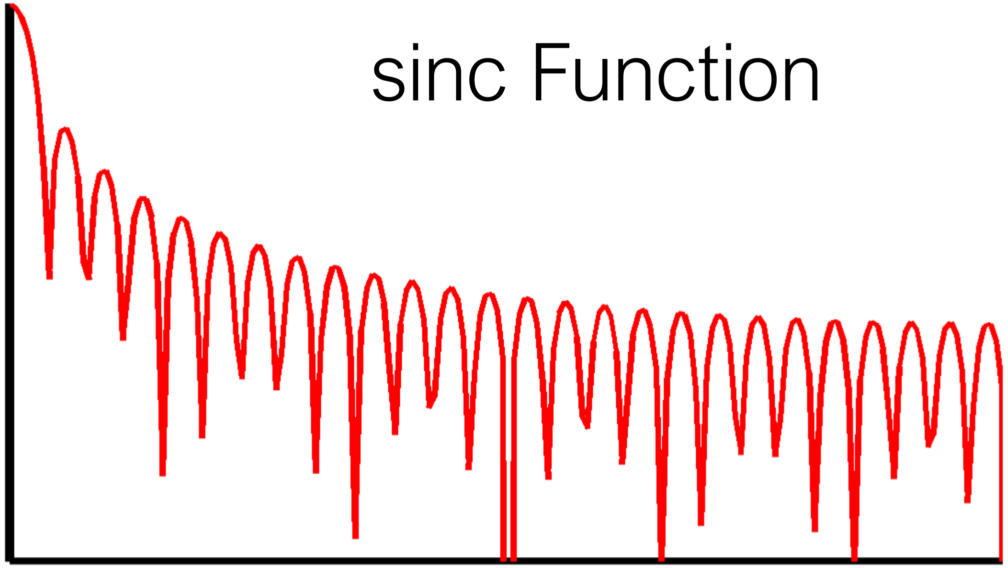
[Raskar et al. 2006]



spatial convolution



sinc Function



Fourier magnitudes

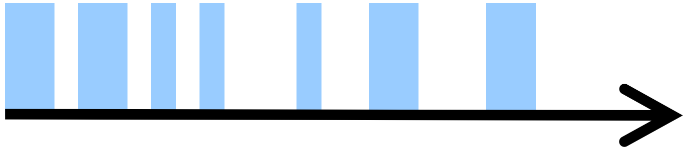
Blurring

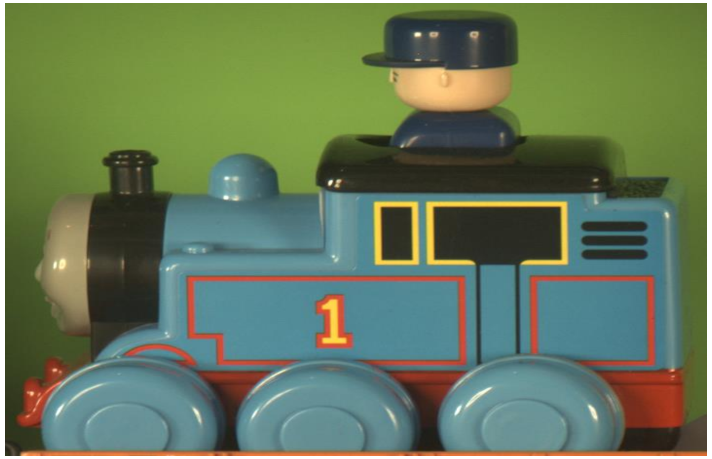
=

Convolution

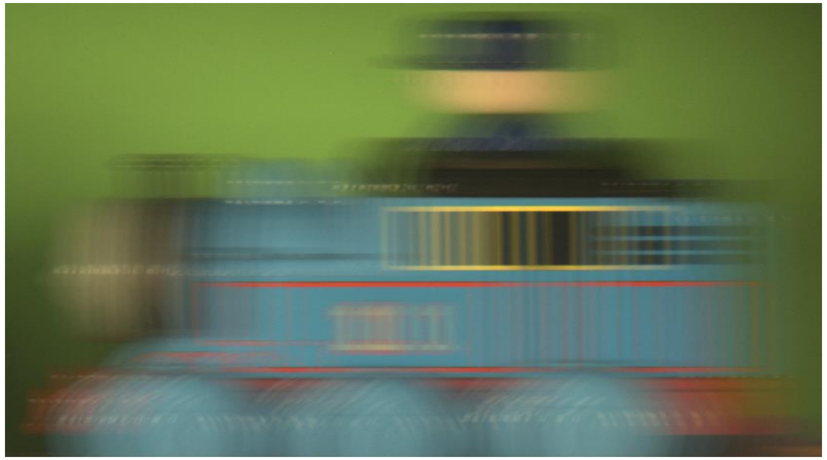
Traditional Camera: Box Filter

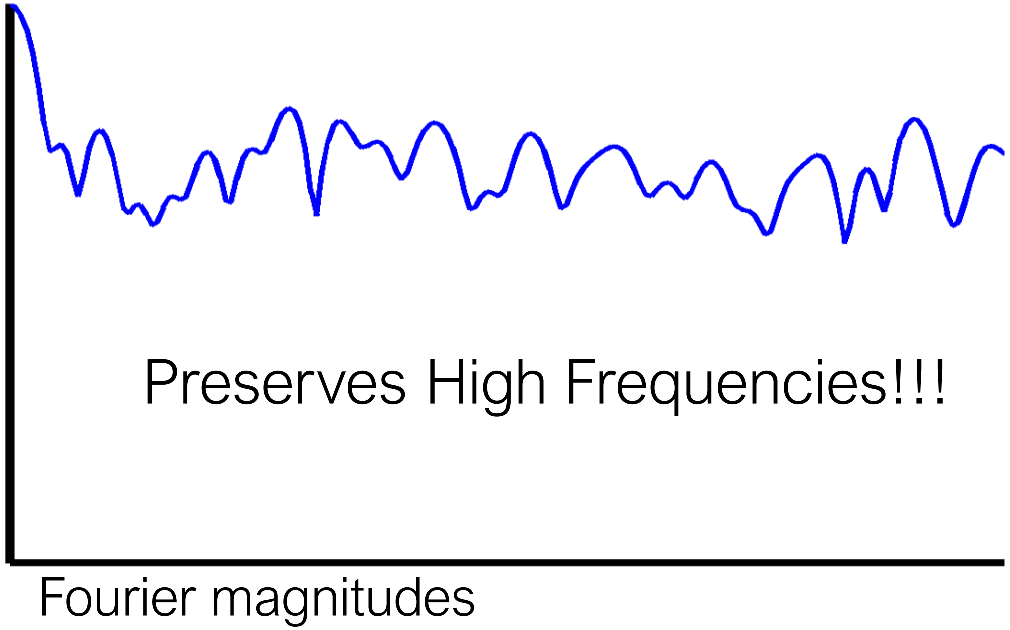
[Raskar et al. 2006]





spatial convolution

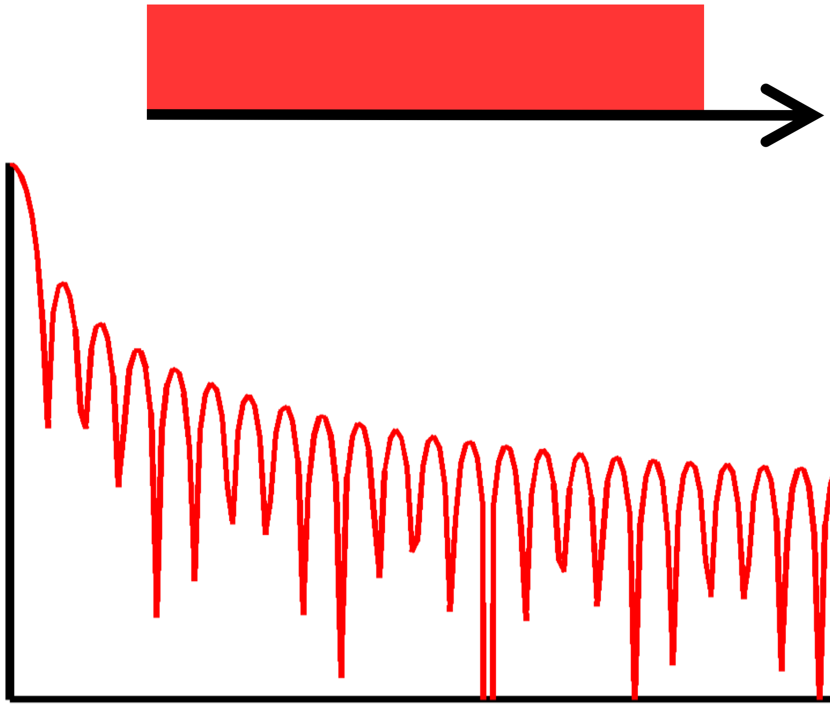


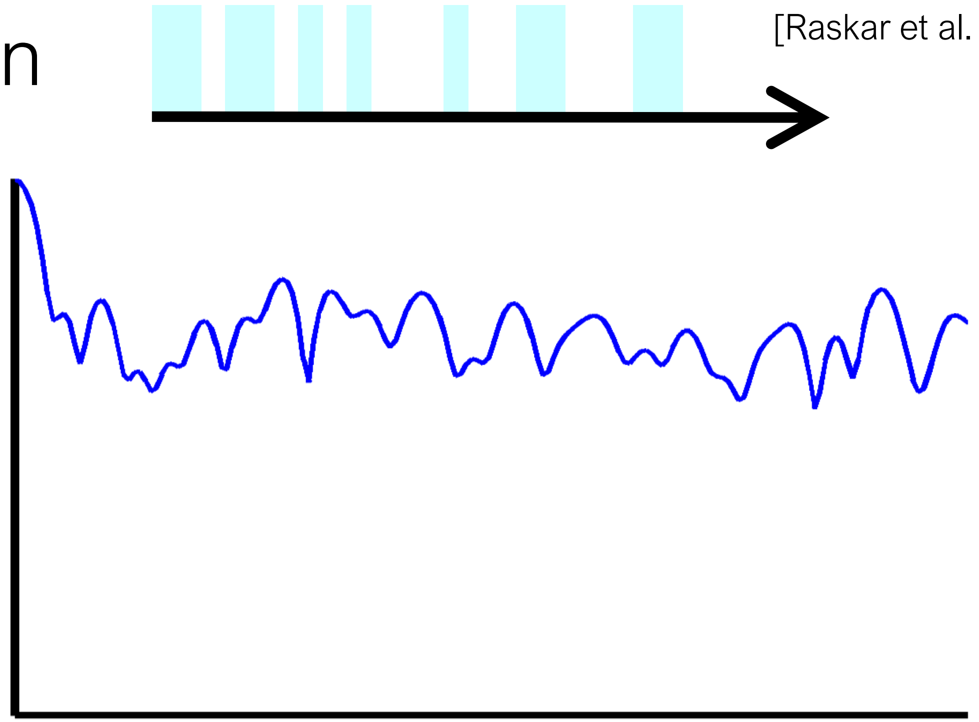


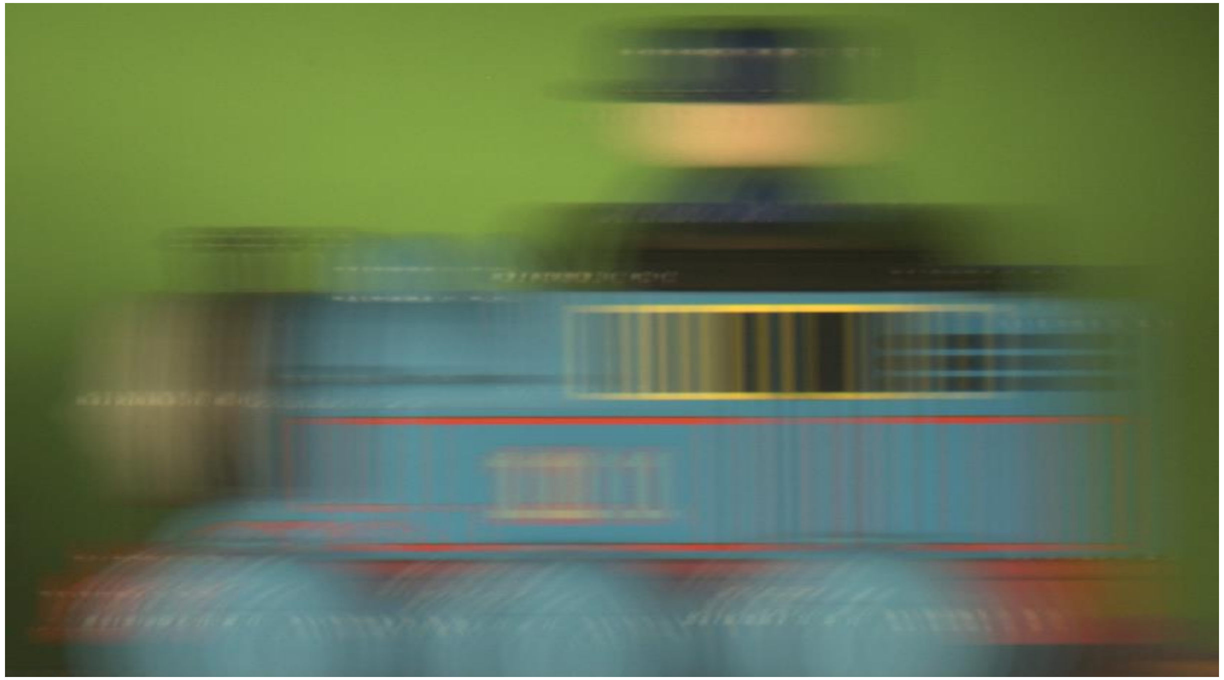
Flutter Shutter: Coded Filter

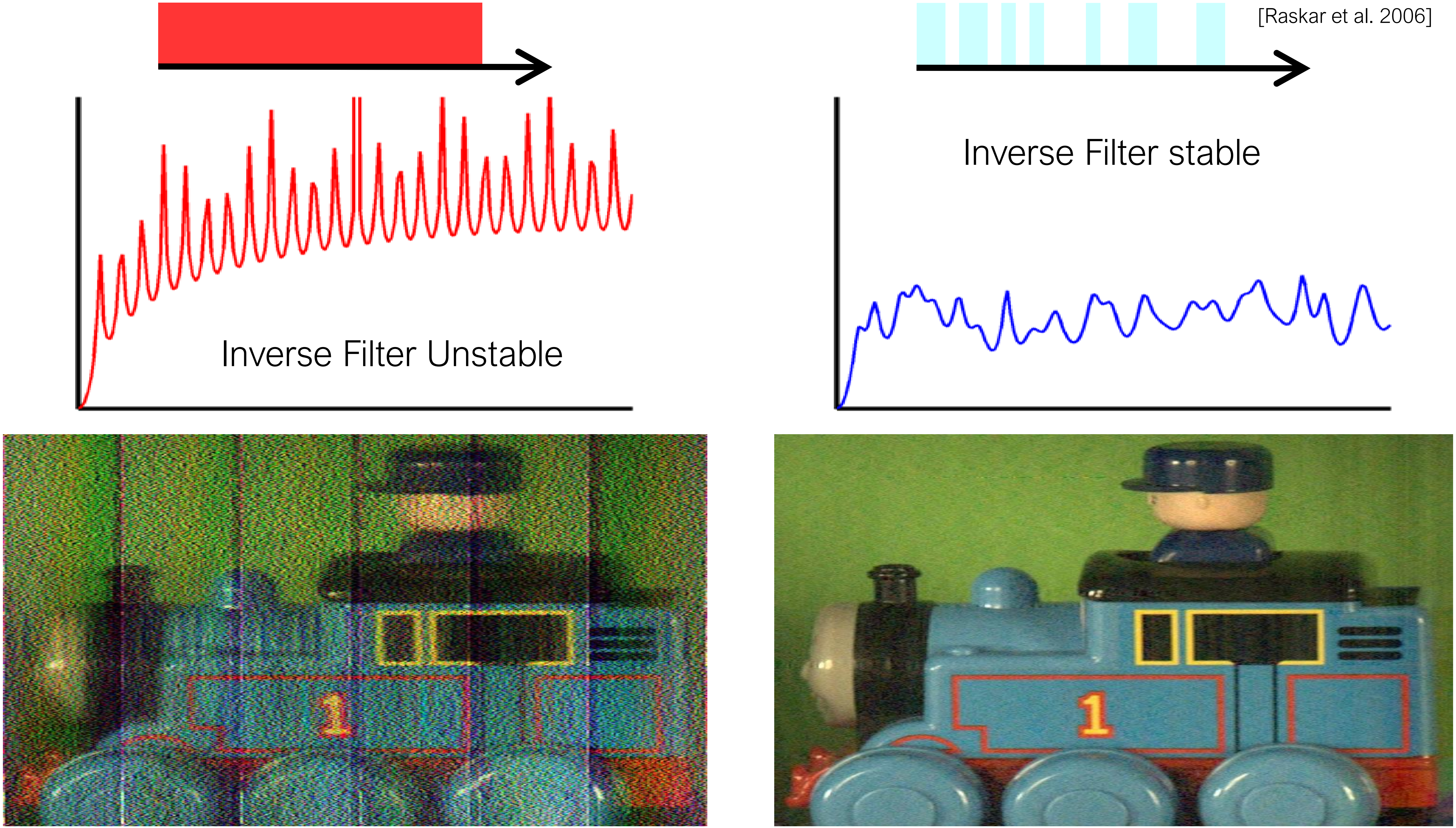
Comparison

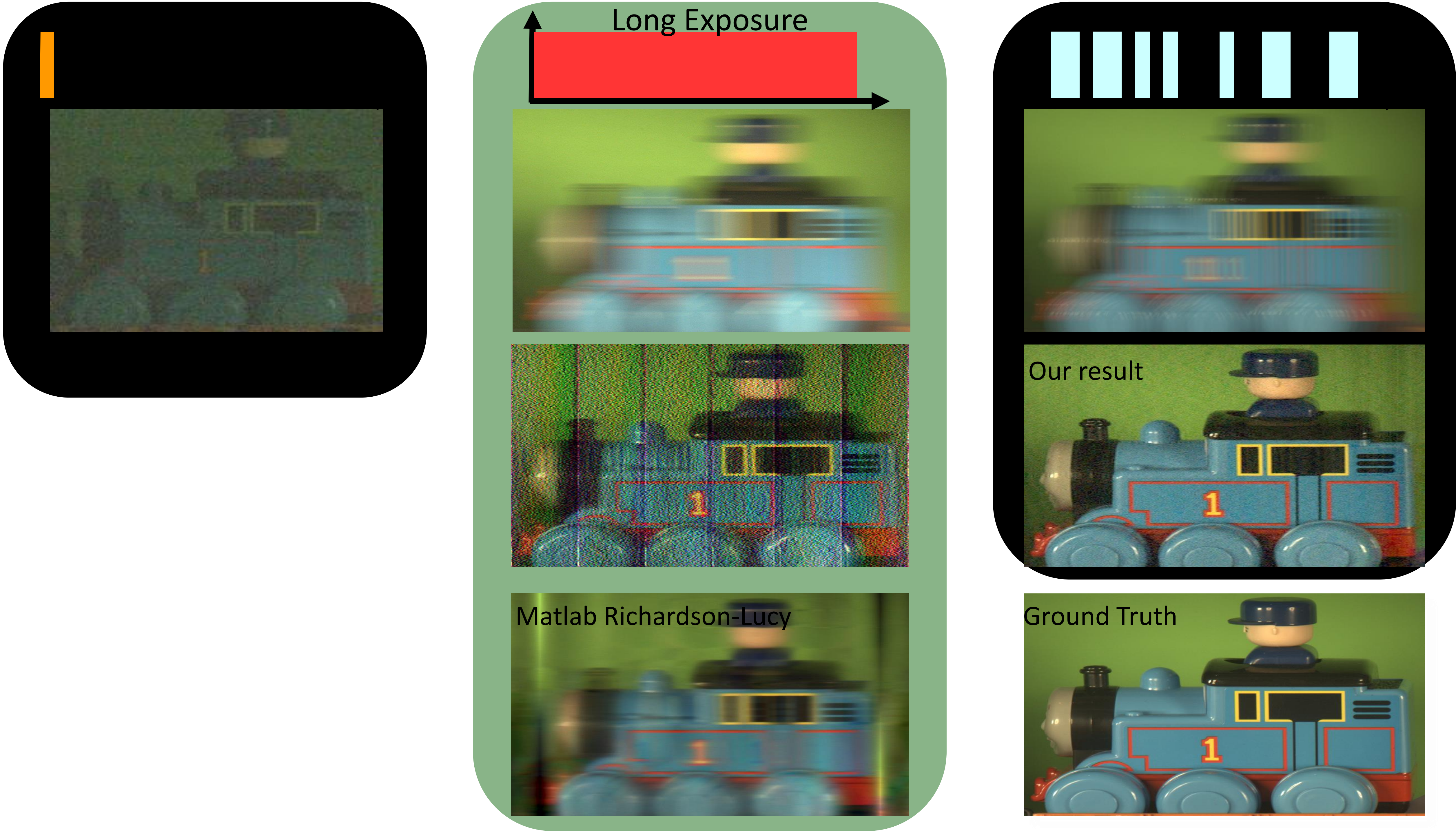
[Raskar et al. 2006]








[Raskar et al. 2006]
Inverse Filter Unstable
Inverse Filter stable
1
1

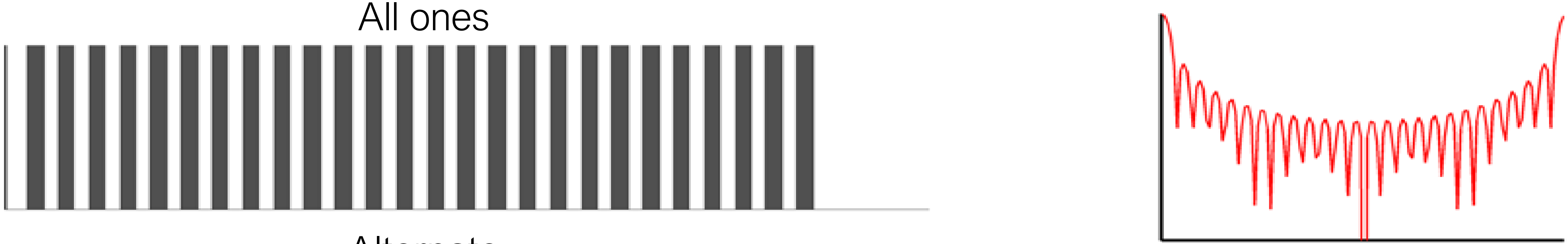


Long Exposure
Our result
Matlab Richardson-Lucy
Ground Truth

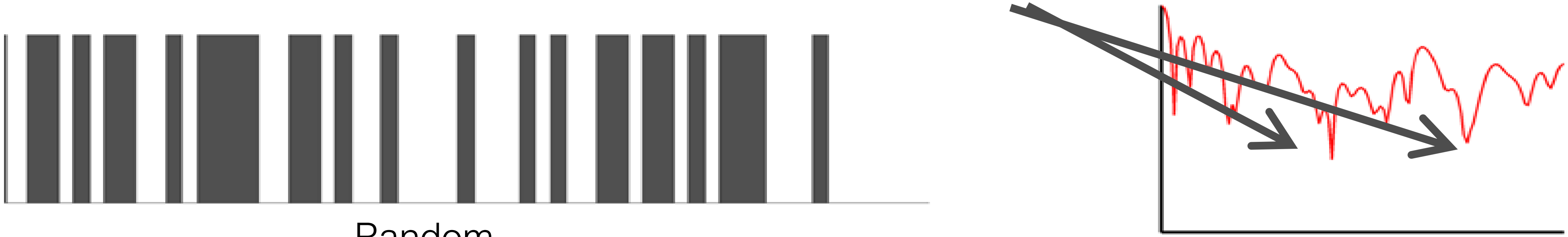
Are all codes "good"?

