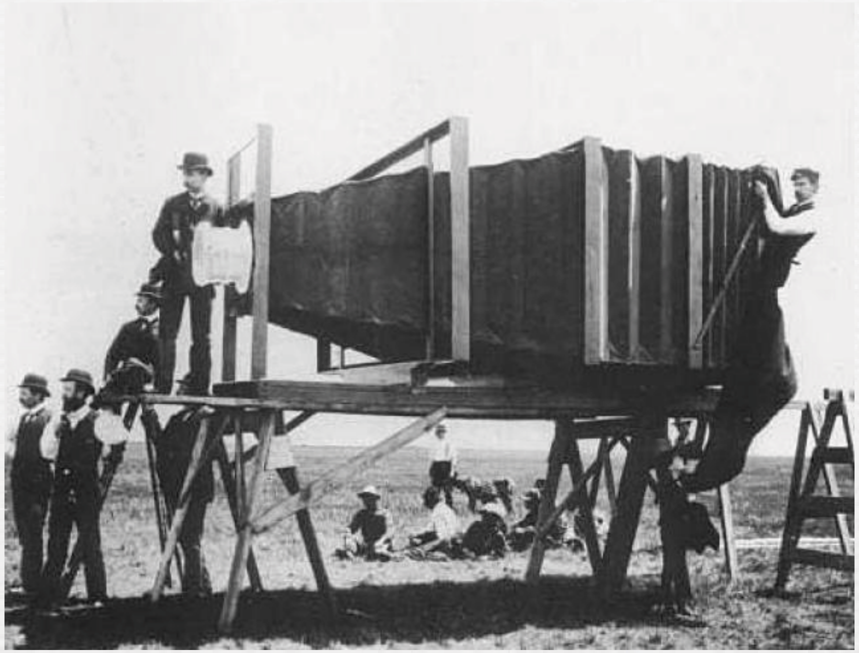
Computation is the New Optics: Coded Imaging in Computational Photography

Frédo Durand
MIT CSAIL

Traditional imaging

- Optics forms the image
- Sensor/film just record
- Displayed image is pretty much the optical image

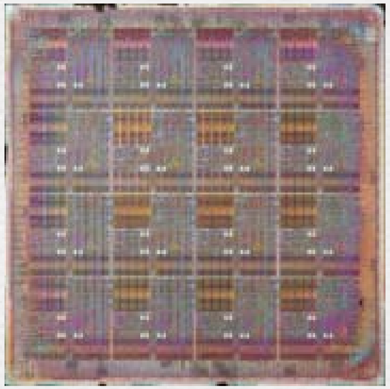


- This is the same for digital camera
  - post-processing is limited

Computational Photography

- Computation between optical and displayed image
  - The optical image is not the final product
  - Can be modified heavily
- Goals:
  - Alleviate physical limitation
  - Capture more information (e.g. depth)
- Best to design computation and optics together

Generalized imaging



Lots of computation

Final image

Related fields

- Computer graphics
  - Try to reproduce reality
- Computer Vision
  - Extract information from visual array
- Image Processing
- Computational Imaging
- Astronomy/telescope
- Radar
- Microscopy
- Medical Imaging