[Raskar et al. 2006]

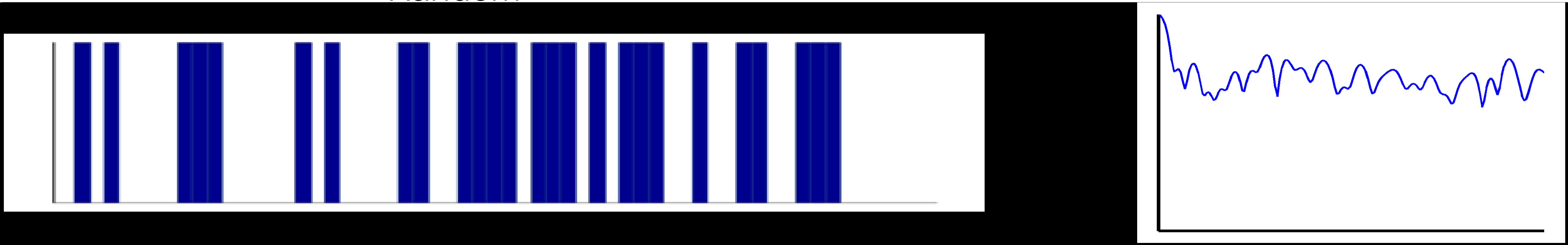
All ones



Alternate



Random



[Raskar et al. 2006]

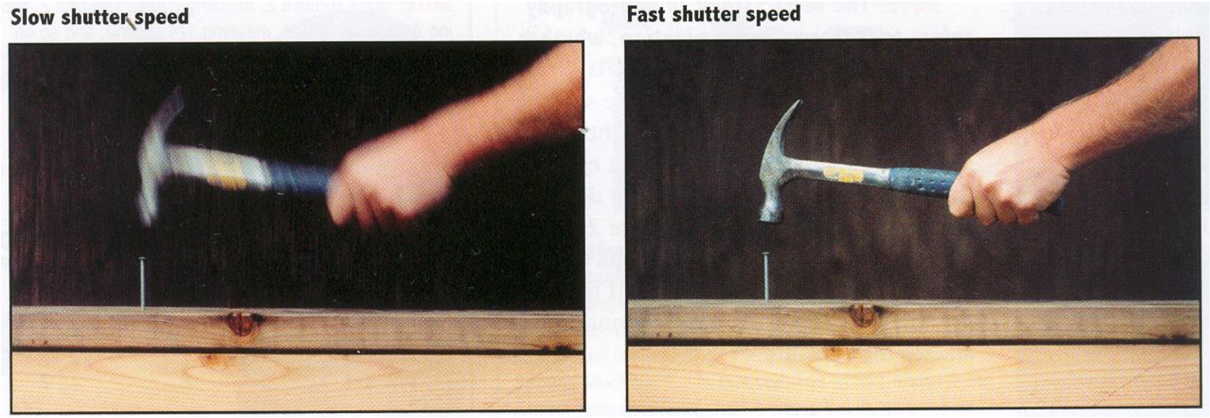
License Plate Retrieval

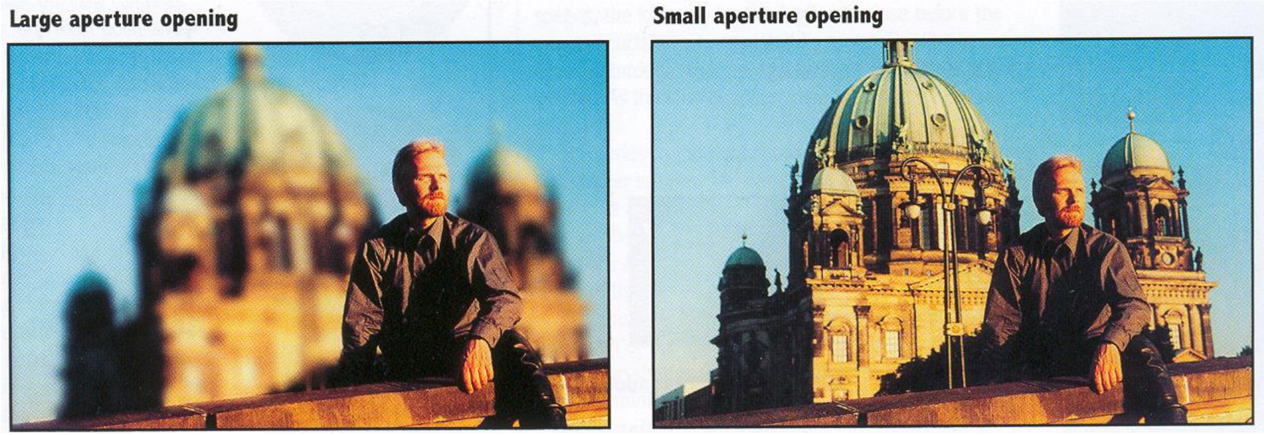
[Raskar et al. 2006]

License Plate Retrieval

Computational photography approaches to blurring problems

- Motion blur
  - Flutter shutter
  - Motion invariant photography
- Defocus blur
  - Coded aperture
  - Lattice focal
  - Flexible depth of field
  - Wavefront coding



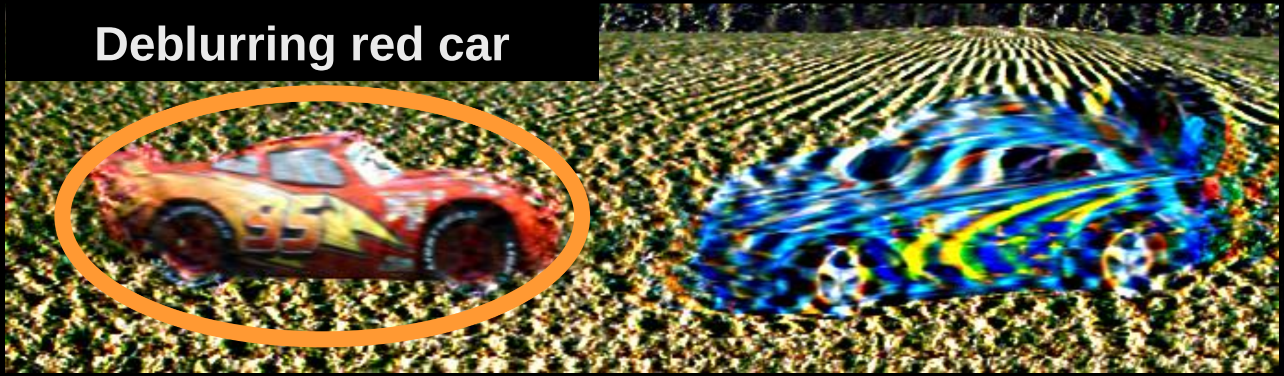


Overcoming motion blur

Levin et al. **Motion Invariant Photography** *SIGGRAPH, 2008.*

Overcoming motion blur

Levin et al. **Motion Invariant Photography** *SIGGRAPH, 2008.*

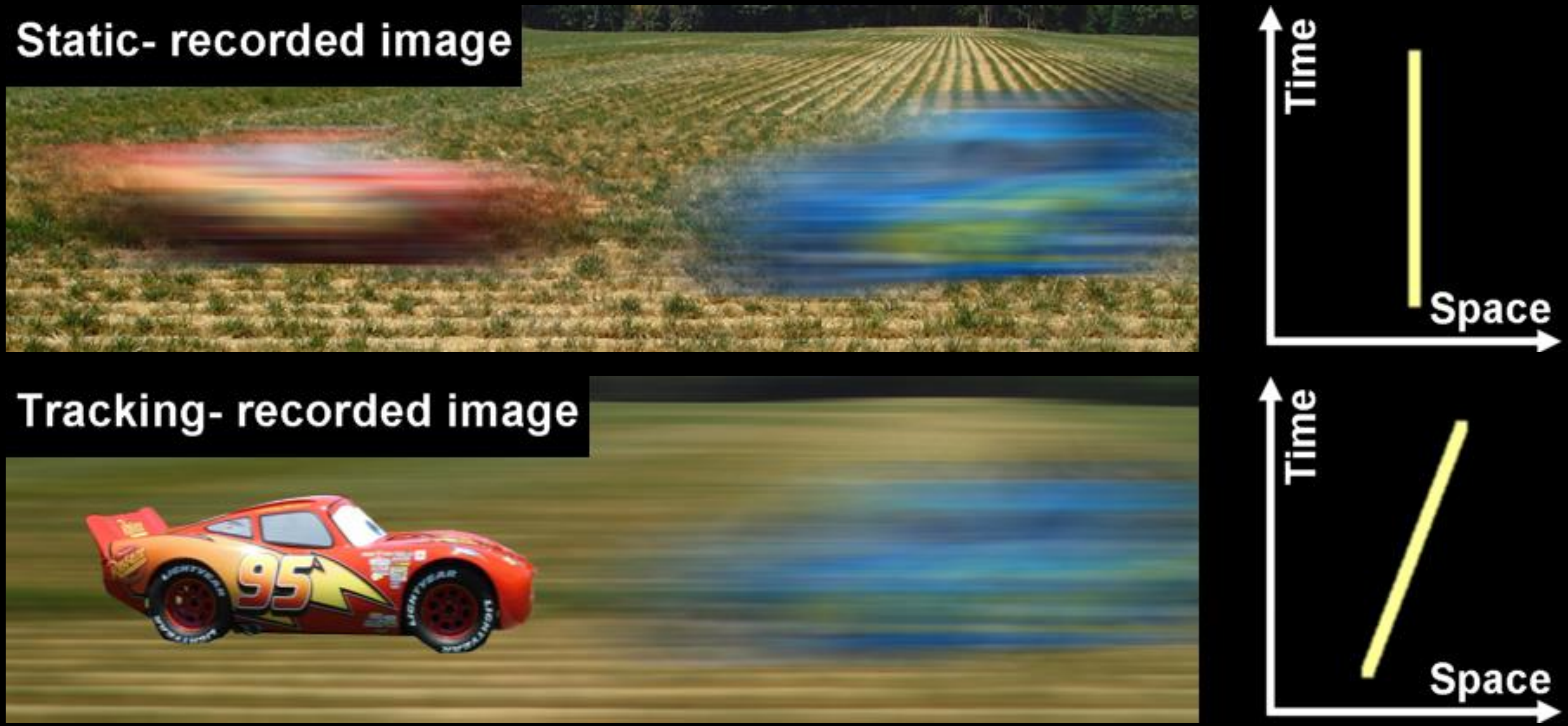


Removing motion blur is hard:

- **Need to know exact motion velocity (blur kernel)**
- **Need to segment image**

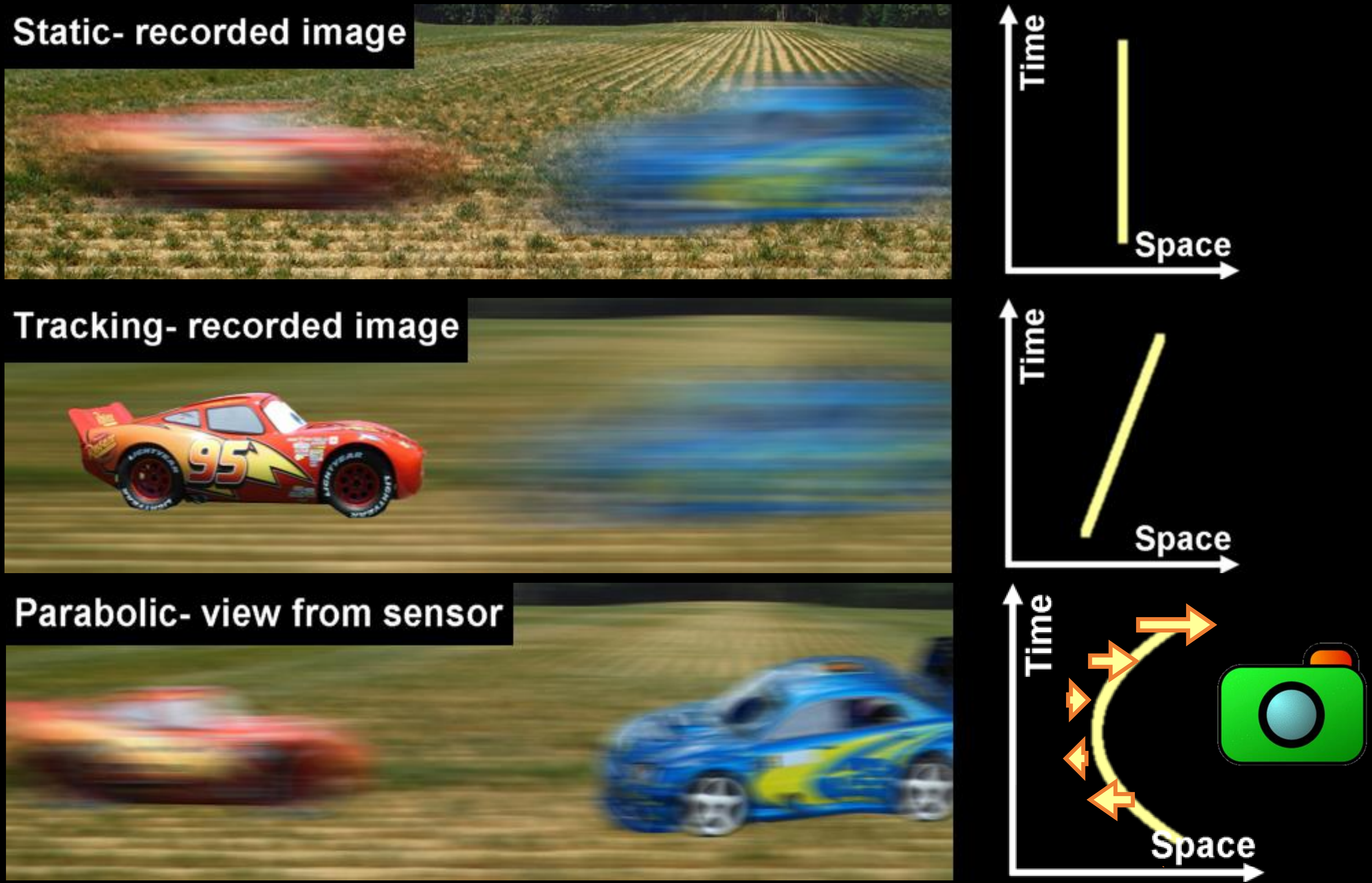
Overcoming motion blur

Levin et al. **Motion Invariant Photography** *SIGGRAPH, 2008.*



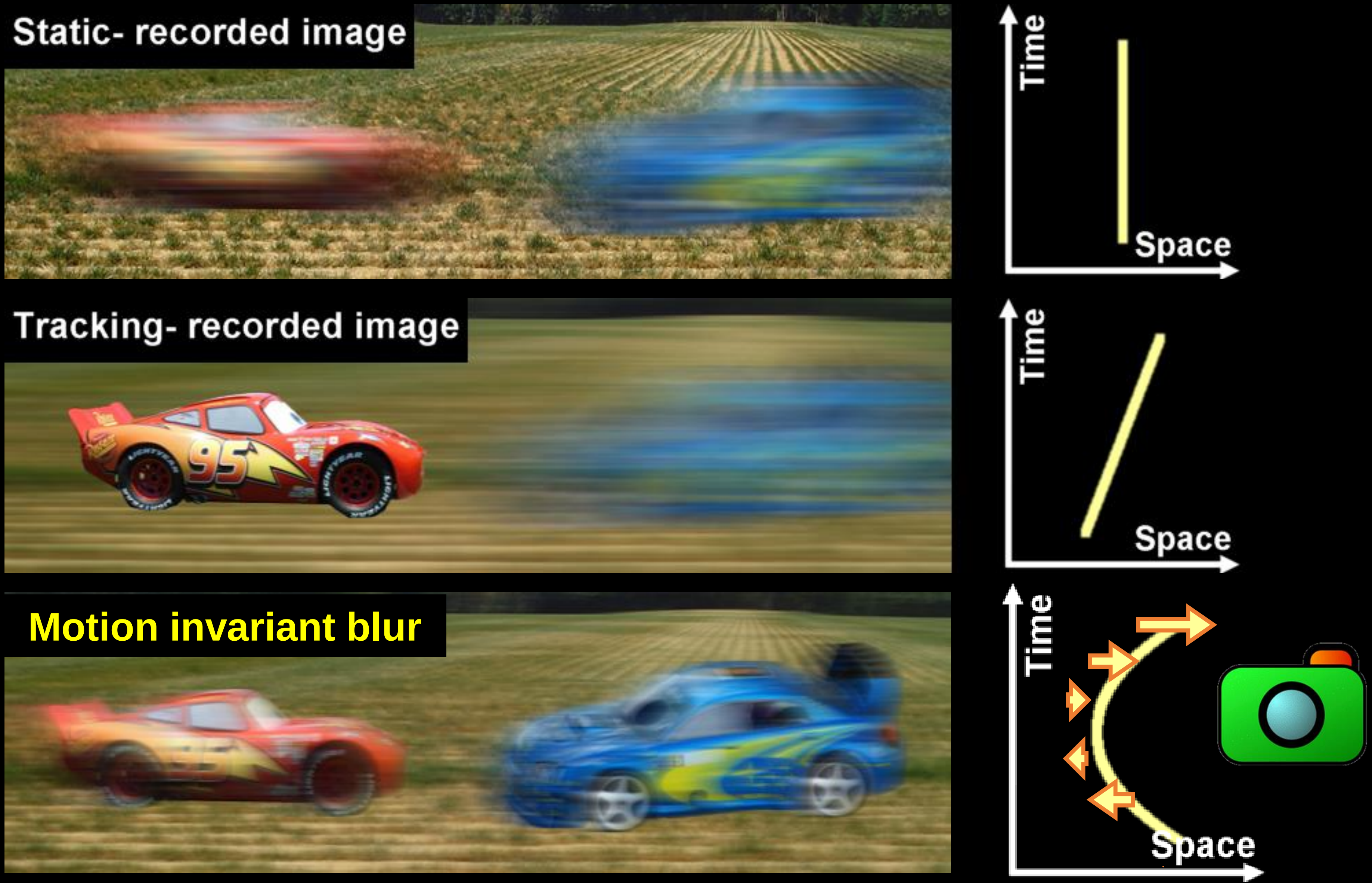
Overcoming motion blur

Levin et al. **Motion Invariant Photography** *SIGGRAPH, 2008.*



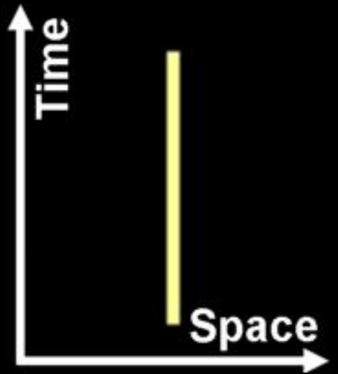
Overcoming motion blur

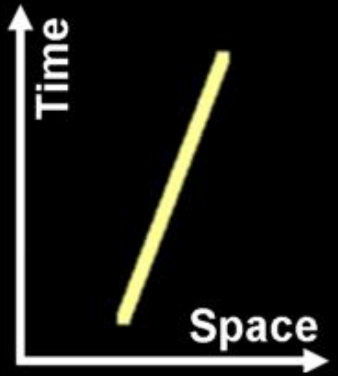
Levin et al. **Motion Invariant Photography** *SIGGRAPH, 2008.*

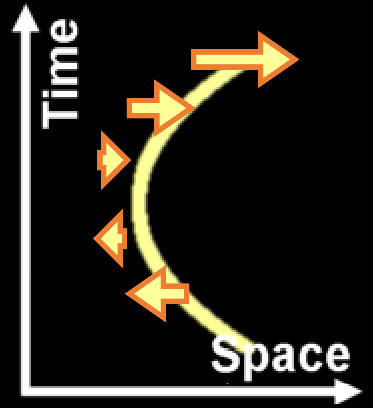


Motion Invariant Photography

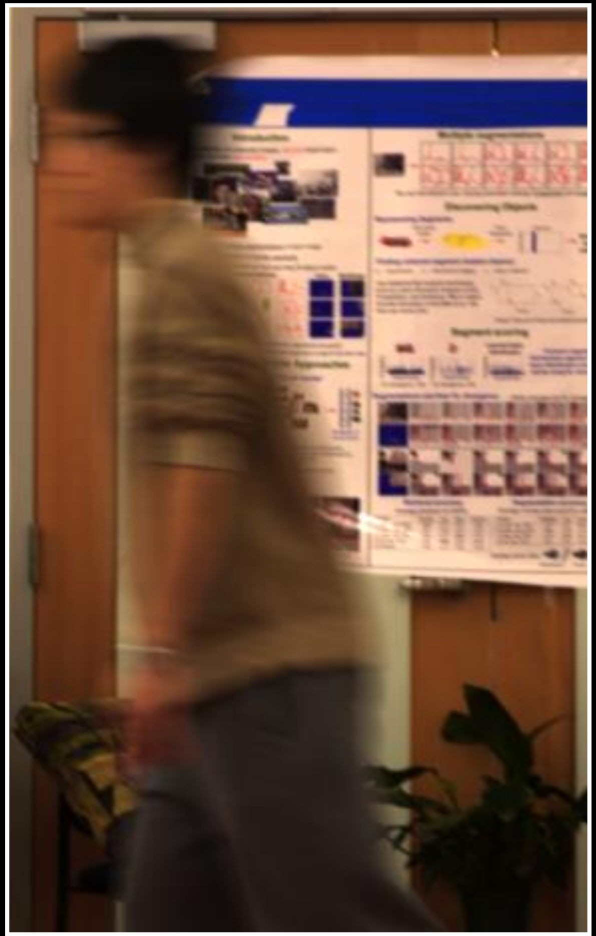
Levin et al. **Motion Invariant Photography** *SIGGRAPH, 2008.*







Levin et al. **Motion Invariant Photography** SIGGRAPH, 2008.



Static camera

Unknown and variable blur

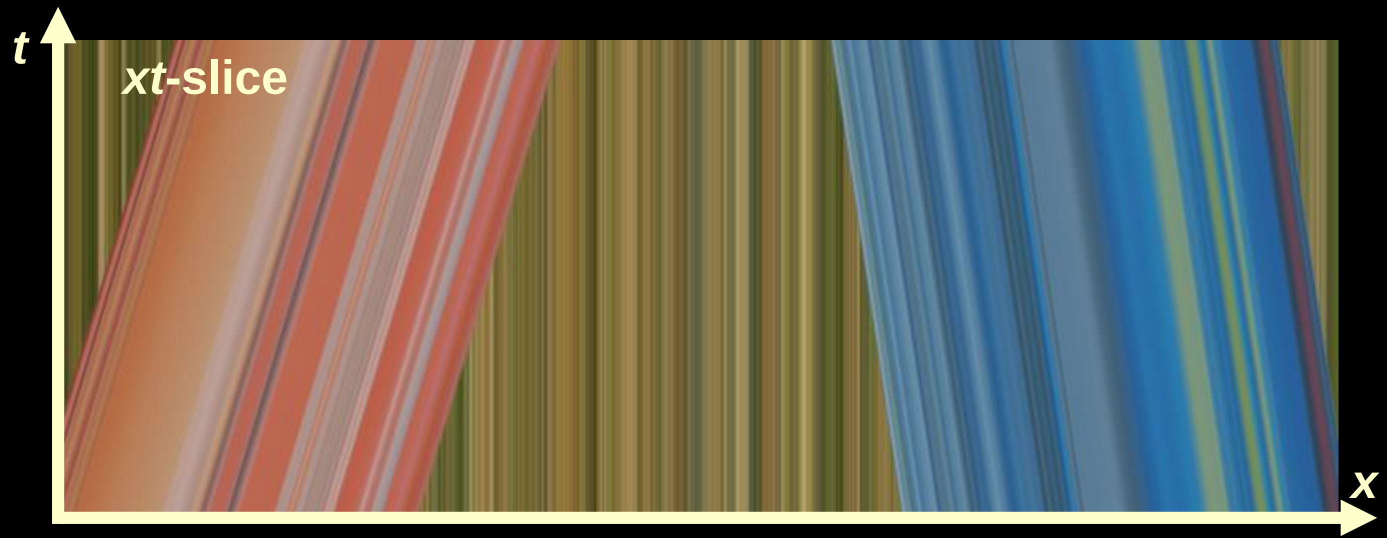
Our parabolic input

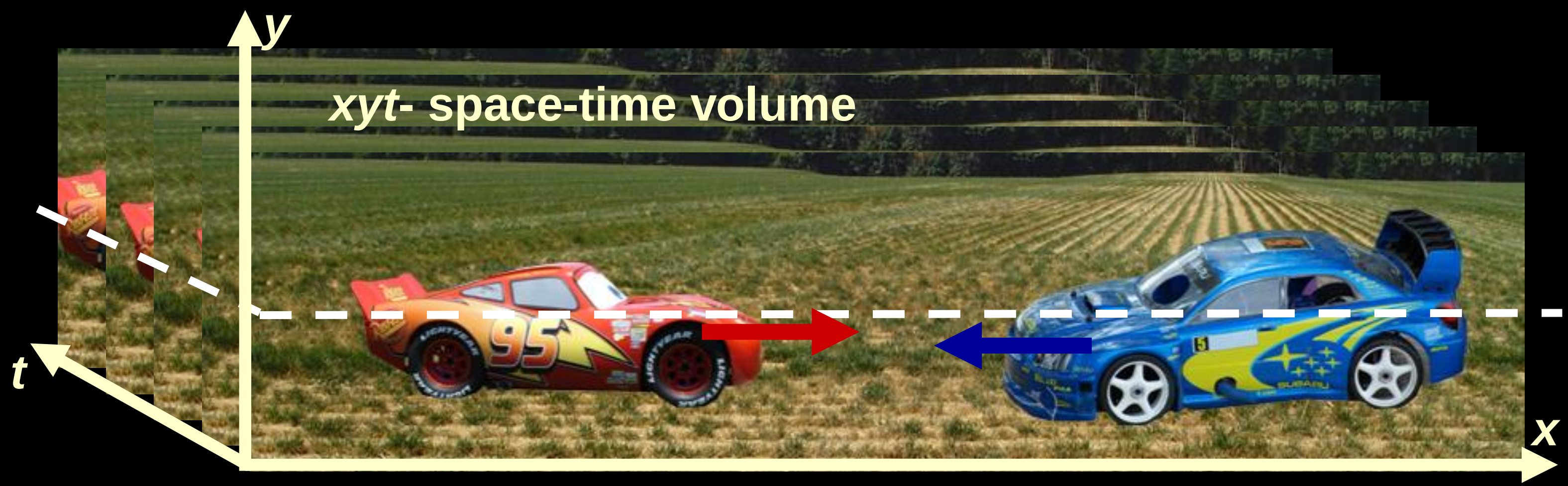
Blur invariant to velocity

Our output after deblurring

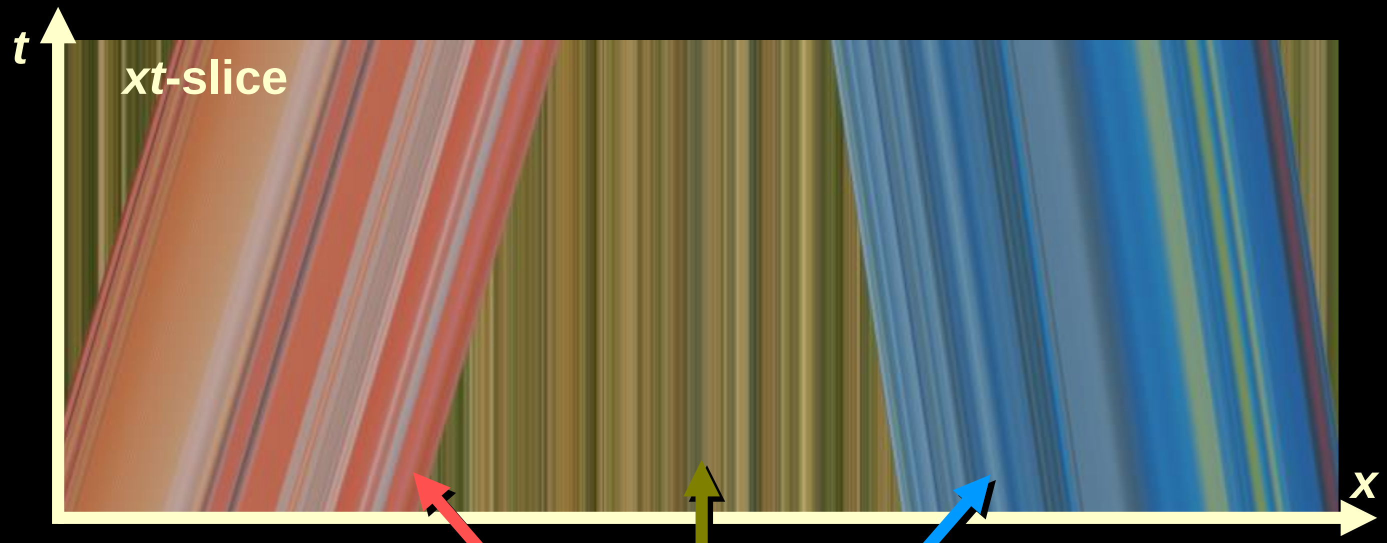
NON-BLIND deblurring

The space time volume





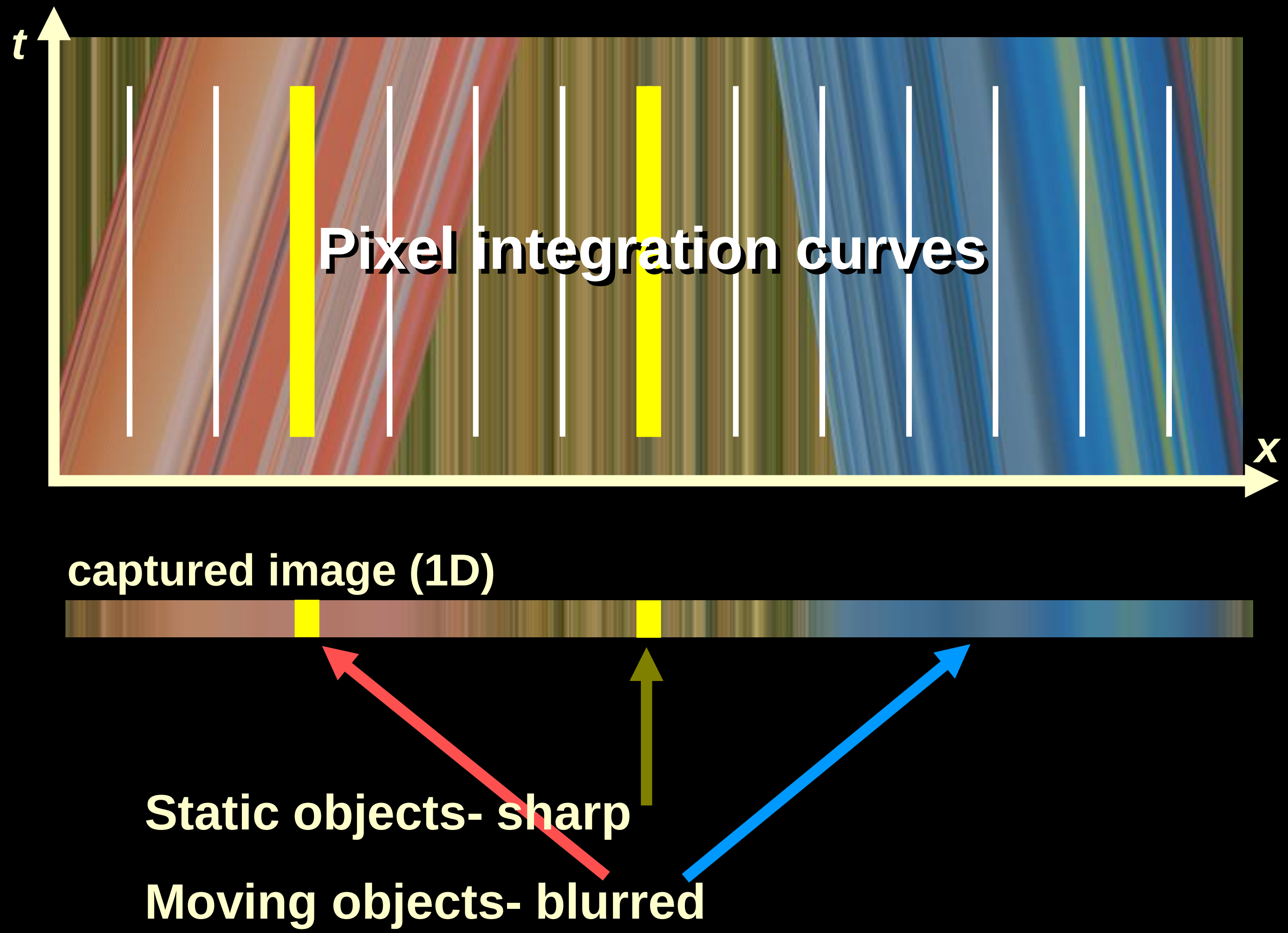
The space time volume



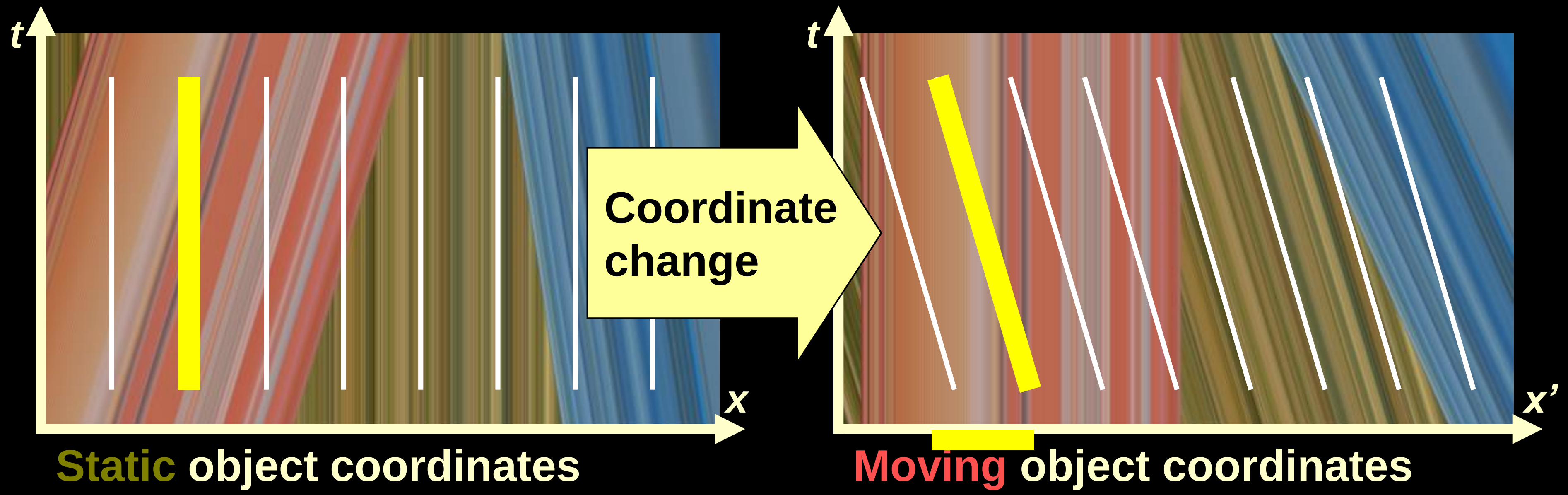
Static objects- vertical lines

Moving objects slanted lines, slope ~ motion velocity

Camera integration

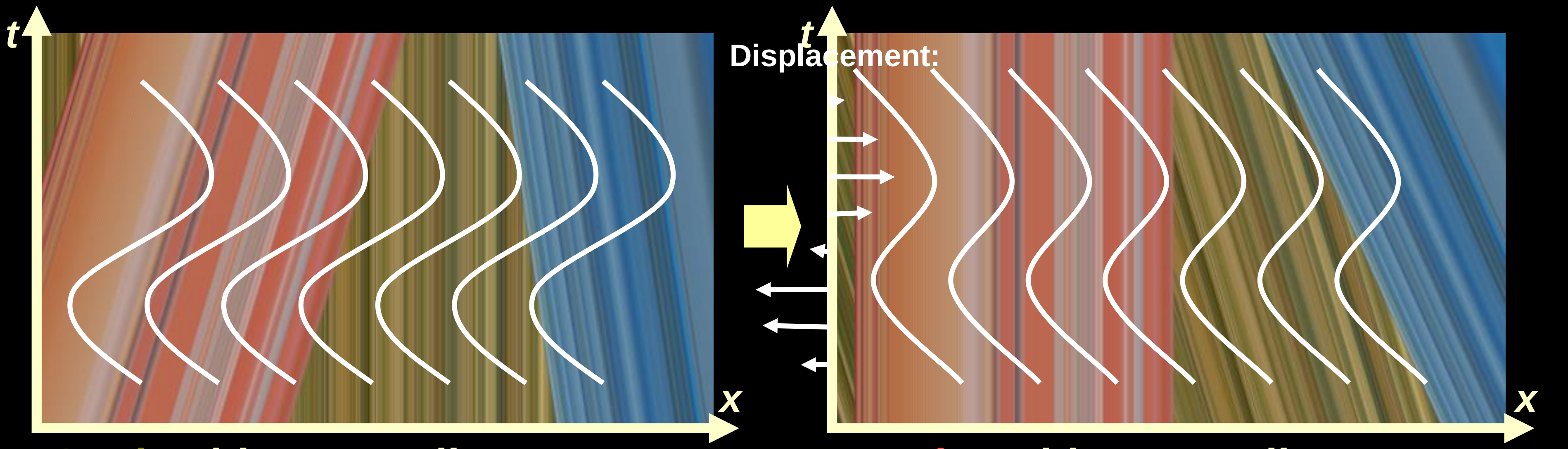


Shearing



Shearing: $(x,t) \rightarrow (x-st,t)$

Shearing



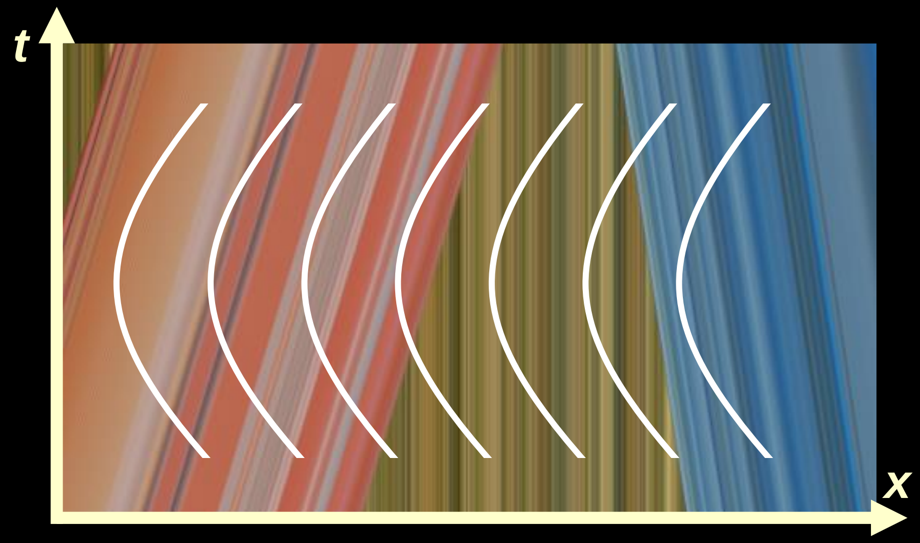
Static object coordinates | Moving object coordinates

Shearing: $(x, t) \rightarrow (x - st, t)$

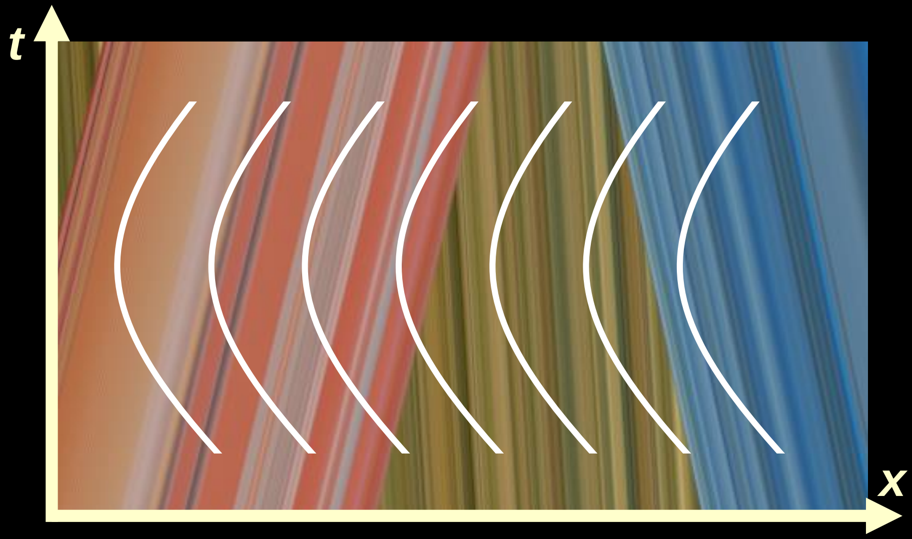
Can we find a shear invariant integration curve?

Solution: parabolic curve!

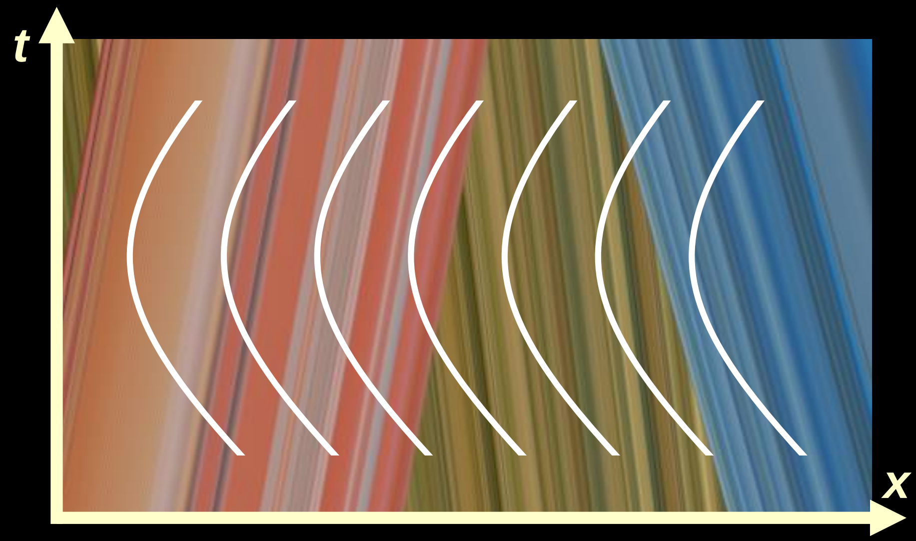
Solution: parabolic curve - shear invariant



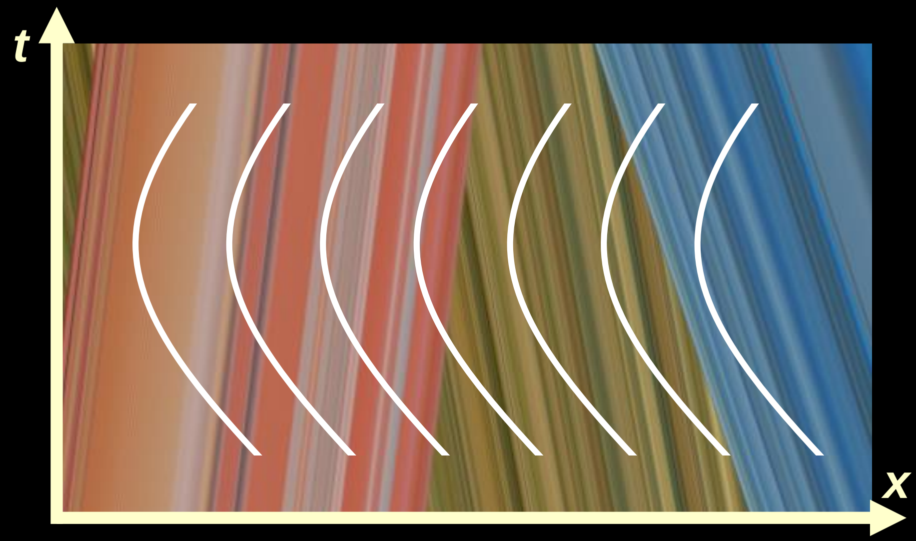
Solution: parabolic curve - shear invariant



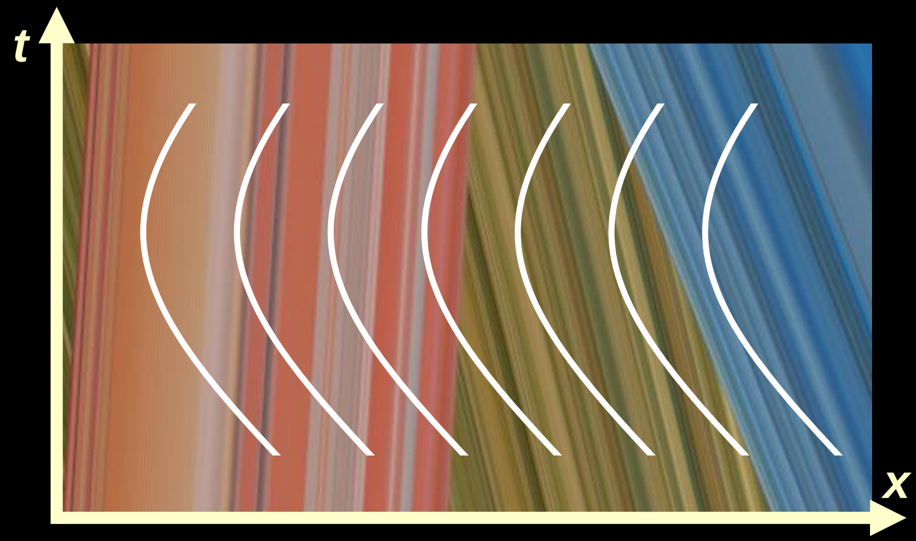
Solution: parabolic curve - shear invariant



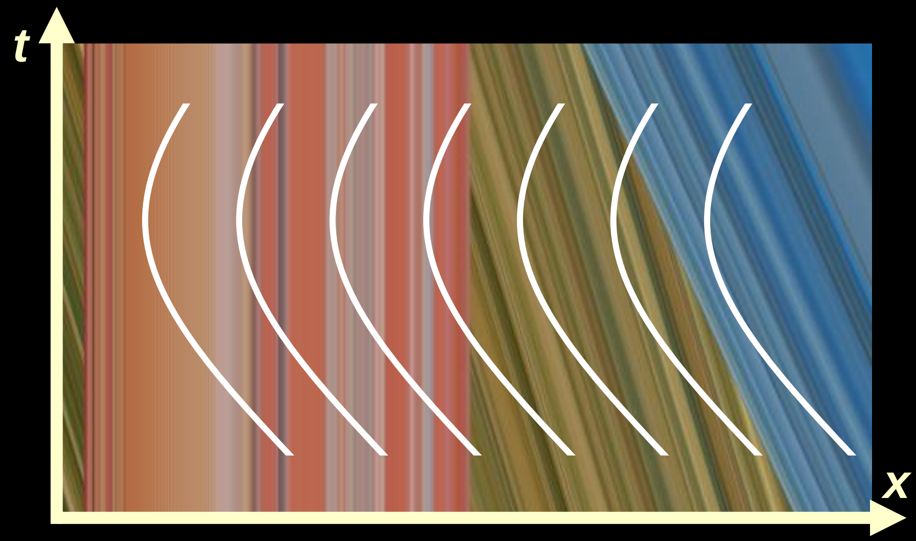
Solution: parabolic curve - shear invariant



Solution: parabolic curve - shear invariant



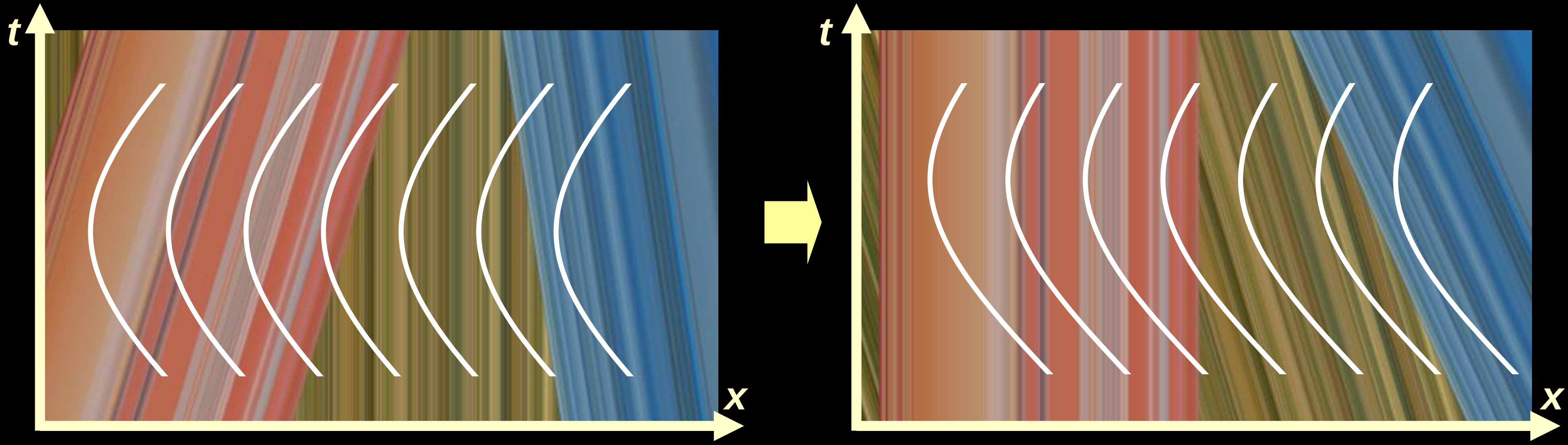
Solution: parabolic curve - shear invariant



Solution: parabolic curve - shear invariant

Static object coordinates

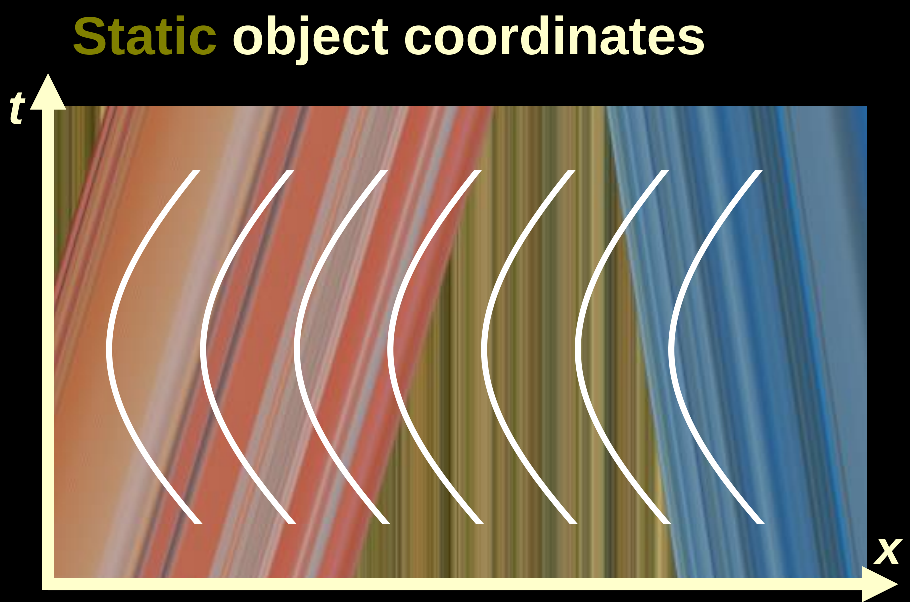
Moving object coordinates

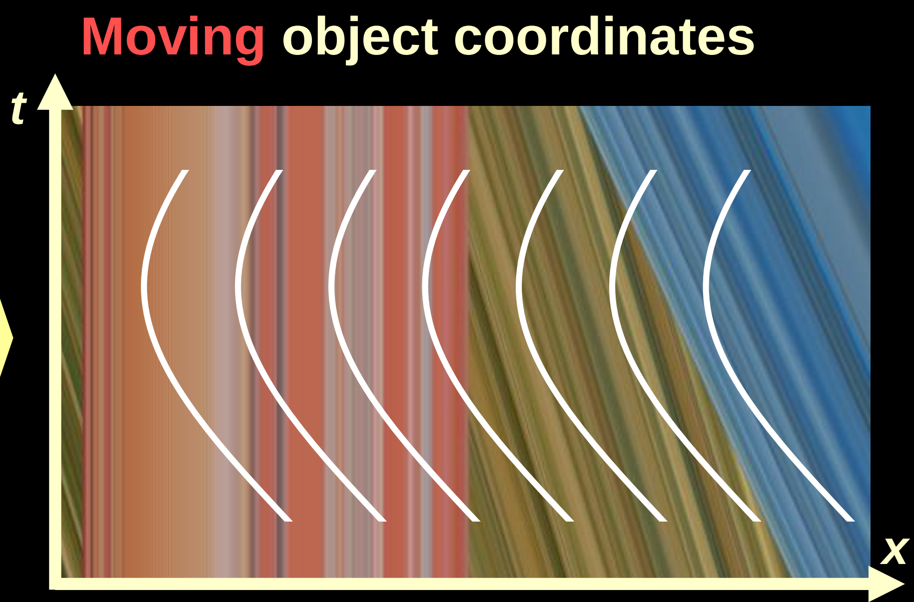


Shearing: $(x,t) \rightarrow (x - st, t)$

Sheared parabola → **Shifted parabola**

Solution: parabolic curve - shear invariant





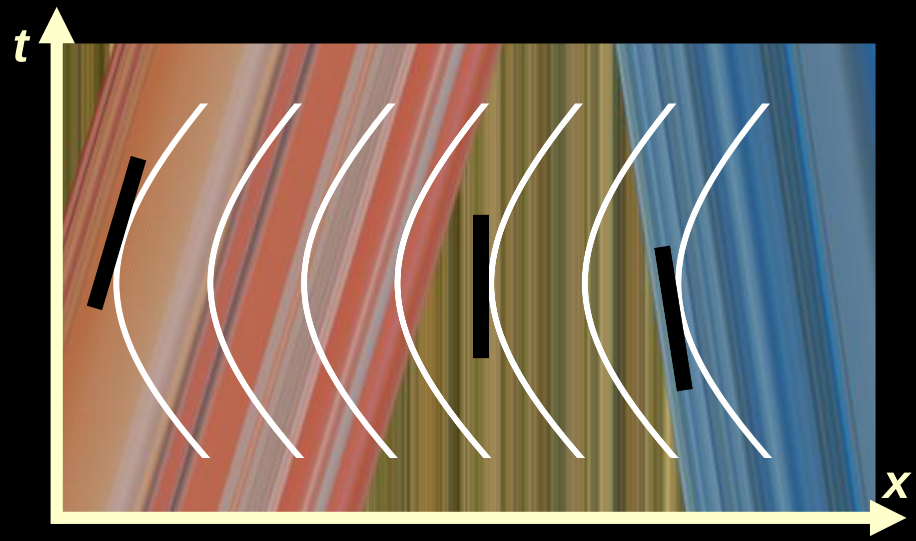
Shearing: $(x,t) \to (x - st, t)$

$$f(t) = t^2$$

$$f_s(t) = t^2 - st$$
$$= \left(t - \underbrace{s/2}_{b}\right)^2 - \underbrace{s^2/4}_{c}$$