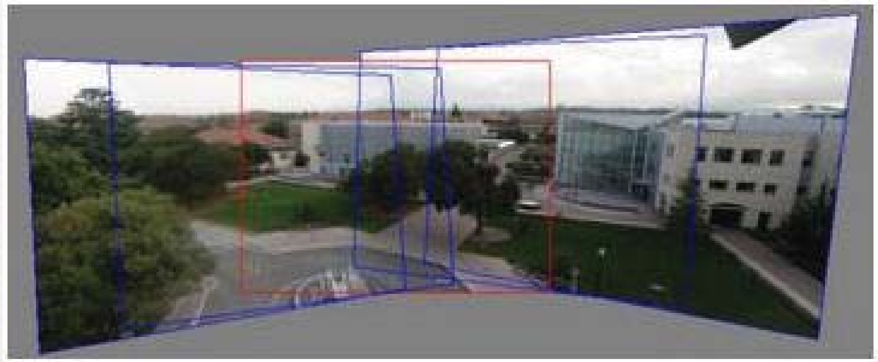
for Eugene

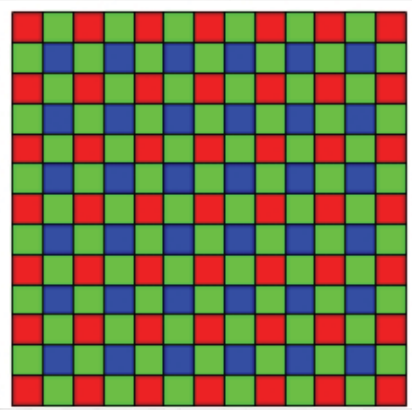
Big ideas in Comp. Photo.

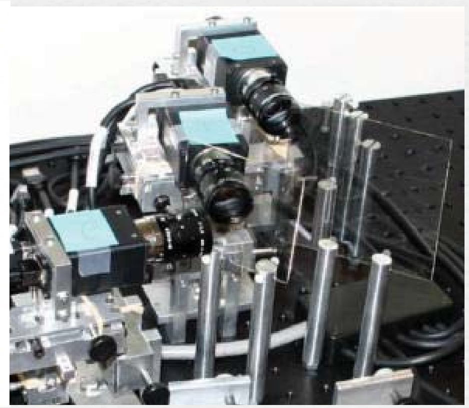
- Goals:
  - Beat physics, better image quality/quantity
  - More data (depth, etc.)
  - Seeing the unseen
  - Creative choices during post-process
  - New visual media
- Multiple-exposure imaging & multiplexing
- Coded imaging
- Prior information
- The raw data is high dimensional
- Active imaging

Multiple-exposure & multiplexing

- Expand capabilities by combining multiple images
- Multiplex through time, assorted pixels, beam splitters, camera array
- e.g.
  - Panorama stitching
  - High-dynamic-range imaging
  - Focus stacks
  - Photomontage
  - Super-resolution