Sheared parabola ➡ **Shifted parabola**

Solution: parabolic curve - shear invariant



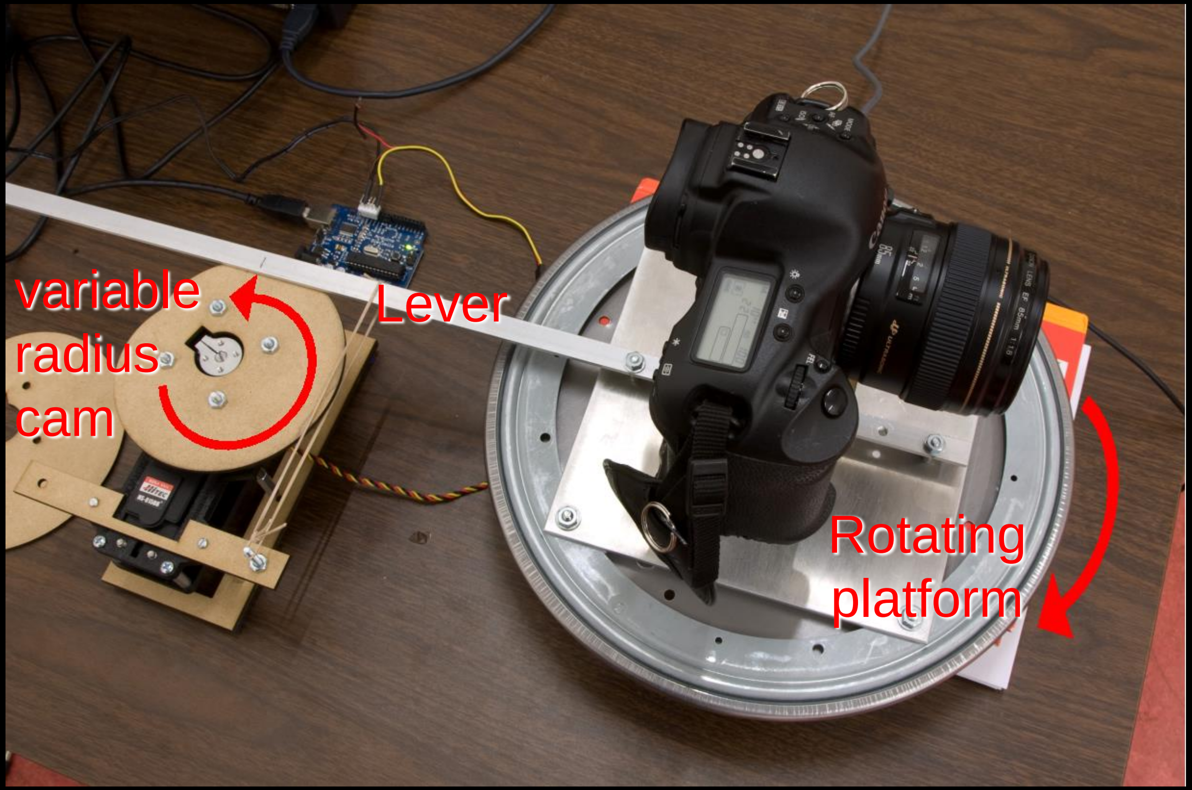
For any velocity (slope),

- there is one time instant where curve is tangent
- corresponds to moment when object is tracked.

• The parabola has a linear derivative
=> spends equal time tracking each velocity.

Hardware construction

- **Ideally move sensor**
  (requires same hardware as existing stabilization systems)
- **In prototype implementation: rotate camera**



Human motion- no perfect linearity

Input from a static camera

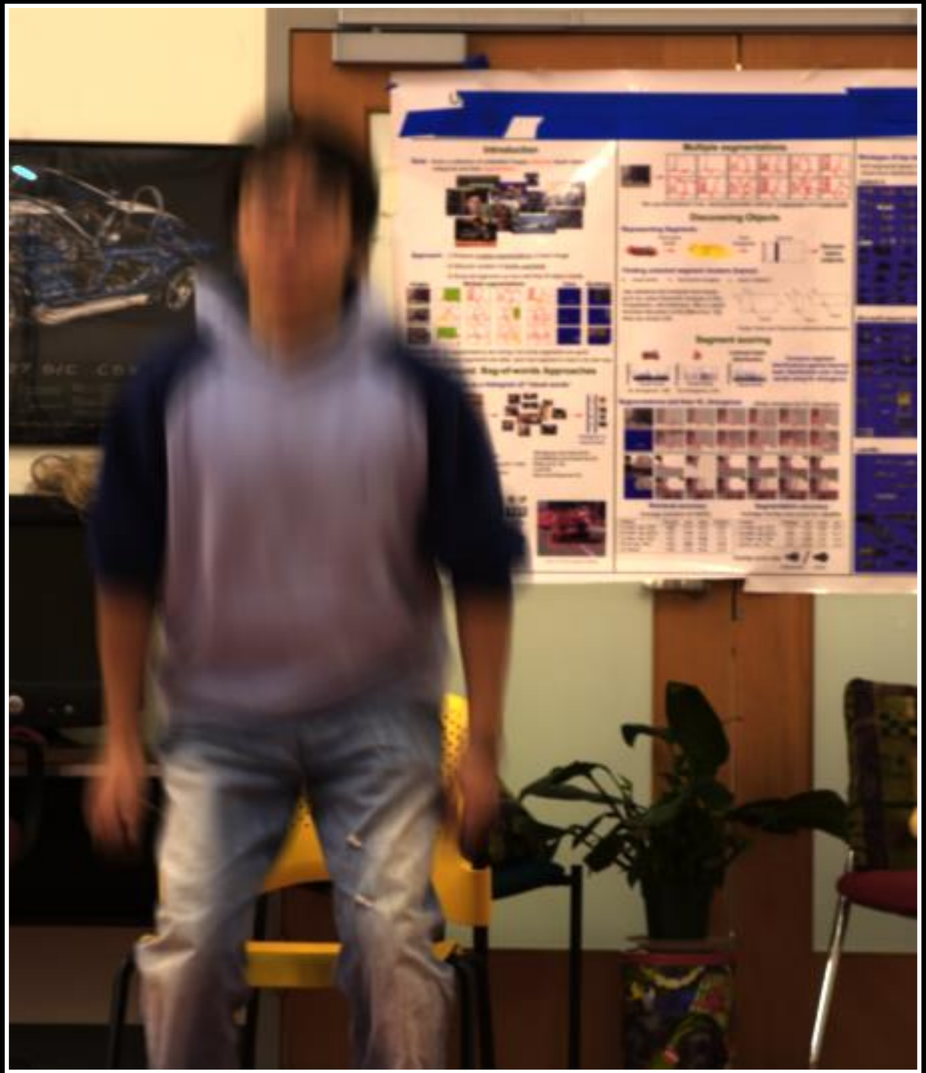
Deblurred output from our camera

Violating 1D motion assumption- forward motion

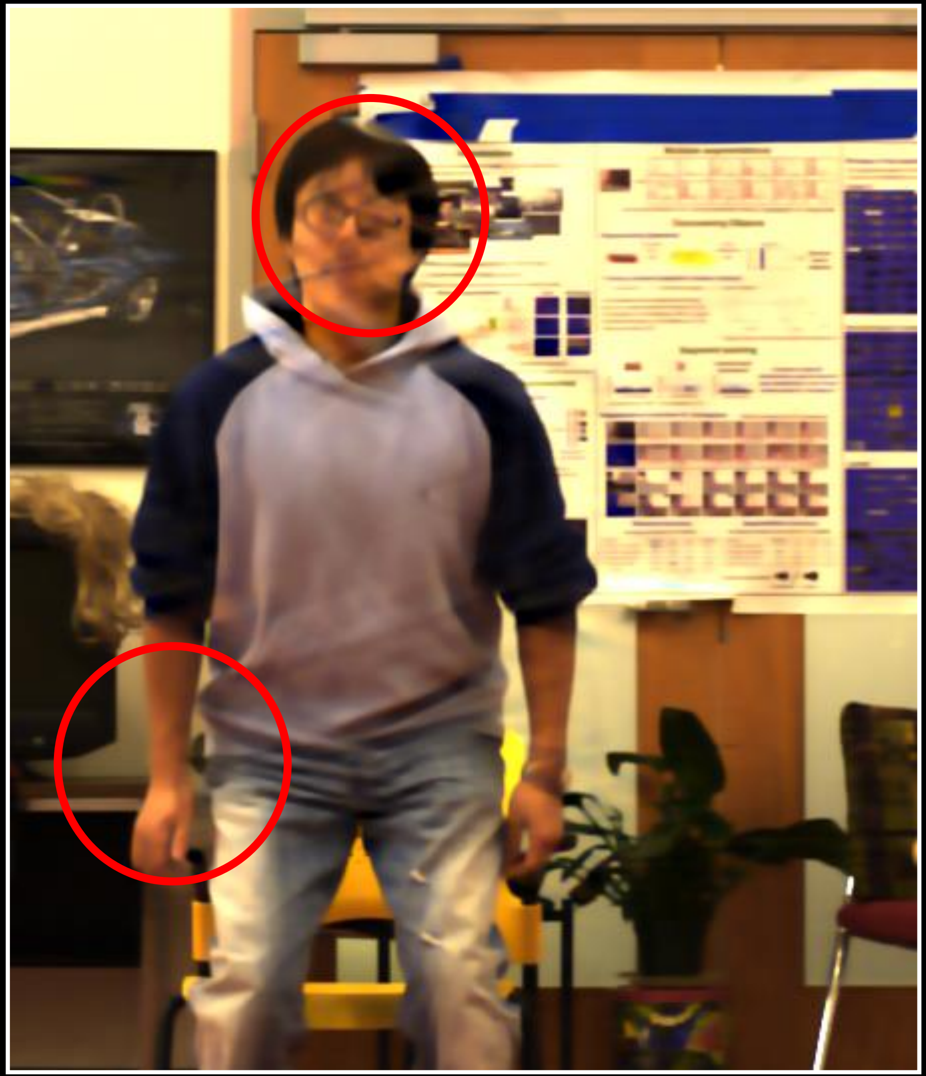
Input from a static camera

Deblurred output from our camera

Violating 1D motion assumption- stand-up motion



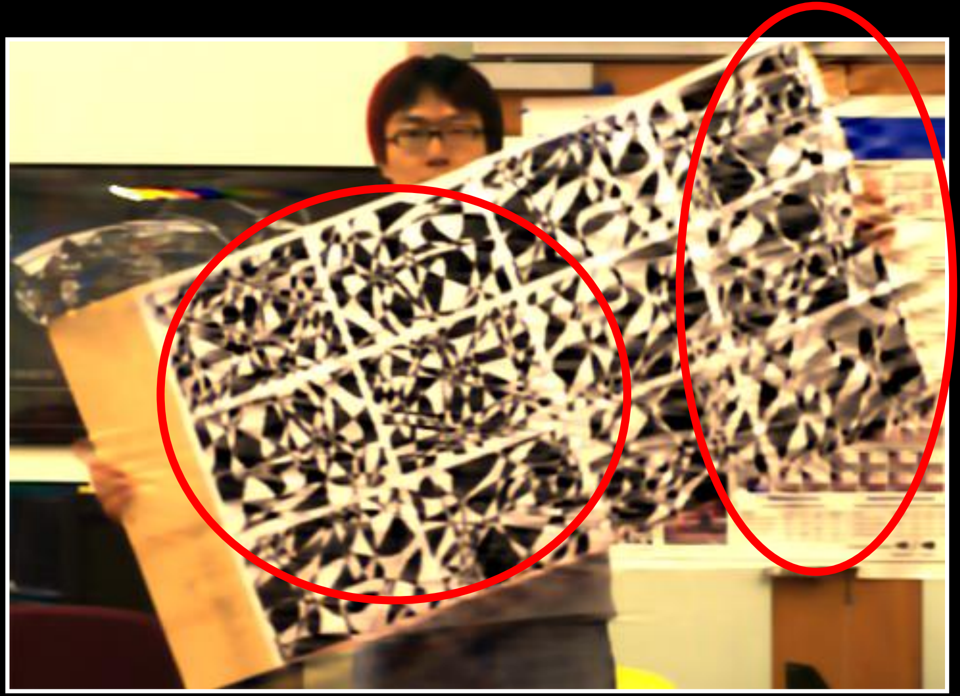
Input from a static camera



Deblurred output from our camera

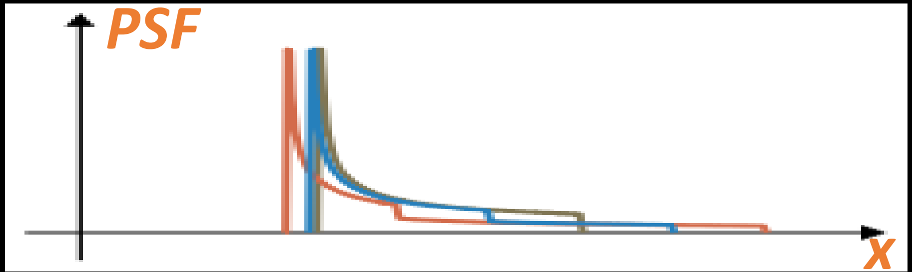
Violating 1D motion assumption- rotation

Input from a static camera



Deblurred output from our camera

Limitations & approximations



Limitations:

- 1-D velocity
- Pre-defined velocities range

Approximations:

- PSFs differs in boundaries for different velocities
- Deblurred objects captured at different times

Uniqueness & optimality

- Uniqueness – Parabola is ***the only*** shear invariant curve
- Optimality – Most stable inversion of PSF: $\widehat{psf(w)}^{-1}$ is the highest you can get, ***provably.***

$$y = psf * x$$

$$\Big\downarrow \mathcal{F}$$

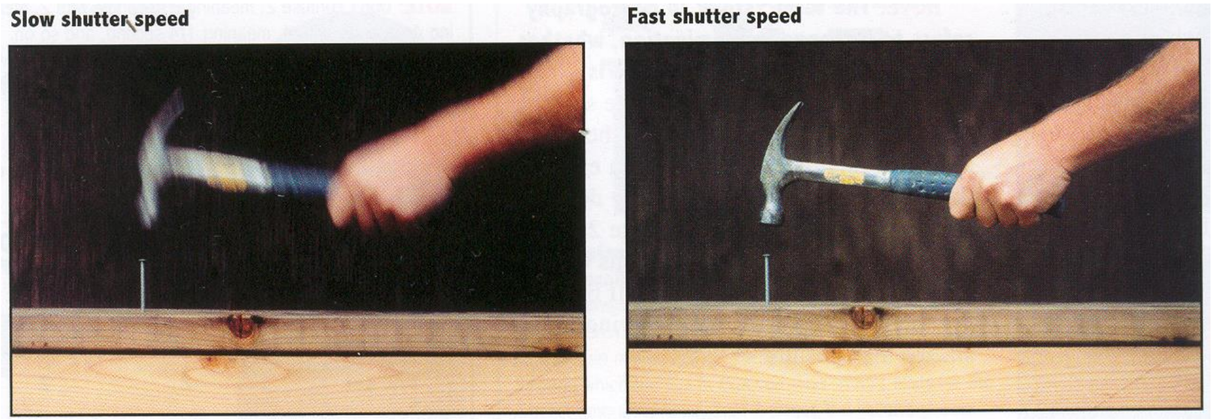
$$\hat{y} = \widehat{psf} \cdot \hat{x}$$

$$\boxed{\Big\downarrow \blacksquare^{-1}}$$

$$\widehat{psf}^{-1} \cdot \hat{y} = \hat{x}$$

Computational photography approaches to blurring problems

- Motion blur
  - Flutter shutter
  - Motion invariant photography



- Defocus blur
  - Coded aperture
  - Lattice focal
  - Flexible depth of field
  - Wavefront coding

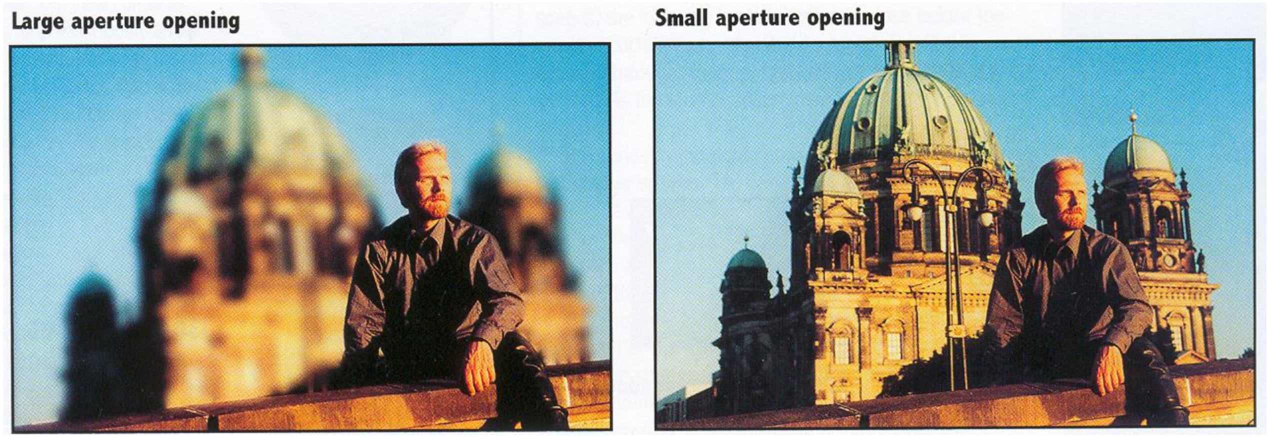
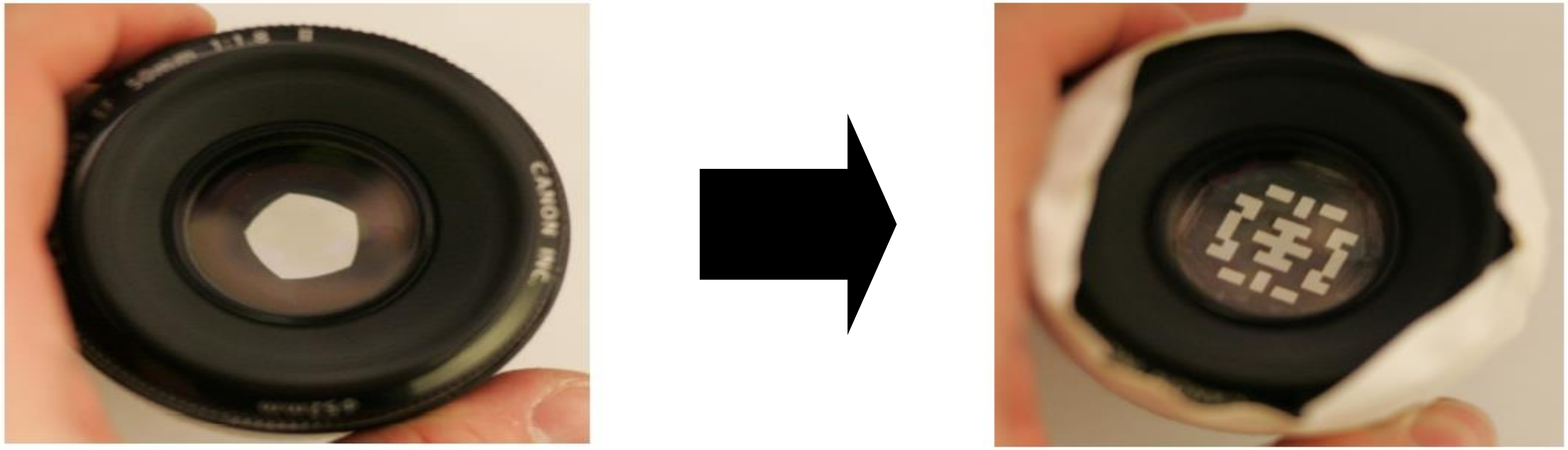


Image and Depth from a Conventional Camera with a Coded Aperture

Anat Levin, Rob Fergus,
Frédo Durand, William Freeman



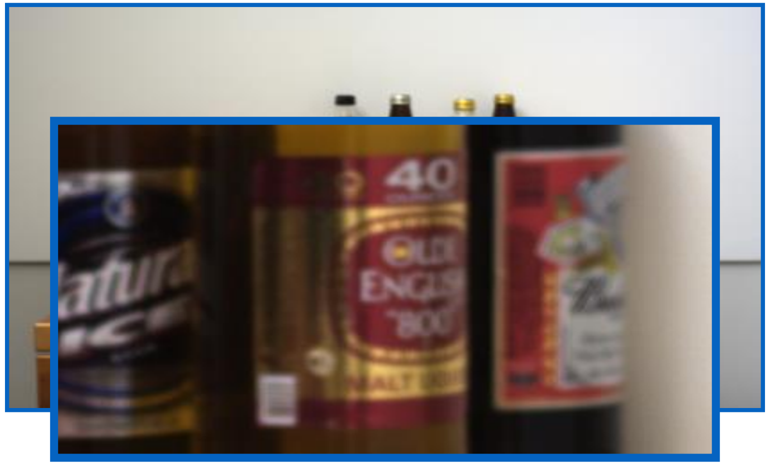
Coded aperture - Introduction

Problem:

Objects that are not in focus seem blurry.

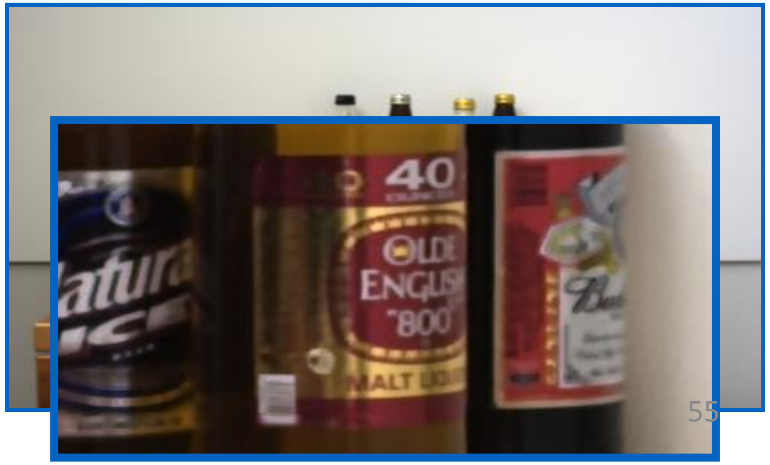
Goal:

Single input image:



Output #1: Depth map

Output #2: All-focused image



55

Defocus blur

Lens' aperture

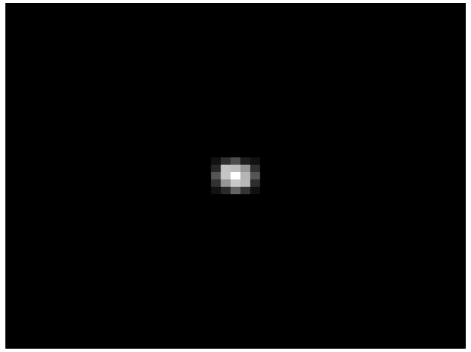
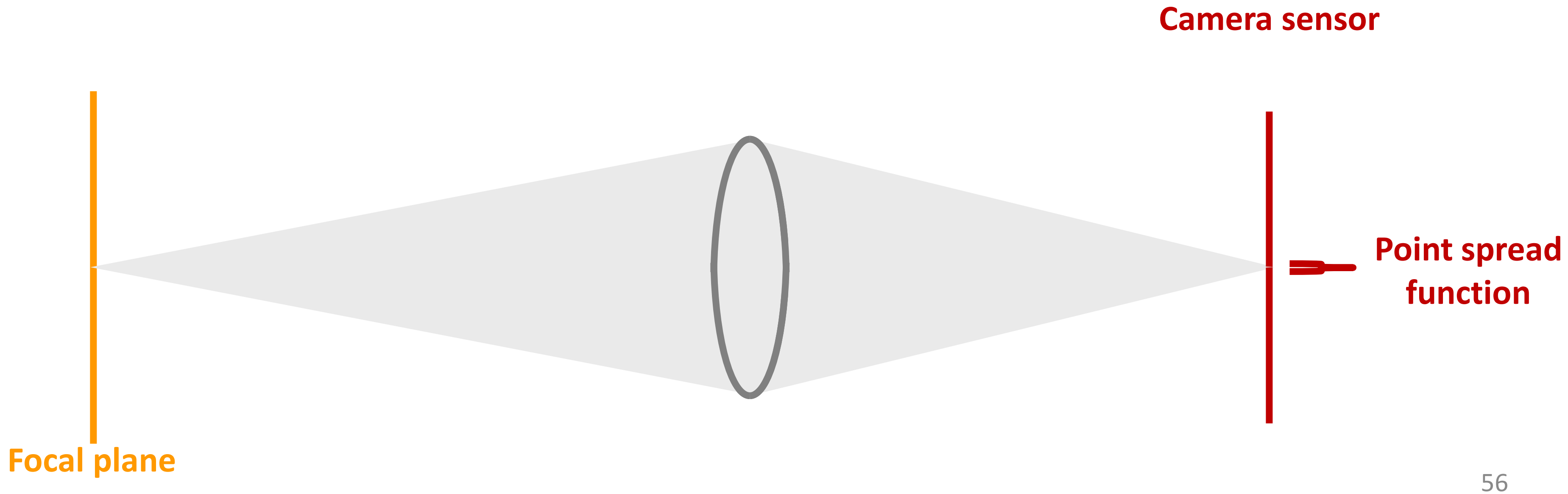


Image of a point light source



Defocus blur

Lens’ aperture

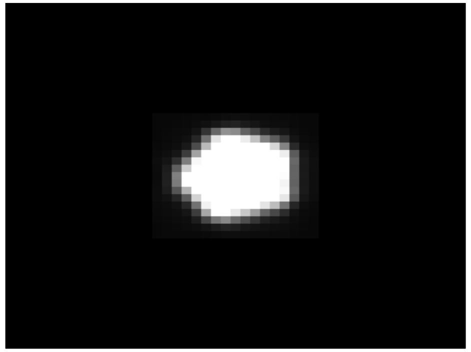
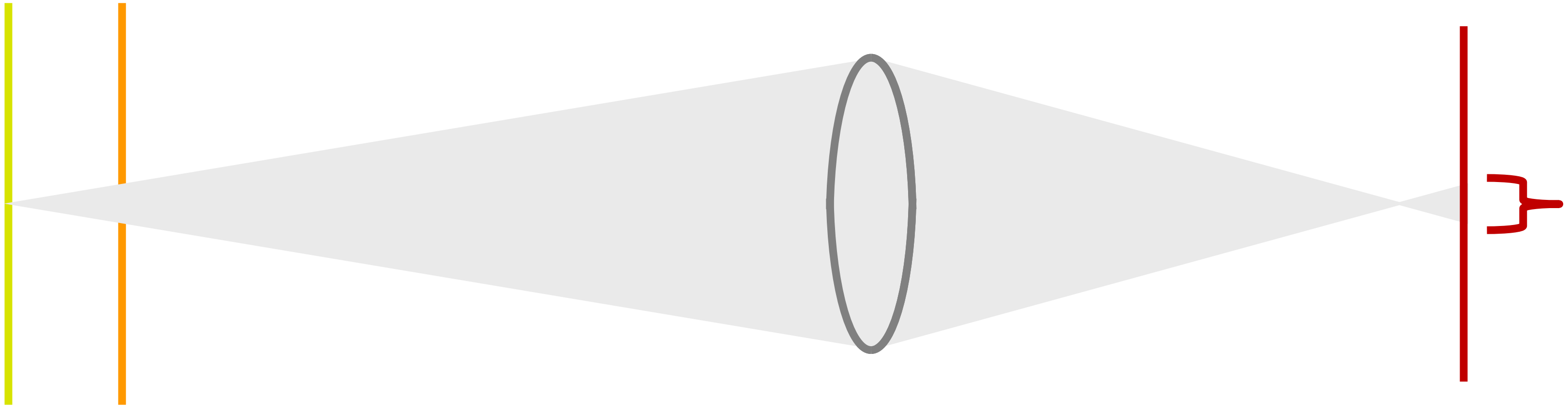


Image of a defocused point light source



Defocus blur

Lens' aperture

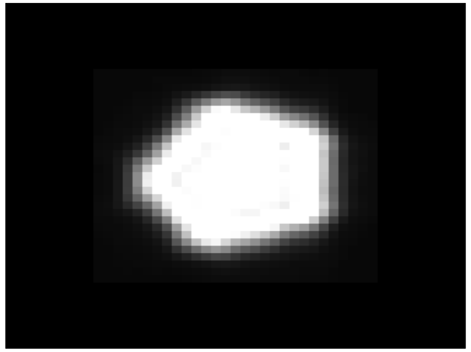
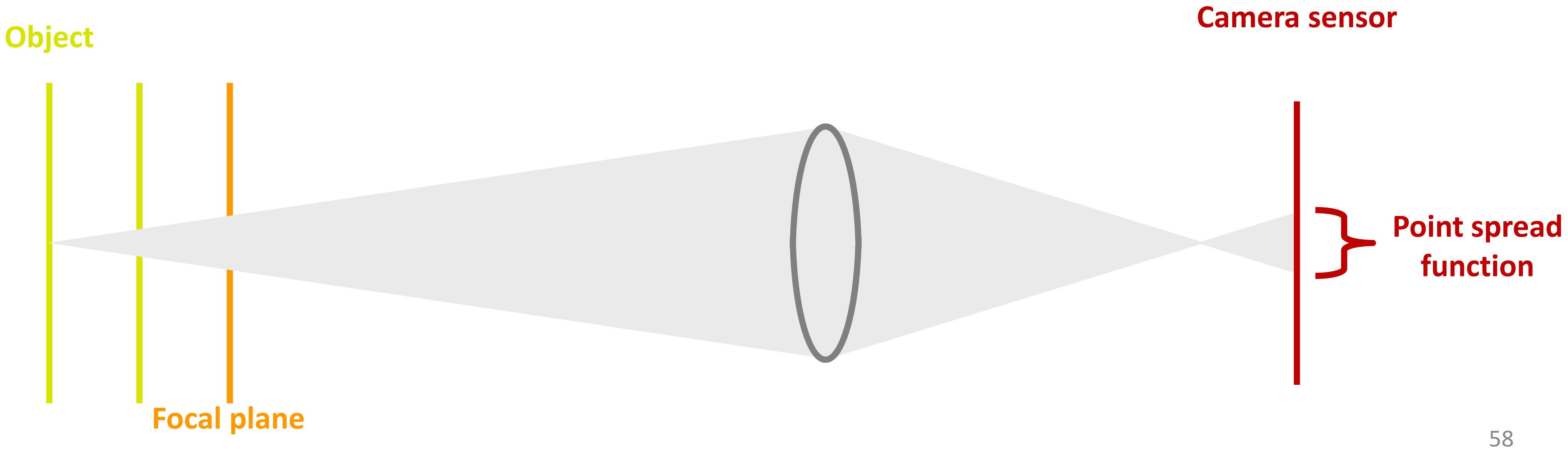


Image of a defocused point light source



Defocus blur

Lens' aperture

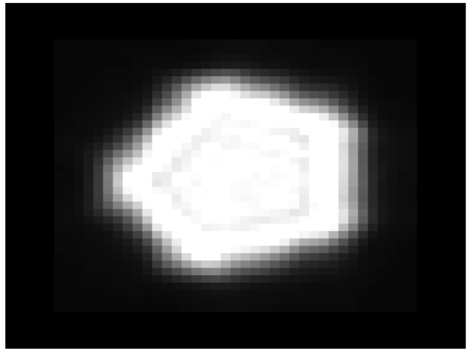
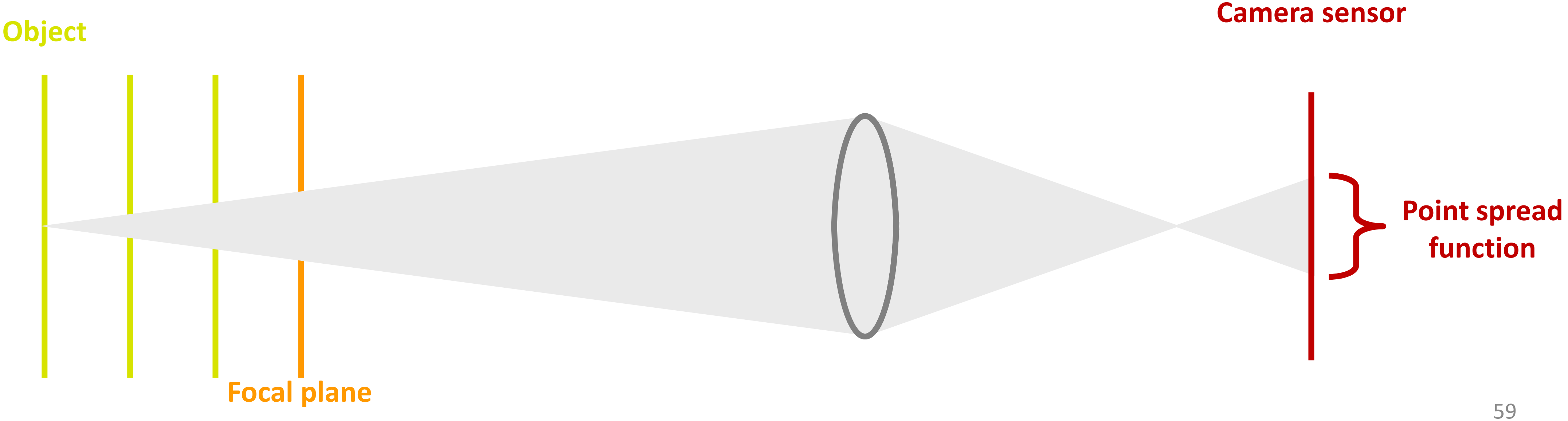


Image of a defocused point light source



Defocus blur

Lens' aperture

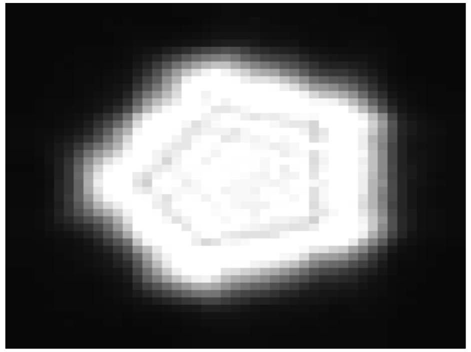
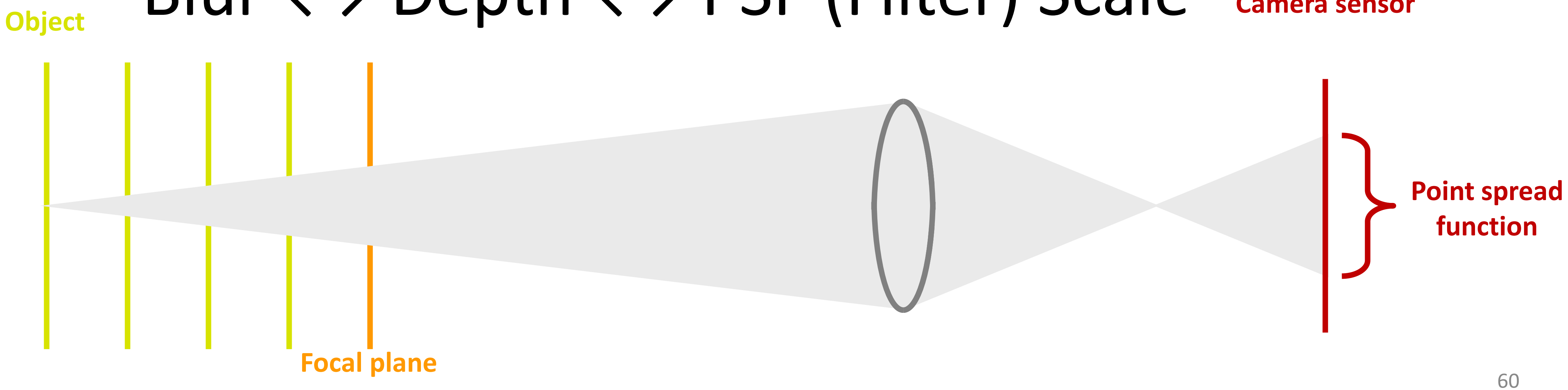
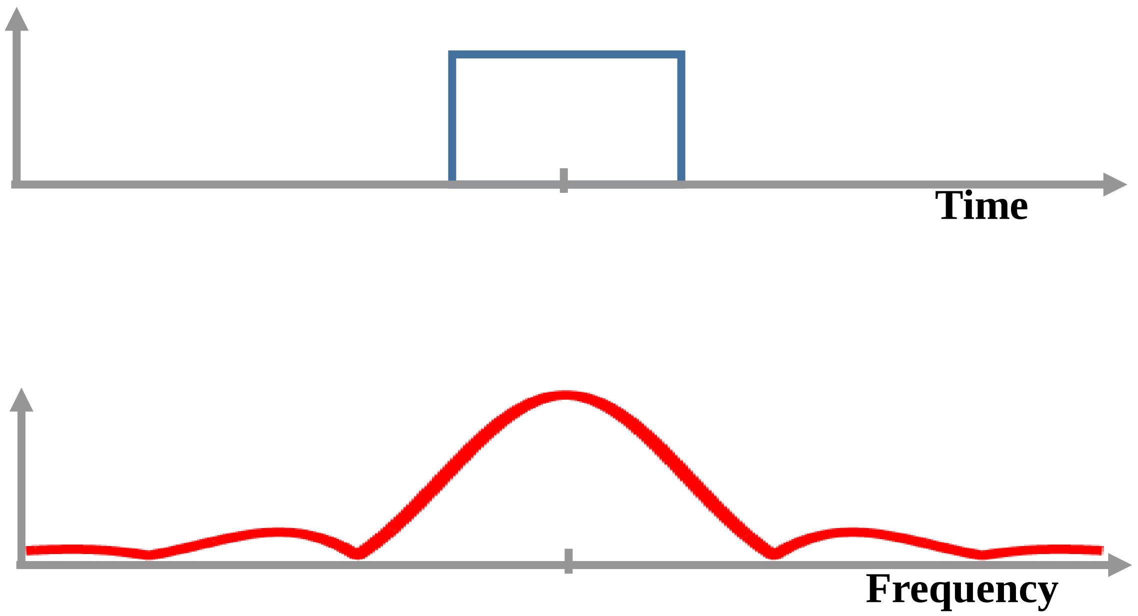


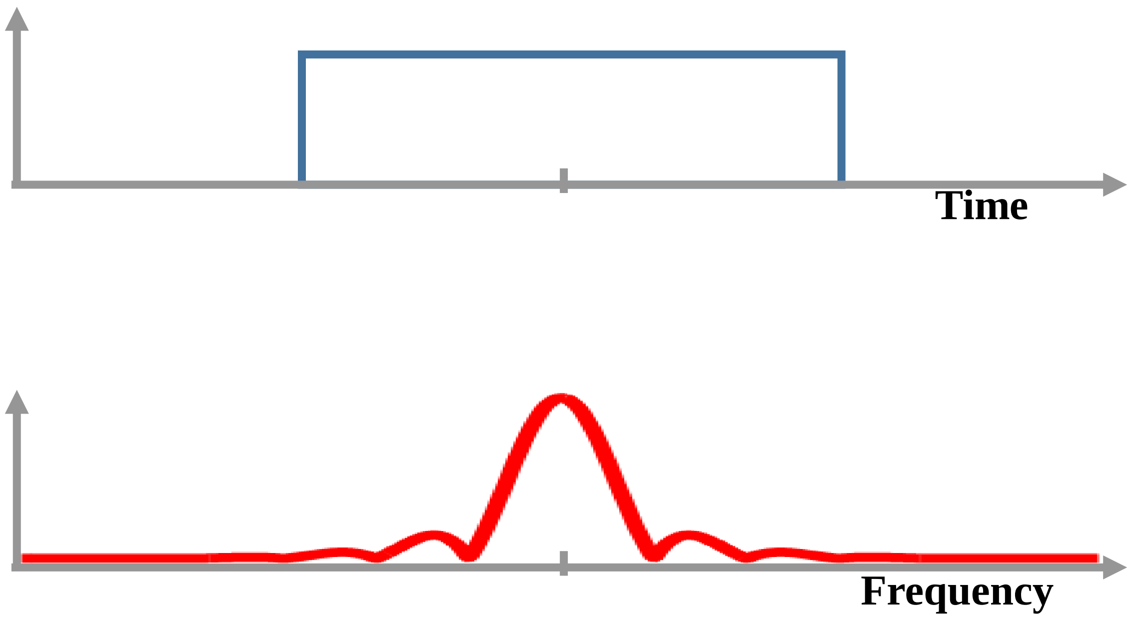
Image of a defocused point light source

Blur ↔ Depth ↔ PSF (Filter) Scale



1-D Frequency analysis





Larger filter scale

$\Longrightarrow$ **Loss of high frequencies**

$\Longrightarrow$ **Reconstruction is difficult**

Main Challenges

1. Depth discrimination

A smooth scene or defocus blur?

Lack or loss of high frequencies?

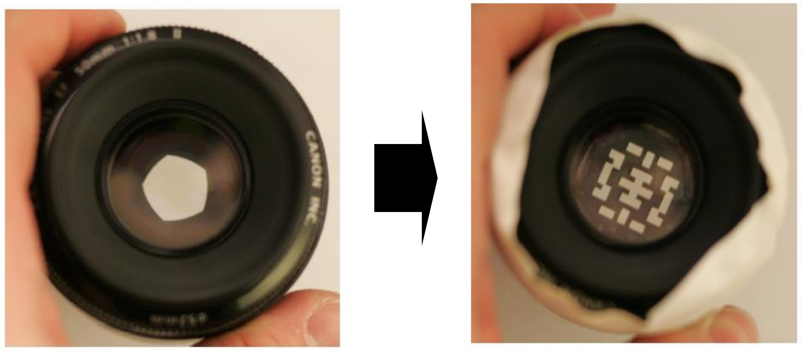
?

Out of focus

2. Loss of high frequencies $\Longrightarrow$ Reconstruction is difficult.

Coded aperture

- Mask within the aperture of the lens



- Defocus patterns differ from natural images
  => Easier depth discrimination

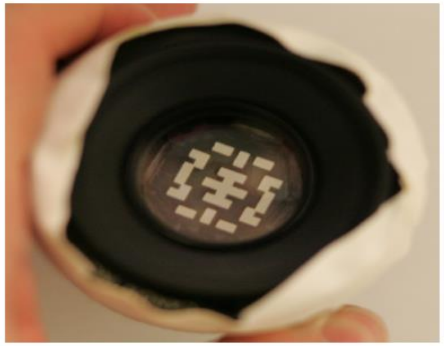
- Defocus kernel preserves more high frequencies (not a LPF)

Build your own coded aperture

Voila!

Coded aperture: lens with occluder

Aperture pattern



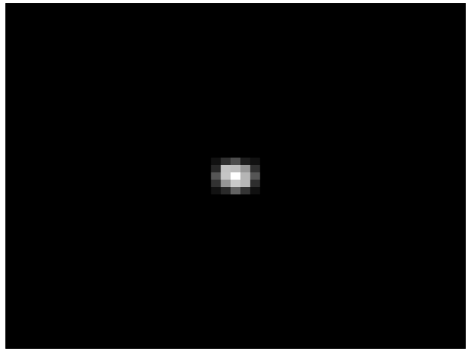
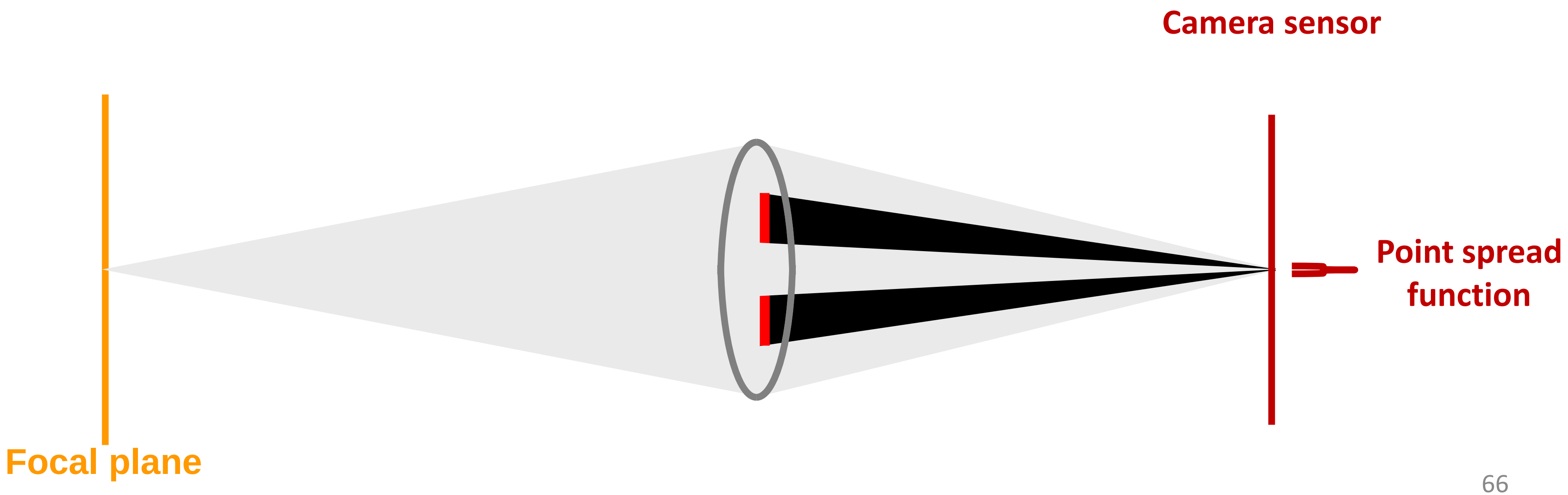
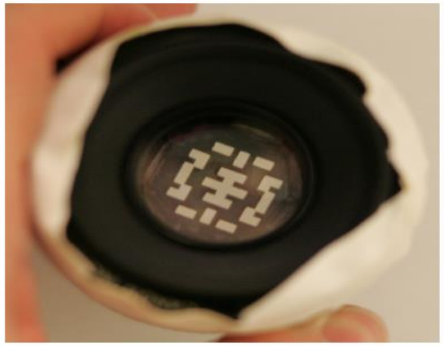


Image of a point light source



Coded aperture: lens with occluder

Aperture pattern



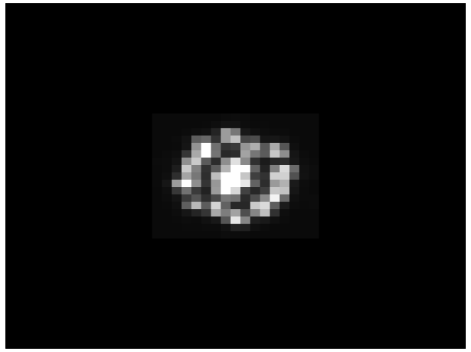
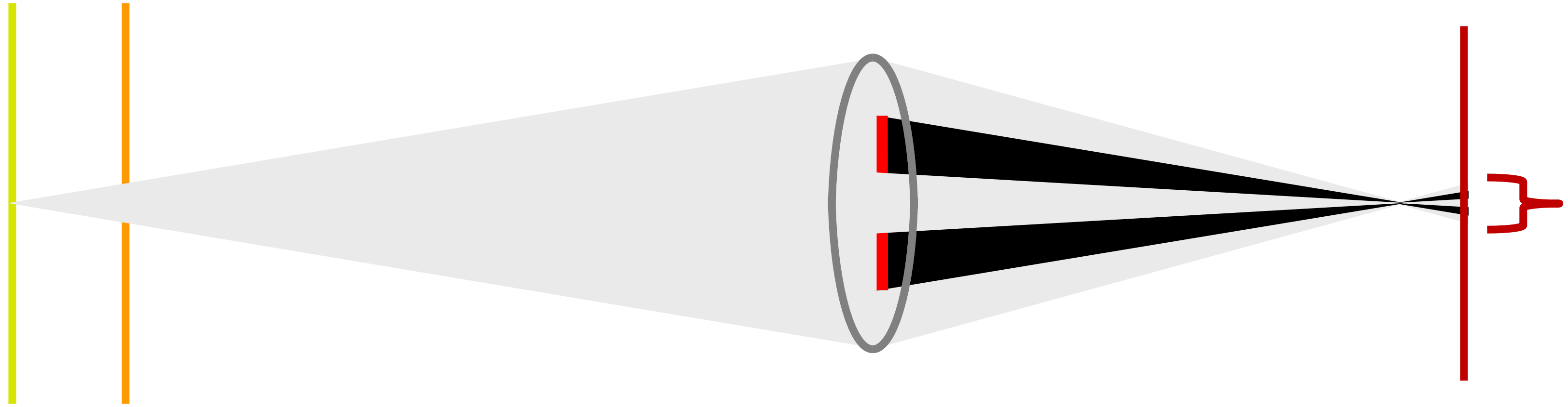
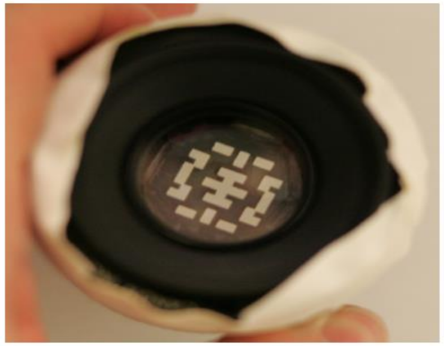