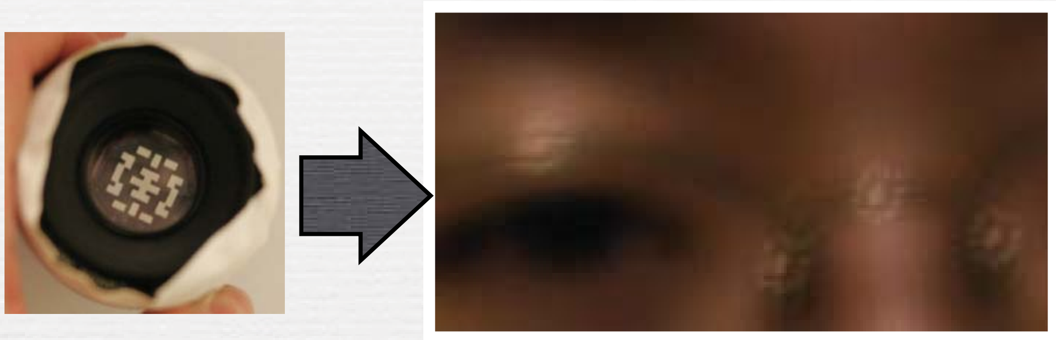




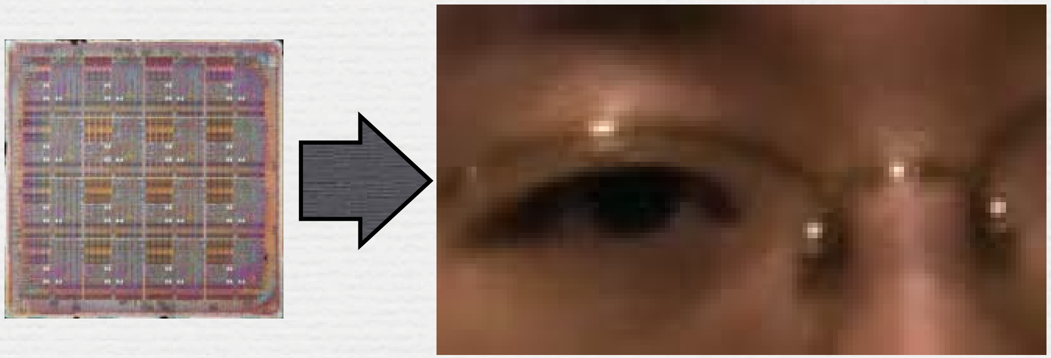


Coded Imaging

Optics encodes information



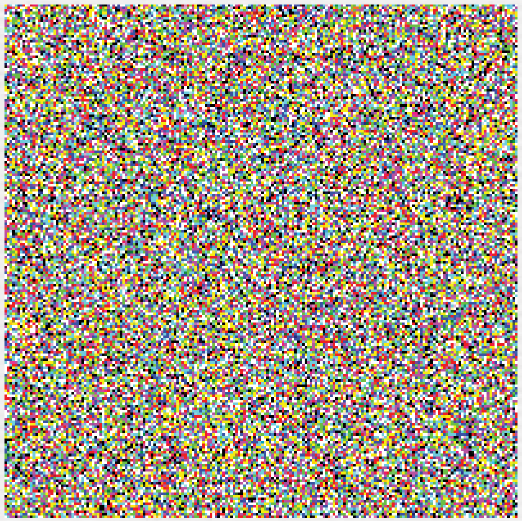
Computation decodes



- e.g.
  - wavefront coding
  - coded aperture
  - flutter shutter
  - motion-invariant
  - compressive sensing
  - heterodyning
  - warp-unwarp

Natural signal prior

- Statistics that distinguish images of the world from random signals
- Use to “bias” algorithms to output more likely results or to disambiguate ill-posed problems
- Extension of regularization
- e.g.
    - Denoising
    - Deconvolution
    - Compressive sensing
    - Light field prior