Image of a defocused point light source



Coded aperture: lens with occluder

Aperture pattern



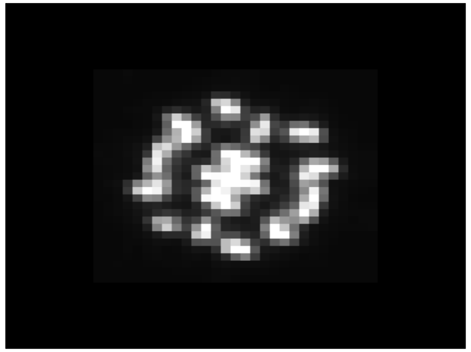
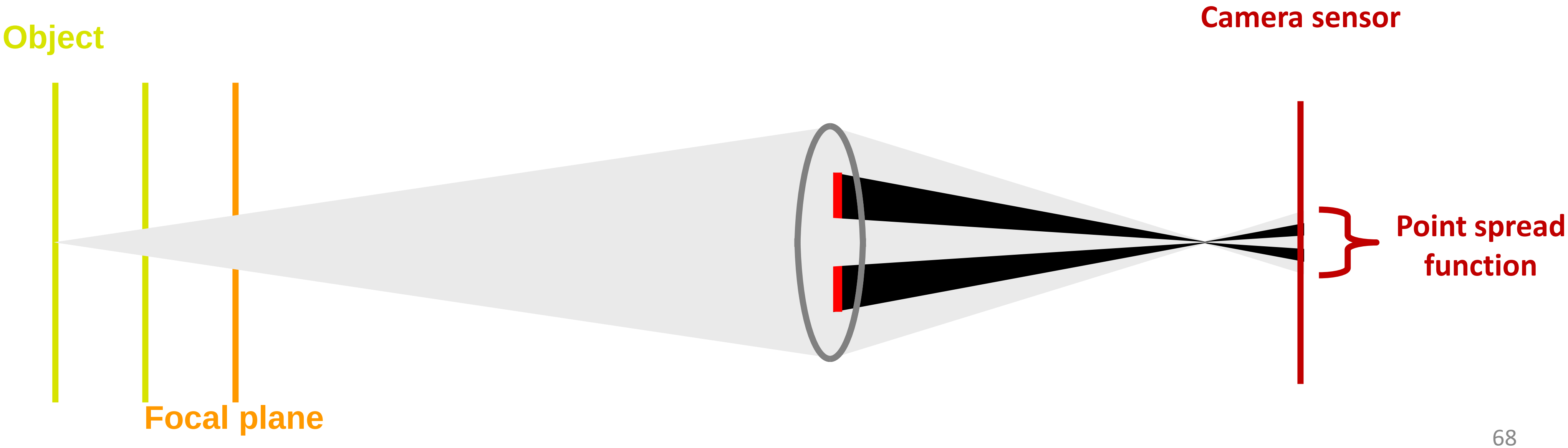
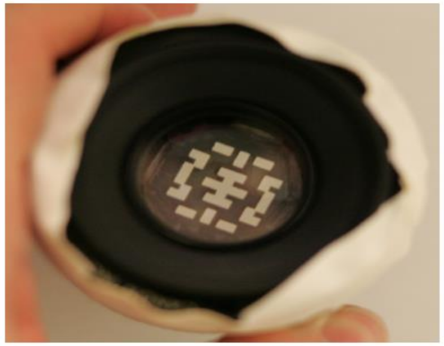


Image of a defocused point light source



Coded aperture: lens with occluder

Aperture pattern



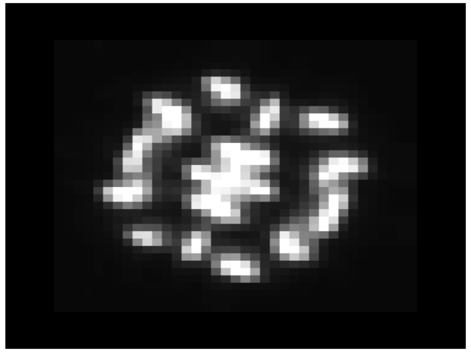
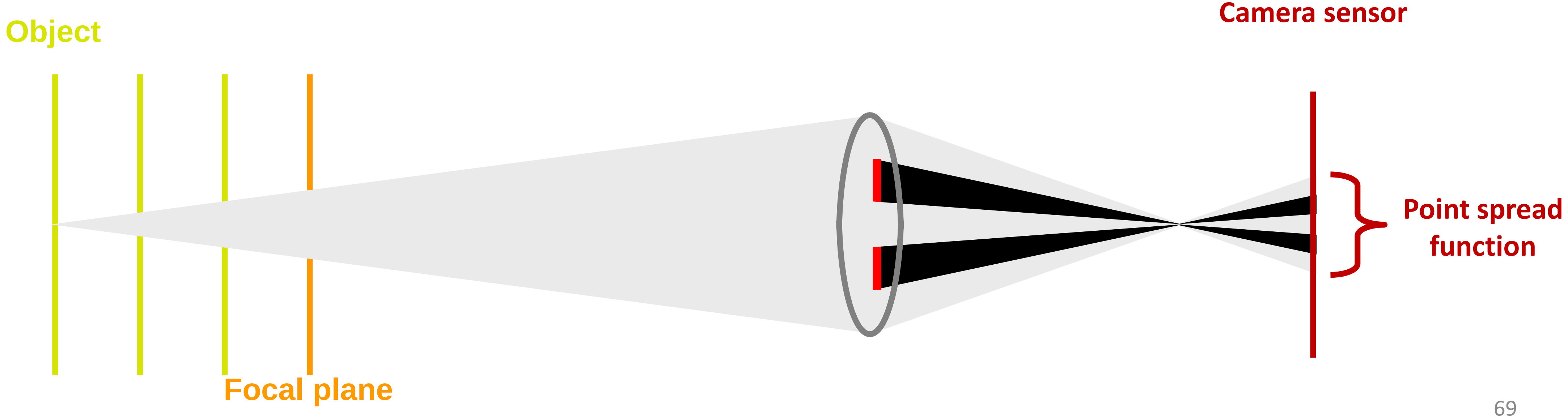
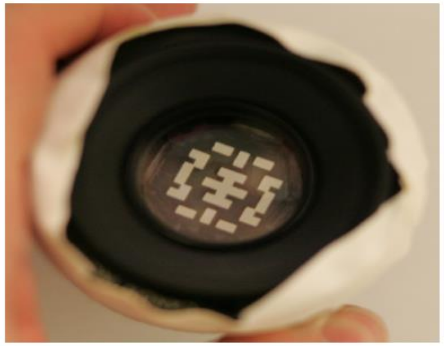


Image of a defocused point light source



Coded aperture: lens with occluder

Aperture pattern



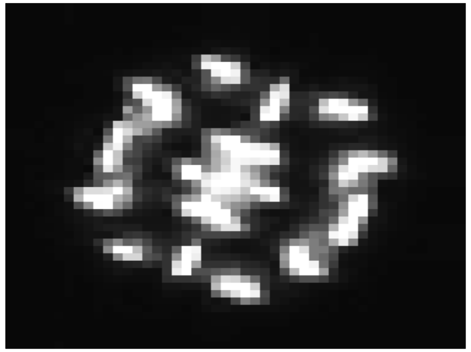


Image of a defocused point light source

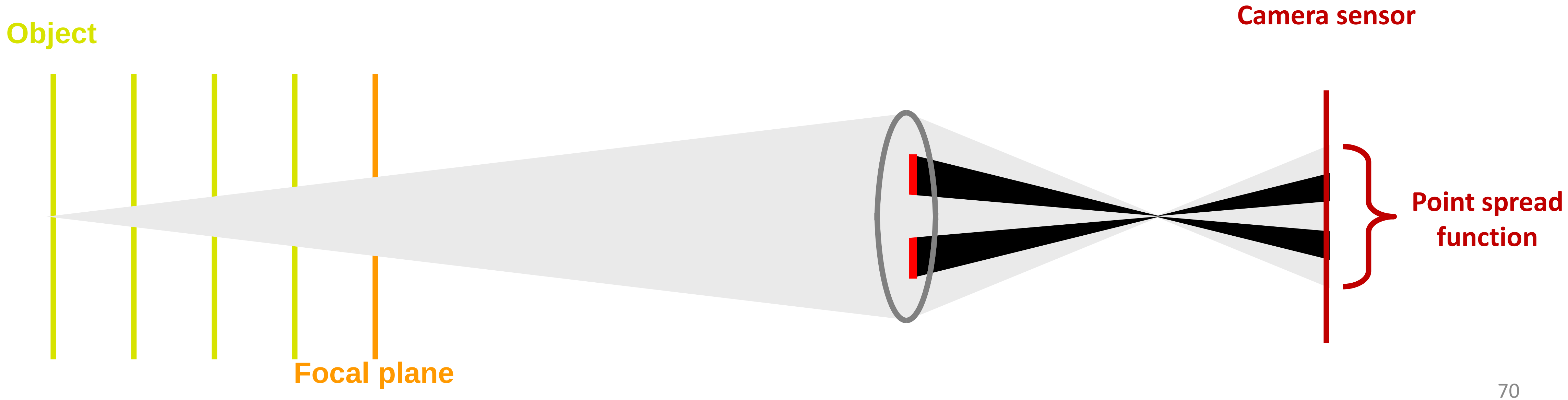
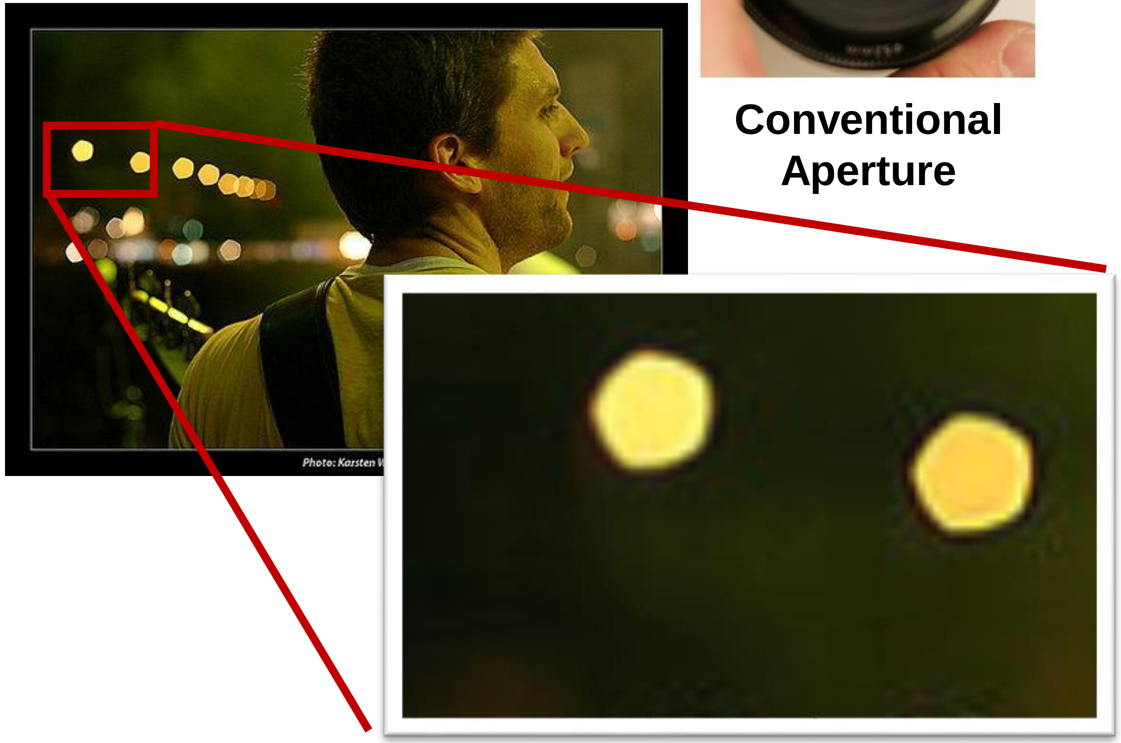


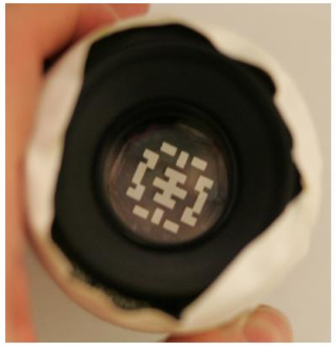
Image of a point light source

Conventional Aperture

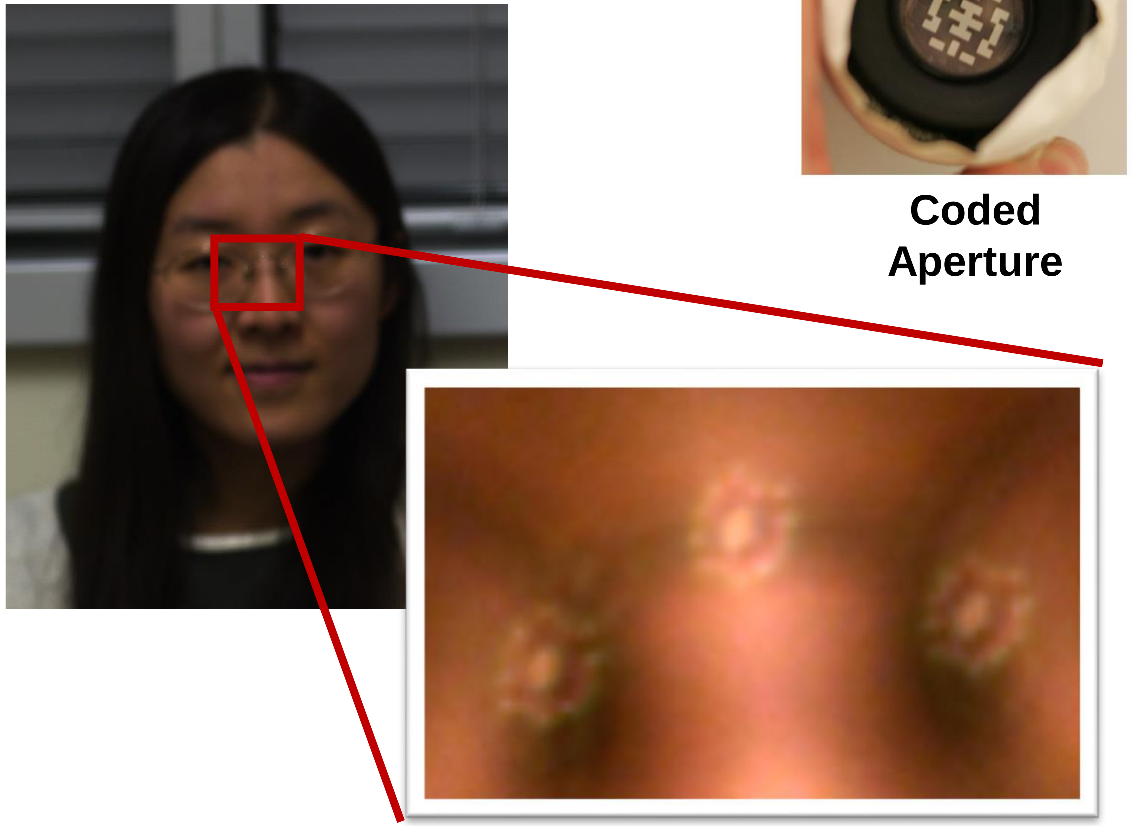


Captured Image

Defocused images
≠ natural images!

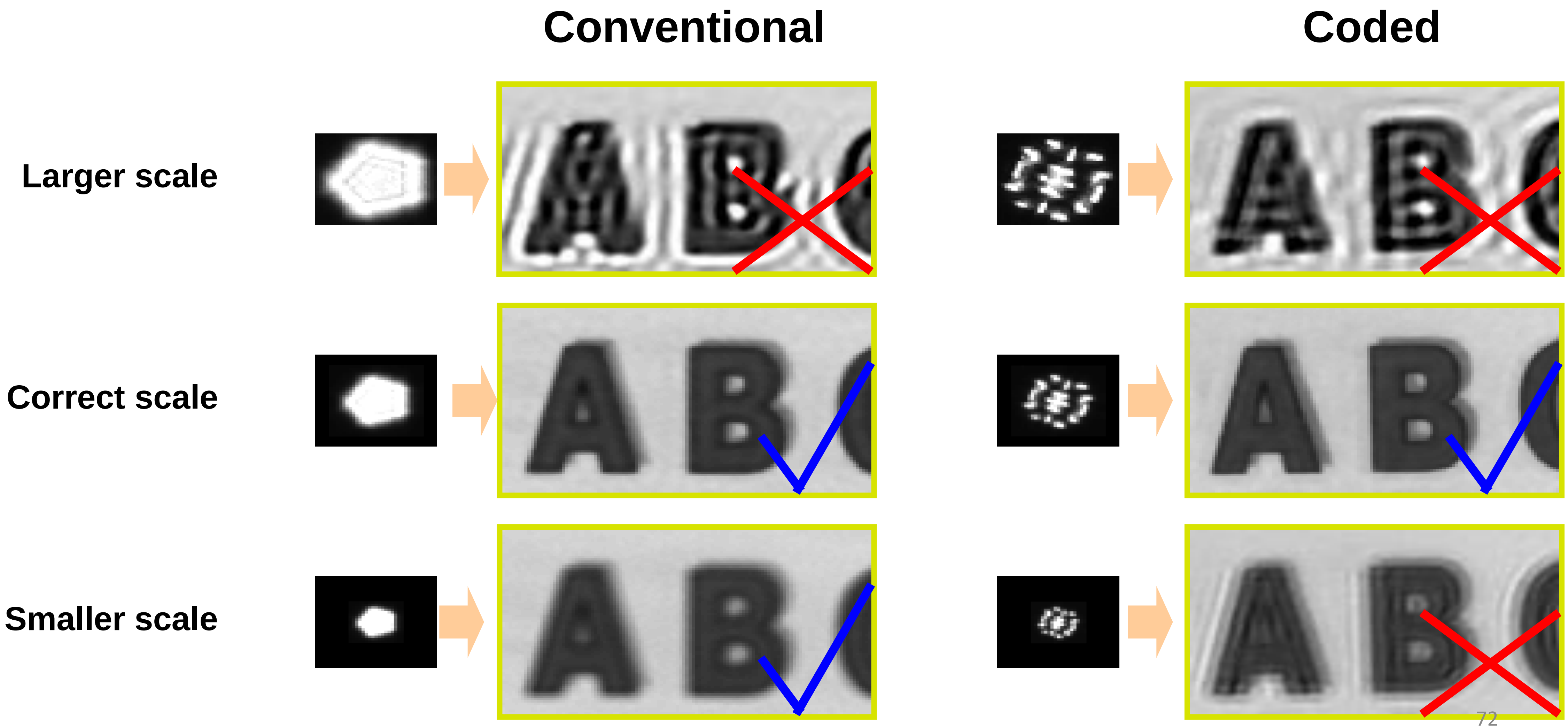


Coded Aperture



Captured Image

Scale estimation - comparison



Input

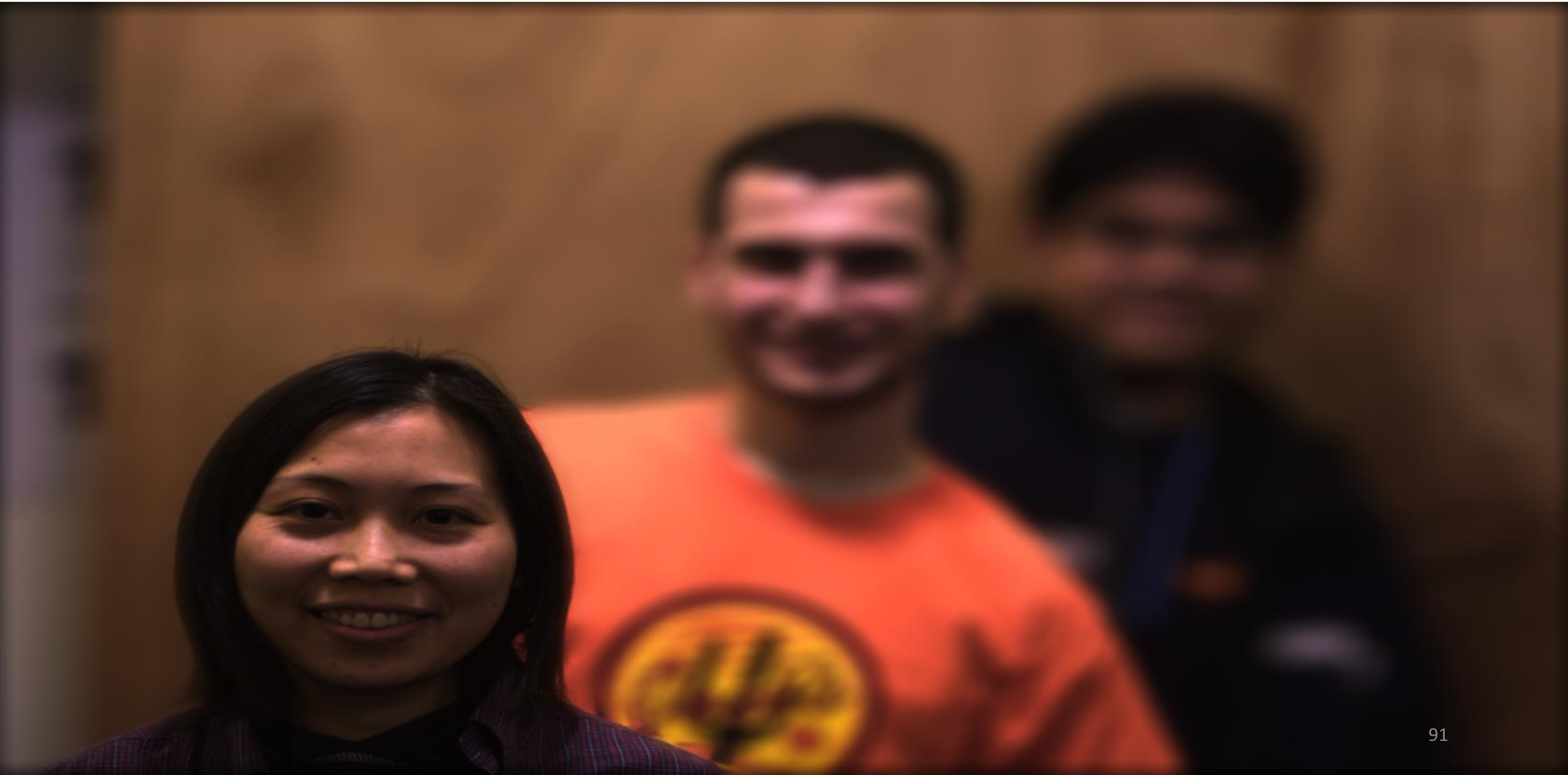
All-focused (deconvolved)


Input


All-focused
(deconvolved)

Digital refocusing from a single image

Digital refocusing from a single image

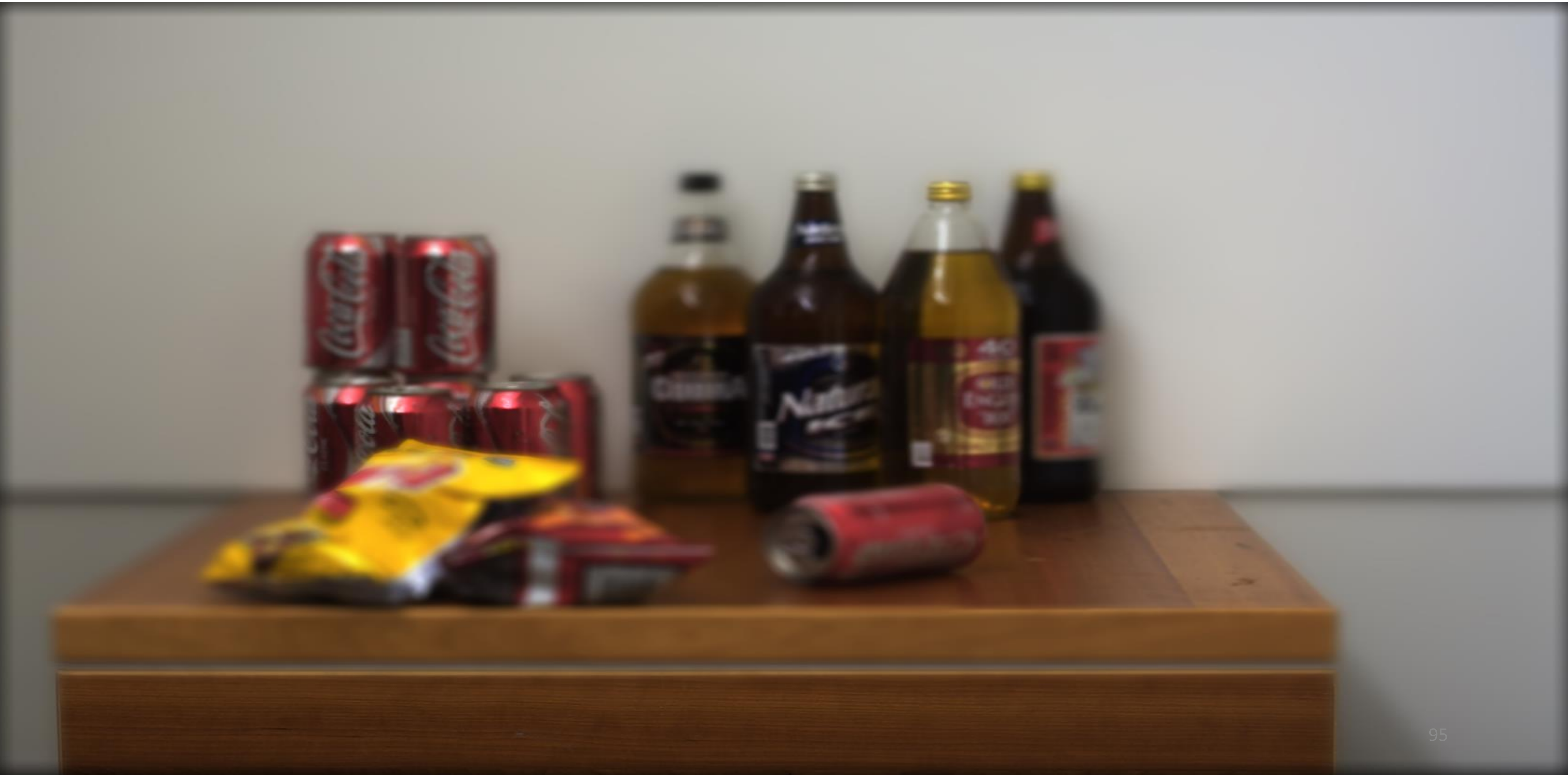


Digital refocusing from a single image

Digital refocusing from a single image

Digital refocusing from a single image

Digital refocusing from a single image



Digital refocusing from a single image

Disadvantages of coded aperture

Blocks light

OTF still have zeros and not so easy to invert

Another solution: Lattice focal lens

Does not block light

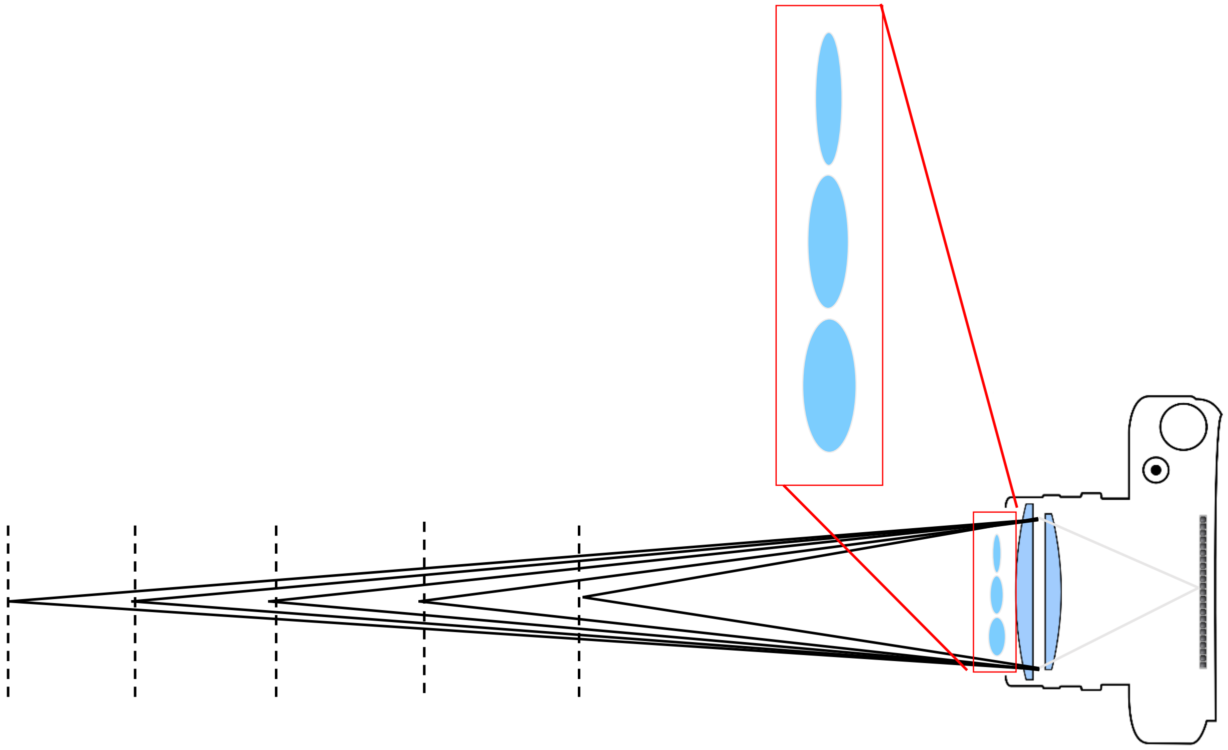
OTF as high as possible

[Levin et al. 2009]

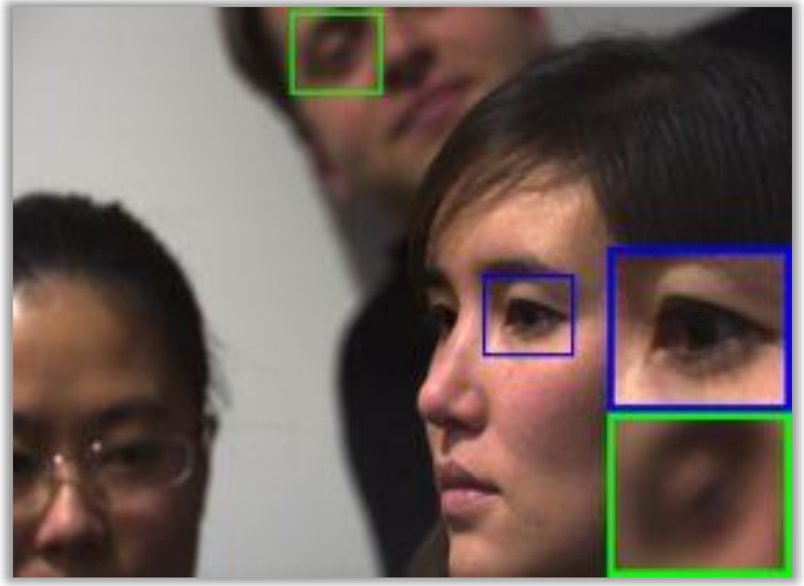
Lattice Focal Lens

[Levin et al. 2009]

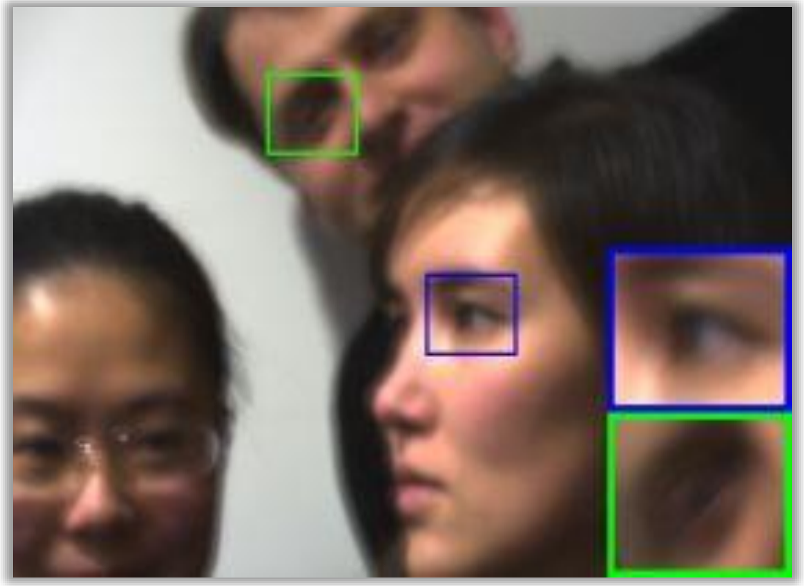
superimpose array of lenses with different focal lengths!



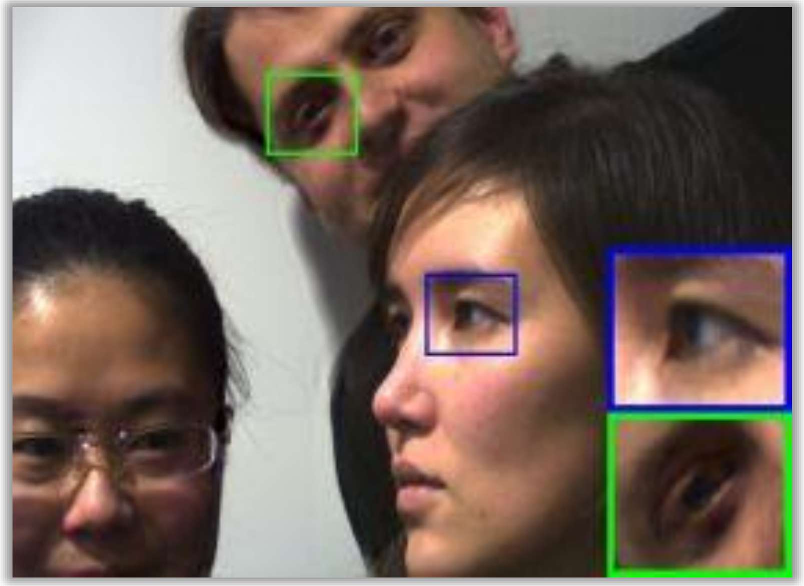
Lattice Focal Lens



conventional camera



lattice focal lens

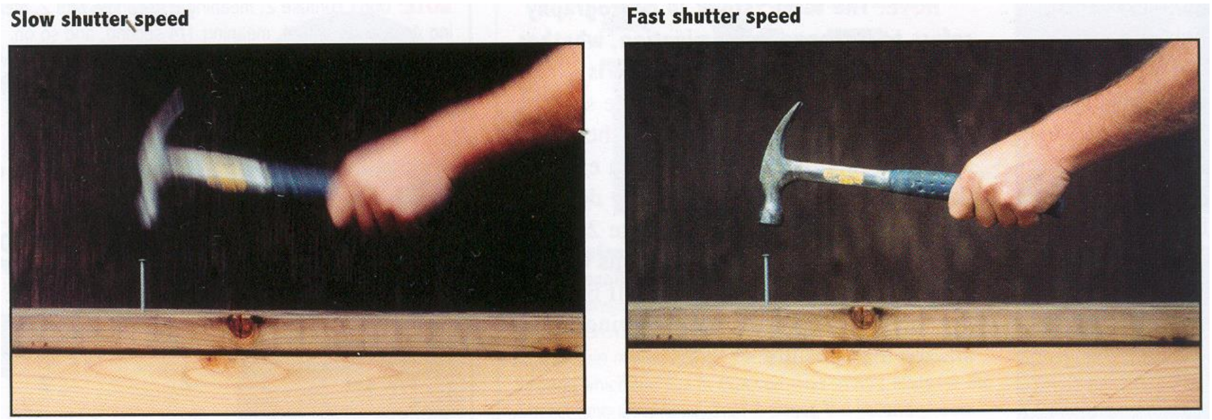


all-in-focus image from lattice focal lens

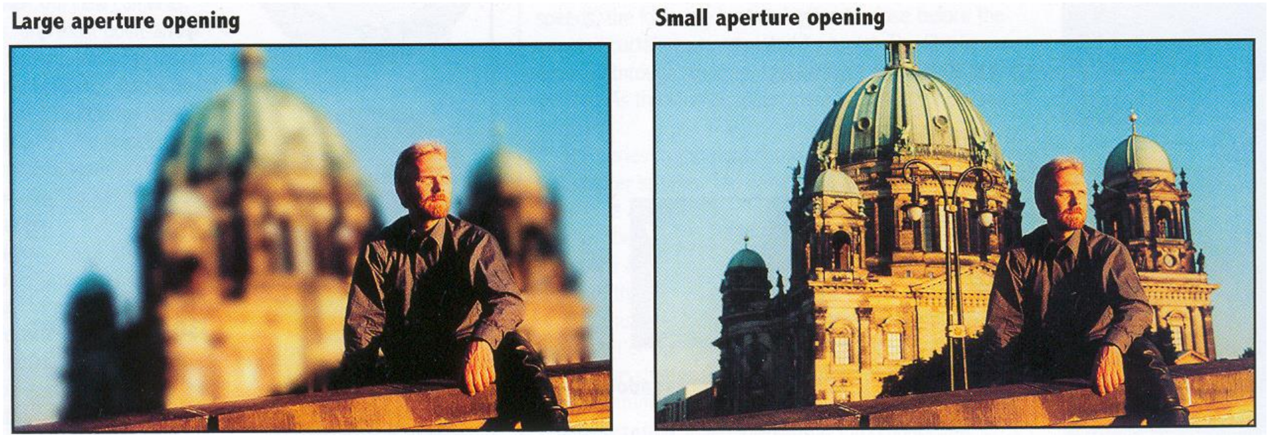
[Levin et al. 2009]

Computational photography approaches to blurring problems

- Motion blur
  - Flutter shutter
  - Motion invariant photography

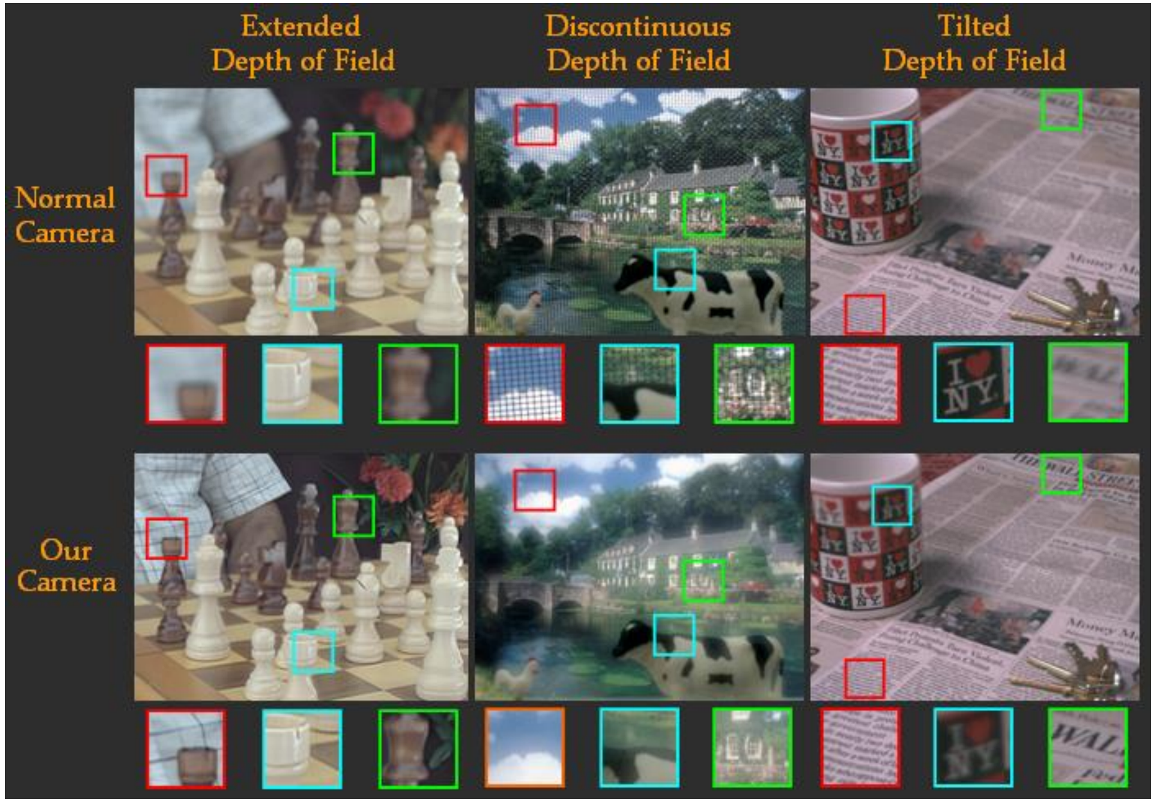


- Defocus blur
  - Coded aperture
  - Lattice focal
  - Flexible depth of field
  - Wavefront coding



Flexible Depth of Field Photography

Hajime Nagahara, Sujit Kuthirummal,
Changyin Zhou, and Shree K. Nayar



Flexible Depth of Field Photography

Problem:

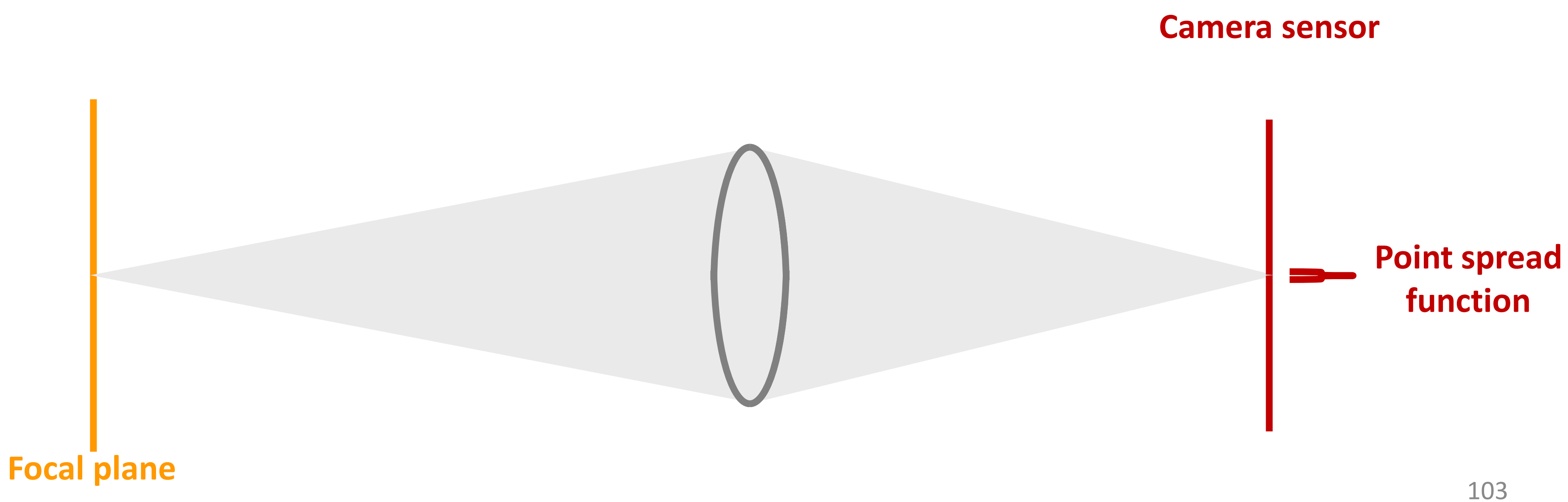
Objects that are not in focus seem blurry.

Goal:

- Compute extended DOF (all-focus image) from a single image.
- Change imaging scheme to achieve *depth-invariant* blur so computational deblurring is easier

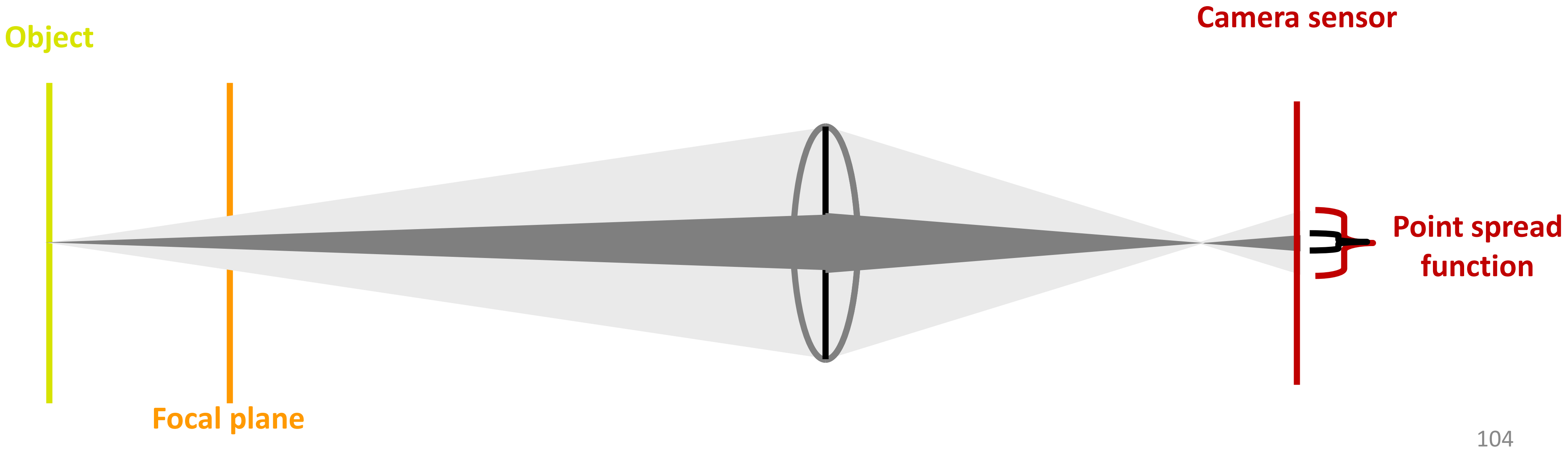
Main Challenge

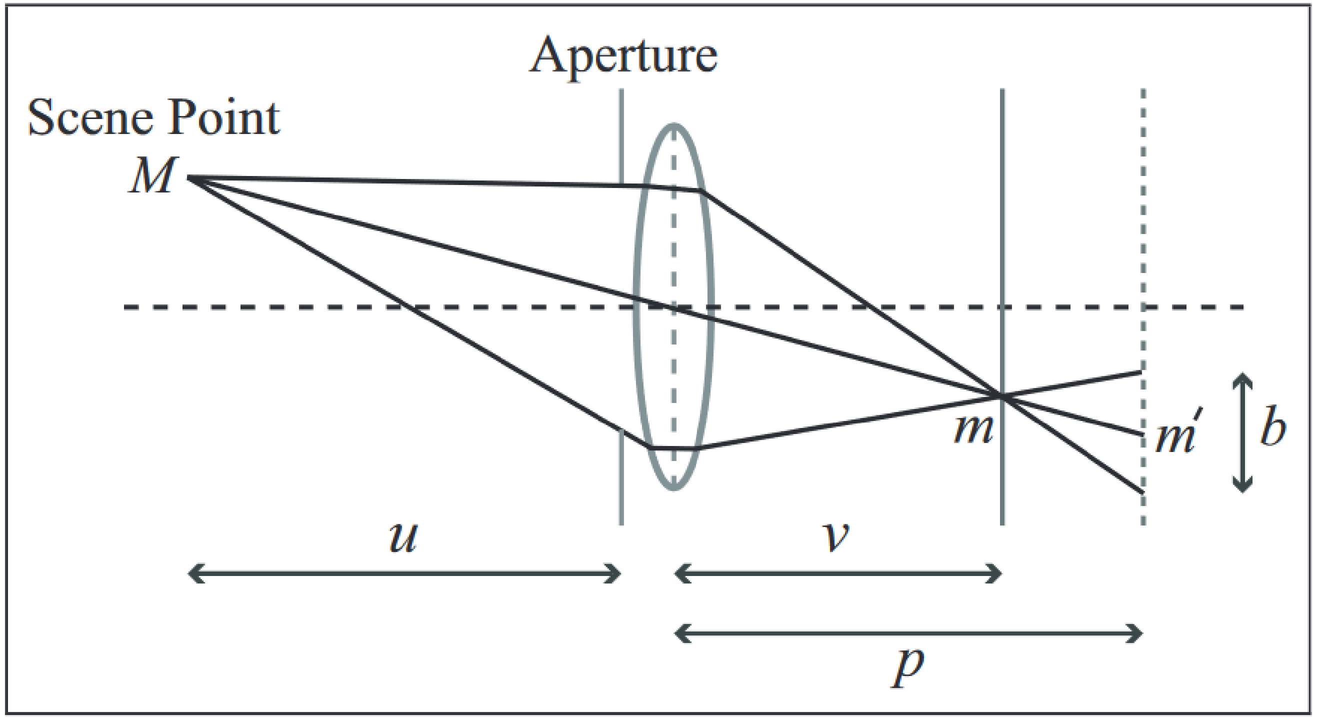
- Trade-off between DOF and SNR
- A lens with a greater f-number projects darker images

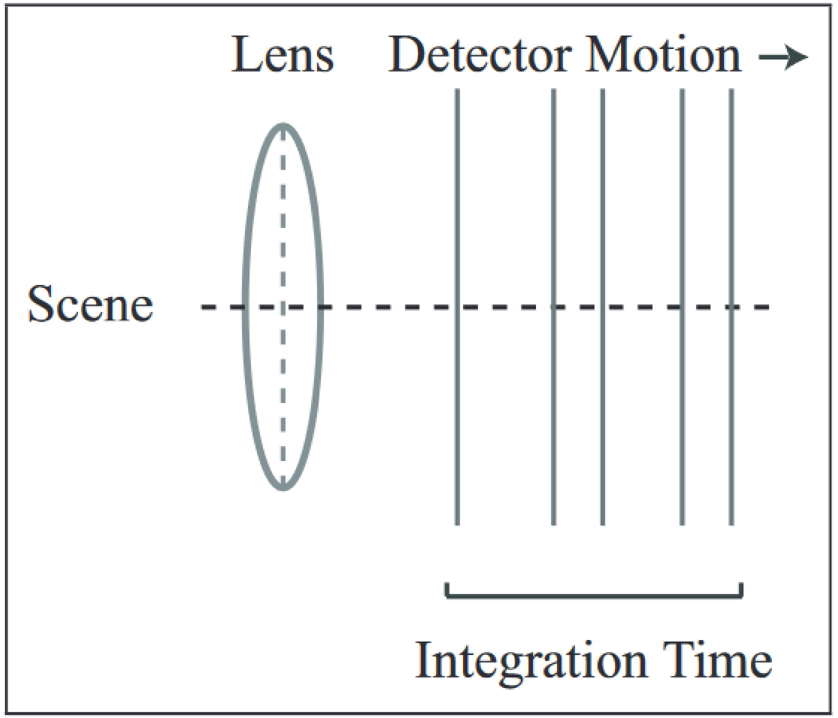


Main Challenge

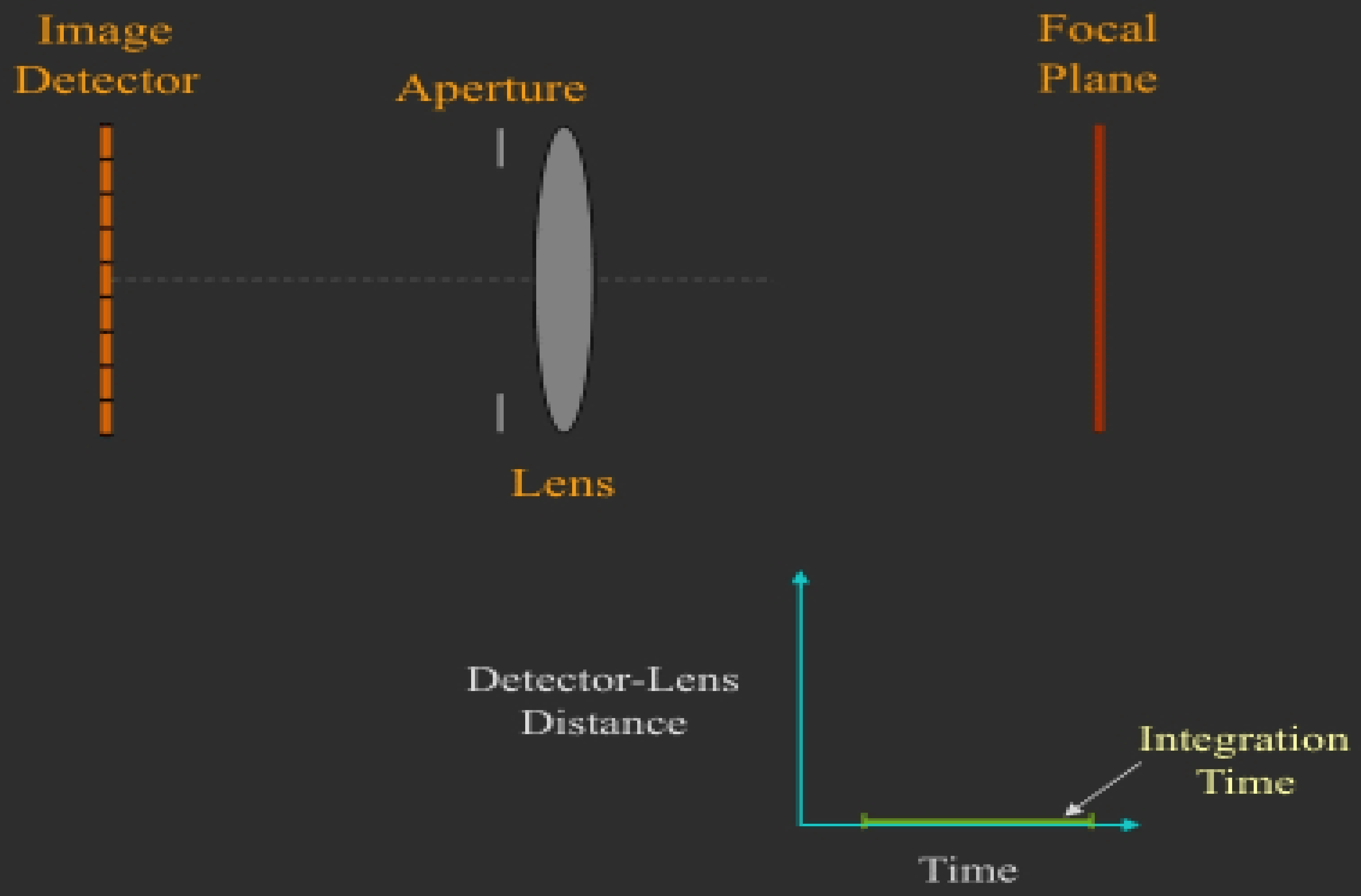
- Trade-off between DOF and SNR
- A lens with a greater f-number projects darker images