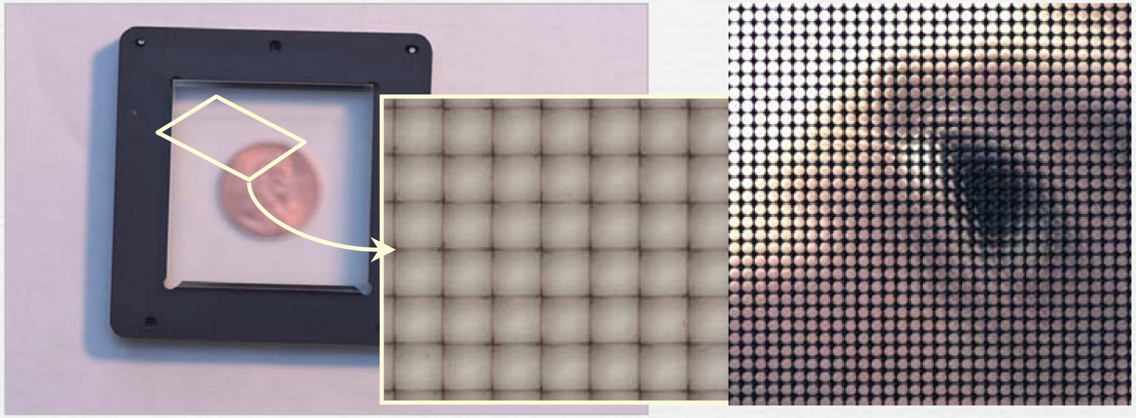


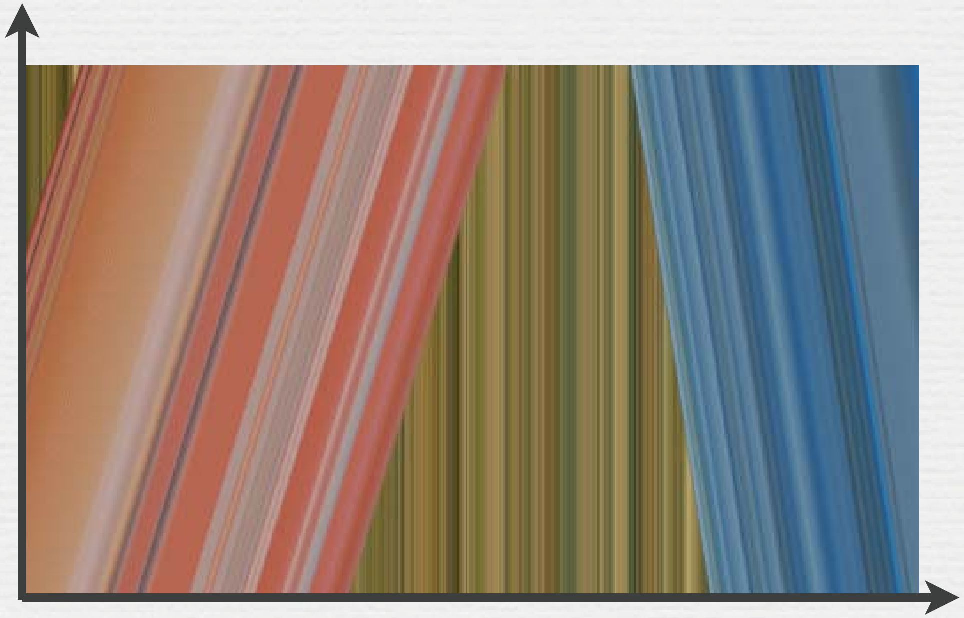
Random

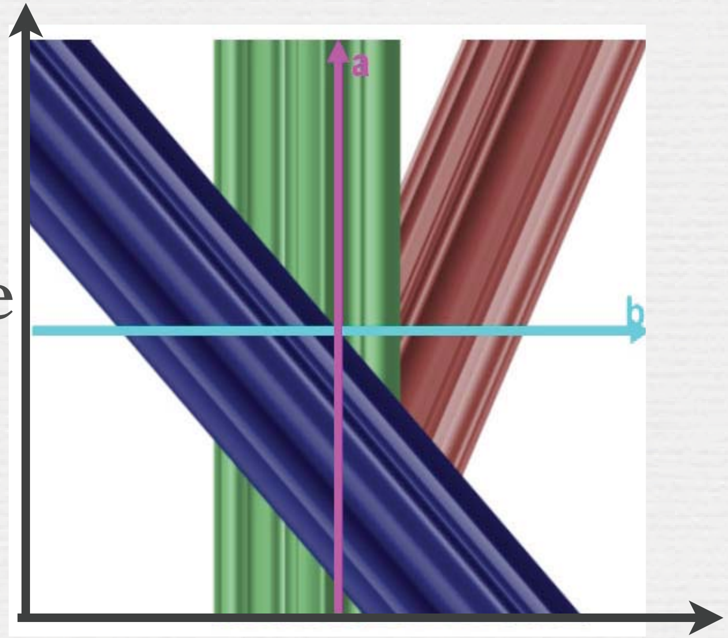
“Natural” image

The raw data is high dimensional

- Light field: 4D (space-angle)
- Time space: 3D
- +Fourier

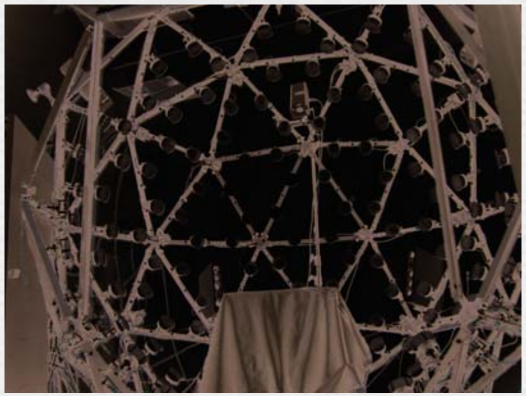


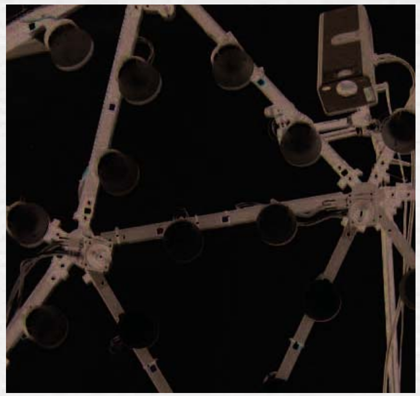




Active imaging