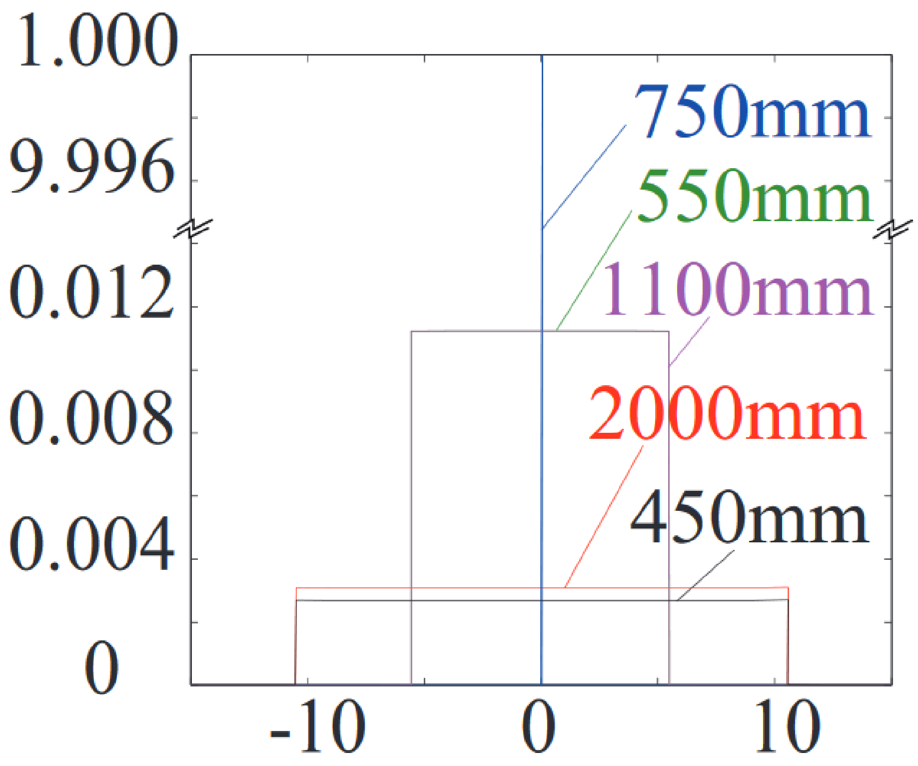



Aperture
Scene Point
M
m
m′
b
u
v
p
(a)

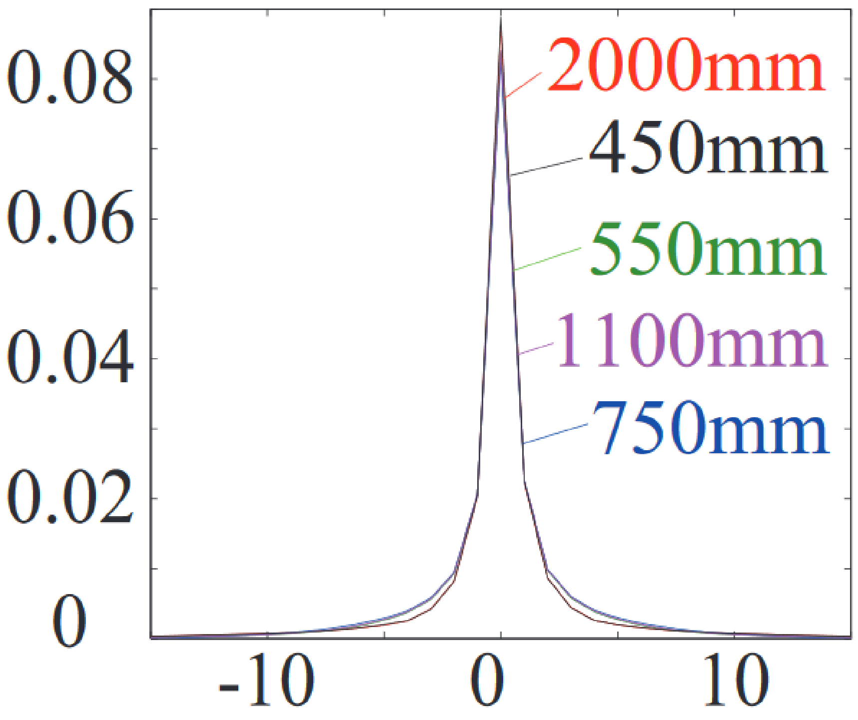

Lens
Detector Motion →
Scene
Integration Time
(b)

Flexible Depth of Field



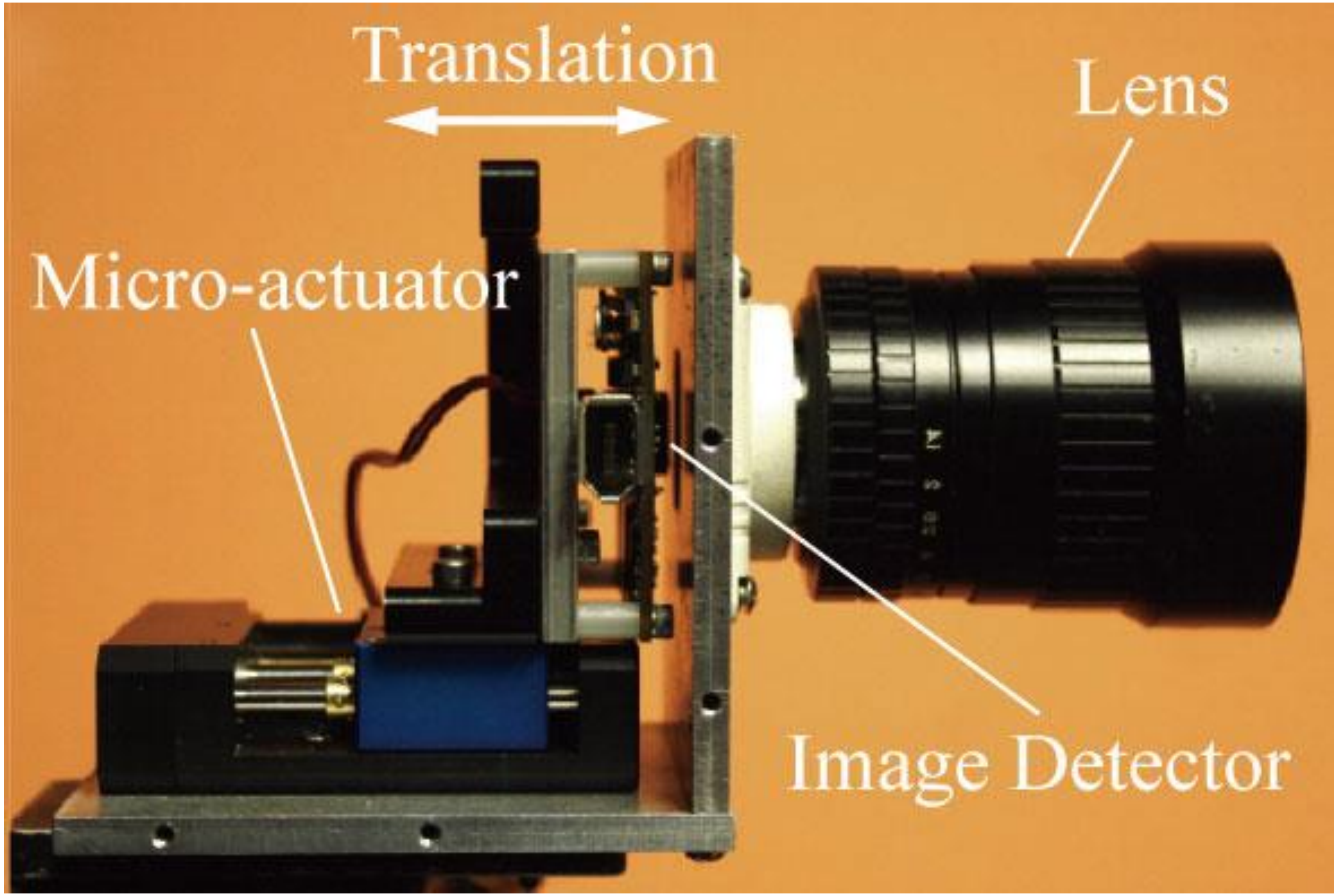


(a) Normal Camera PSF (Pillbox)



(b) EDOF Camera IPSF

Prototype System



Extended Depth of Field

Uniform kernel

Captured Image (*f*/1.4, *T*=0.36sec)

Computed EDOF Image

Image from Normal Camera (*f*/1.4, *T*=0.36sec, Near Focus)

Image from Normal Camera (*f*/8, *T*=0.36sec, Near Focus) with Scaling

Extended Depth of Field: Low Light Imaging

Captured Image
(*f*/1.4, *T*=0.72sec)

Computed EDOF Image

Image from Normal Camera
(*f*/1.4, *T*=0.72sec, Near Focus)

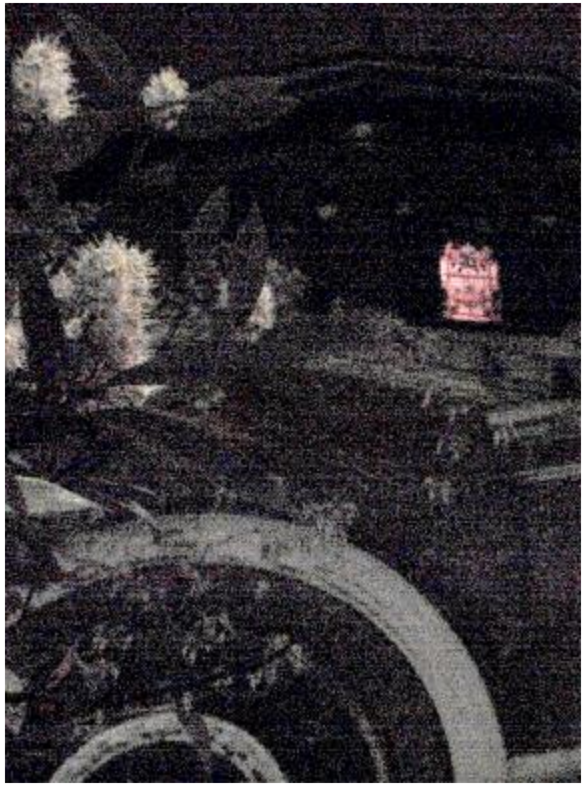


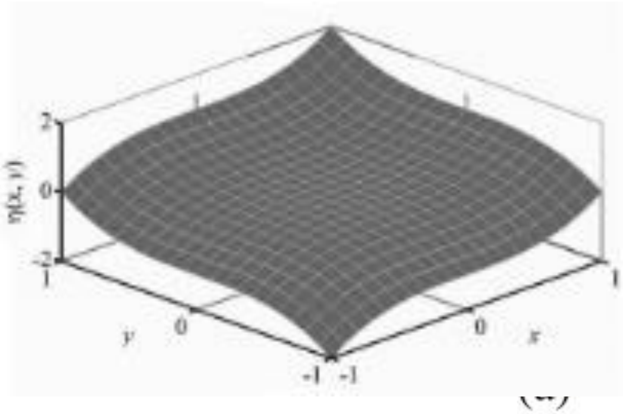
Image from Normal Camera
(*f*/8, *T*=0.72sec, Near Focus)
with Scaling

Wavefront Coding

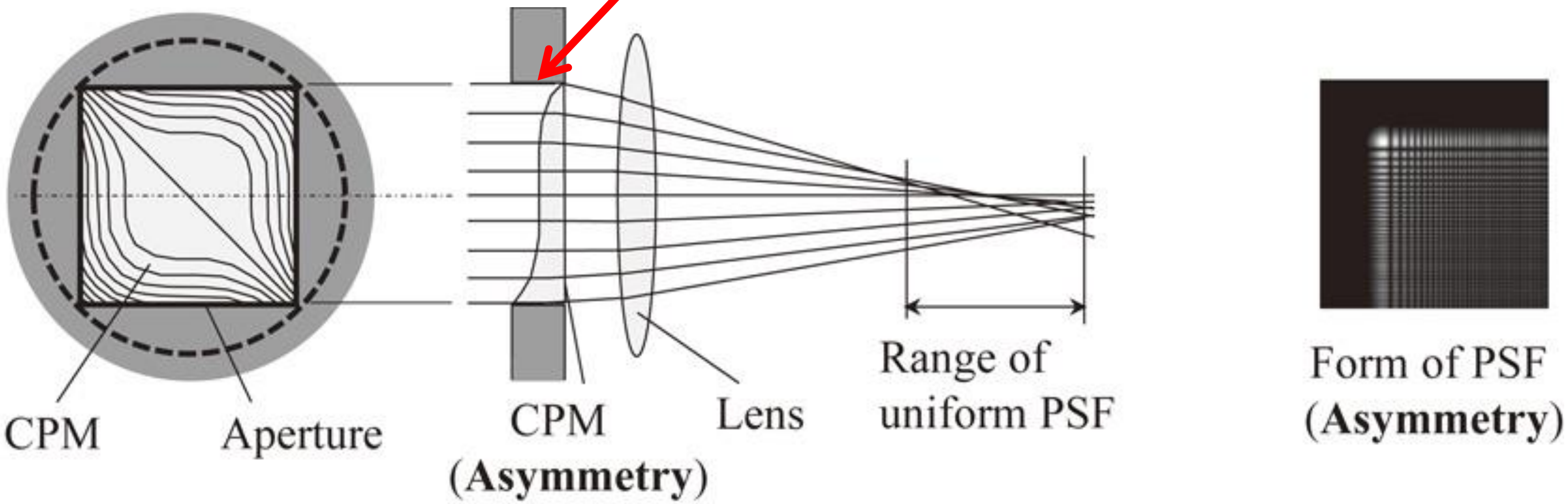
[Dowski and Cathey 1995]

- how to obtain a depth invariant PSF without mechanically moving parts

  → change the lens!



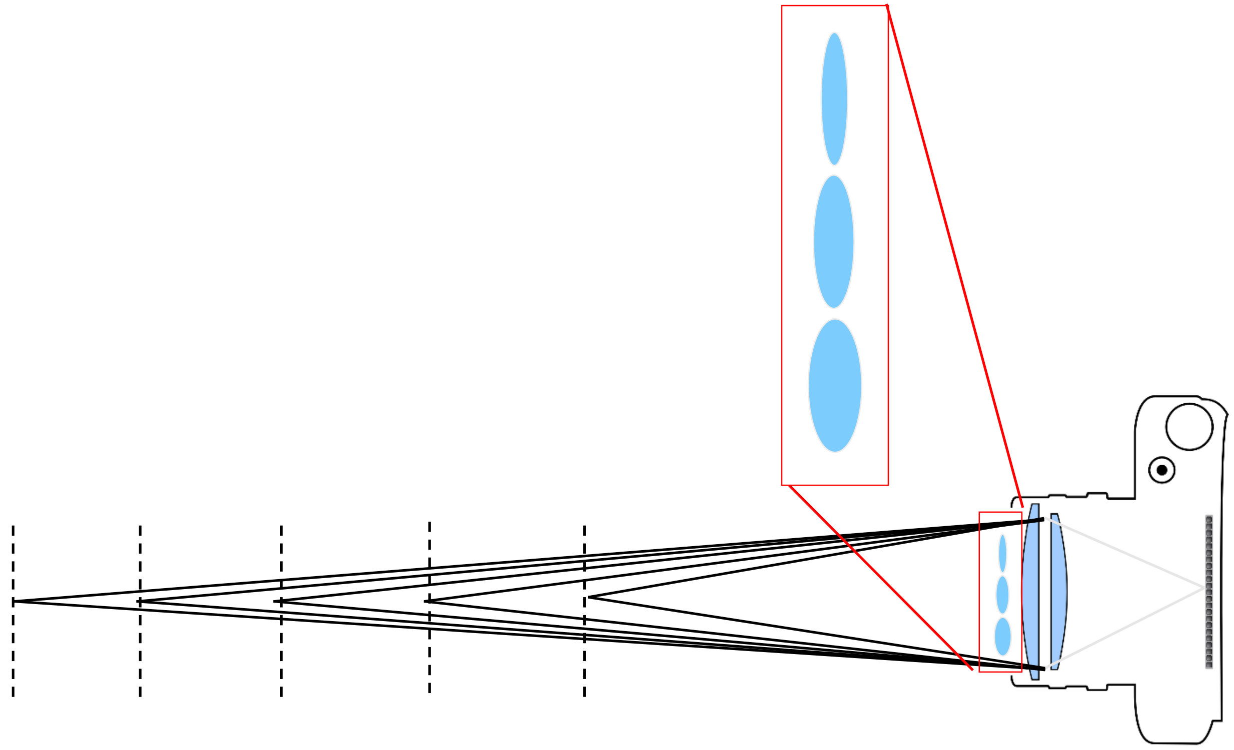
cubic phase plate



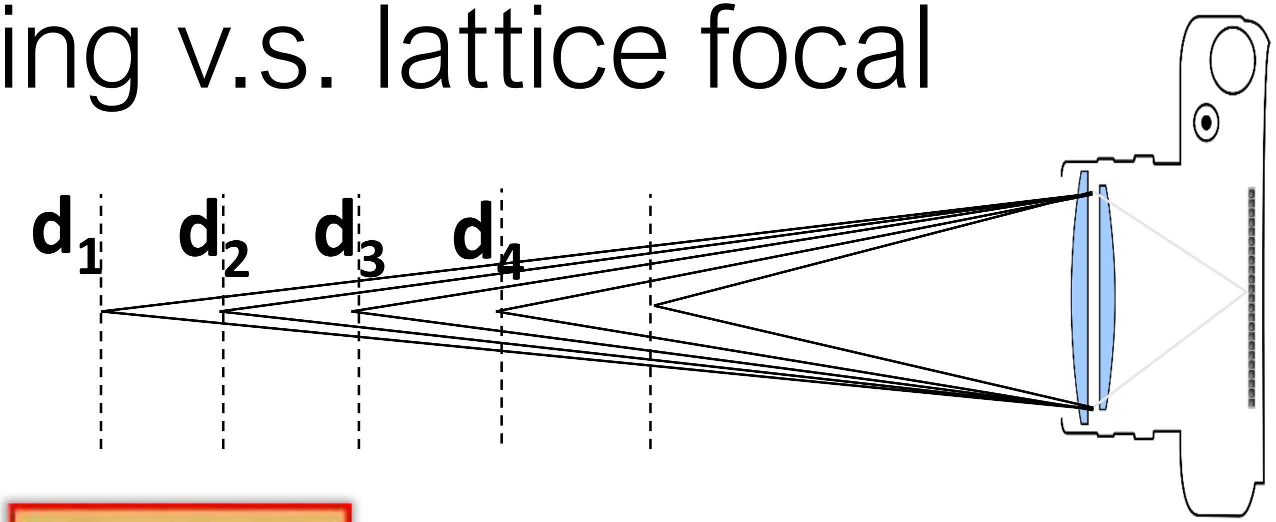
Lattice Focal Lens

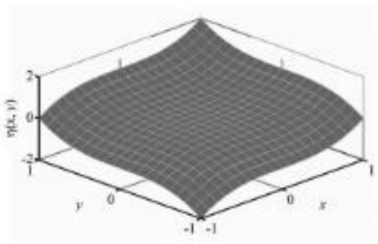
[Levin et al. 2009]

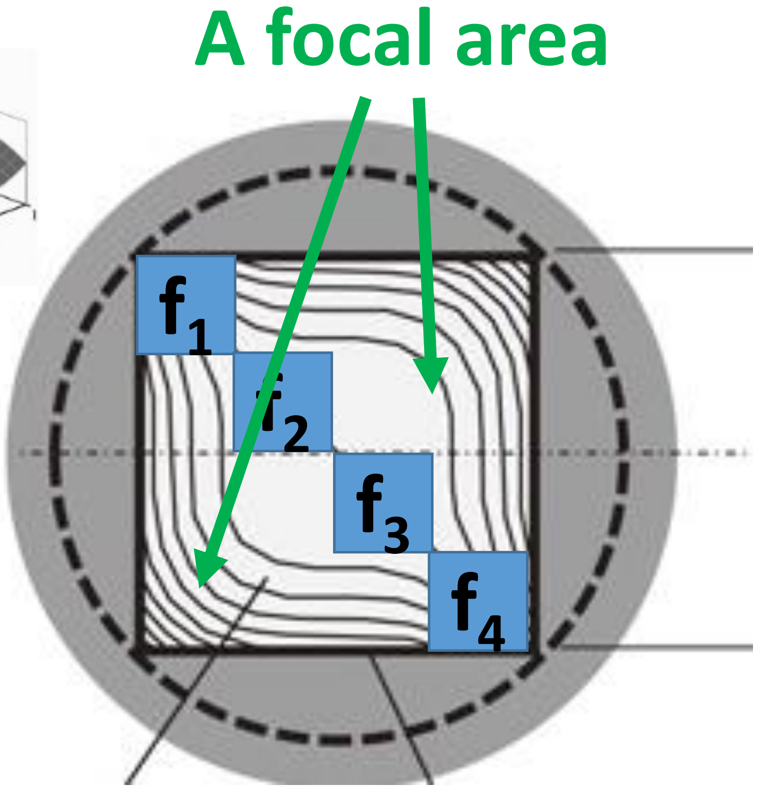
superimpose array of lenses with different focal lengths!

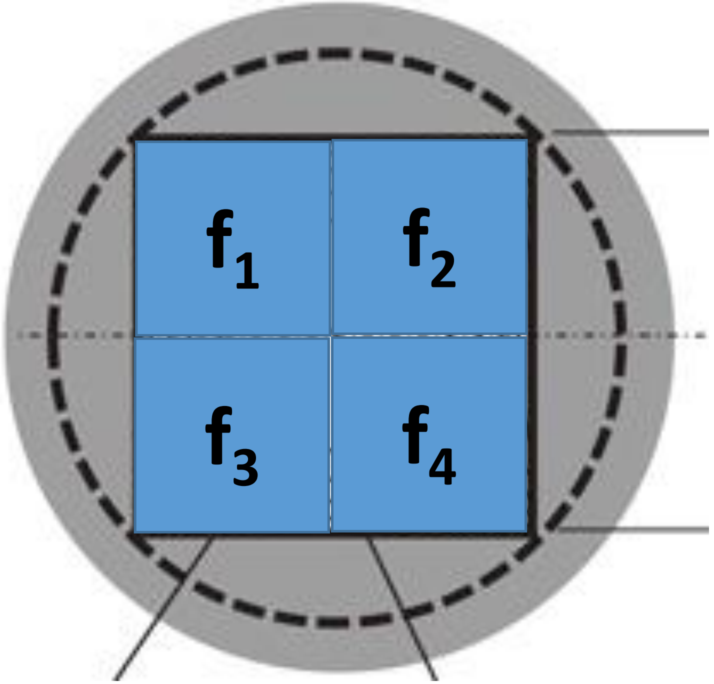


Wavefront Coding v.s. lattice focal









Cubic phase plate

Lattice focal phase plate

Extended DOF Solutions

- Coded aperture
- Wavefront coding
- Focal Sweep
- Lattice-Focal

Depth invariant

Need to estimate depth as part of inversion

-> Highest possible OTF