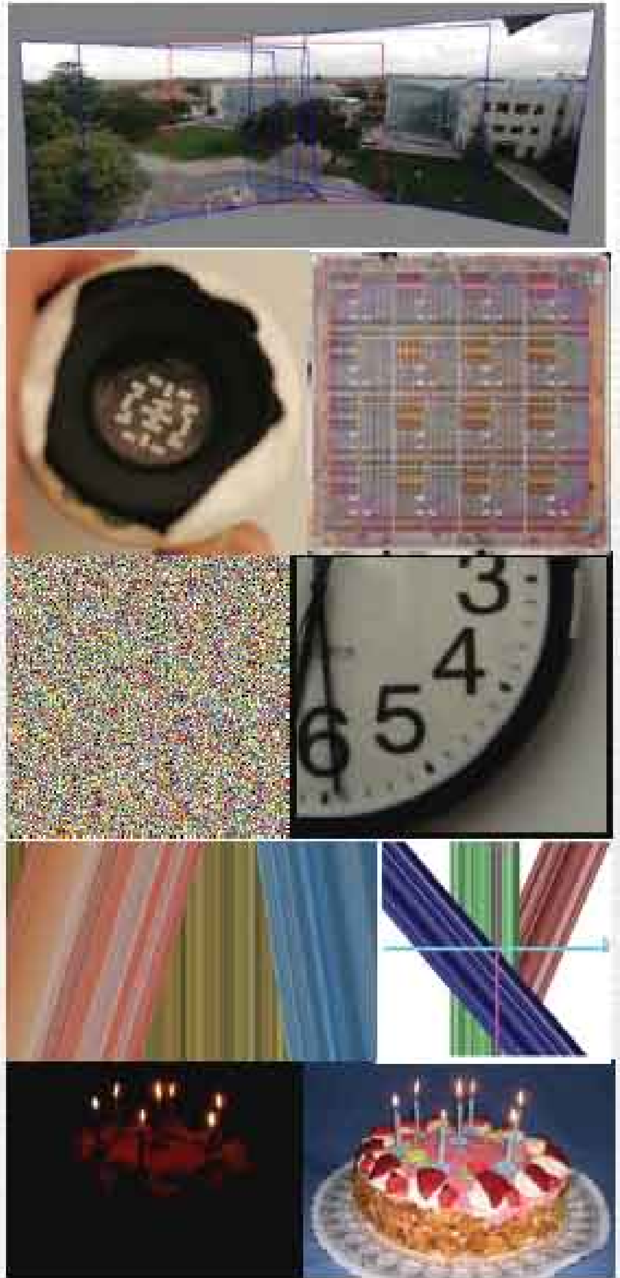
- Modulate light to facilitate information gathering
- e.g.
  - Flash/no flash
  - Light stages
  - Dual imaging
  - Structured-light scanning





Recap: Big ideas in Comp. Photo.

- ✦ Multiple-exposure & multiplexing
- ✦ Coded imaging
- ✦ Prior information
- ✦ The raw data is high dimensional
- ✦ Active imaging

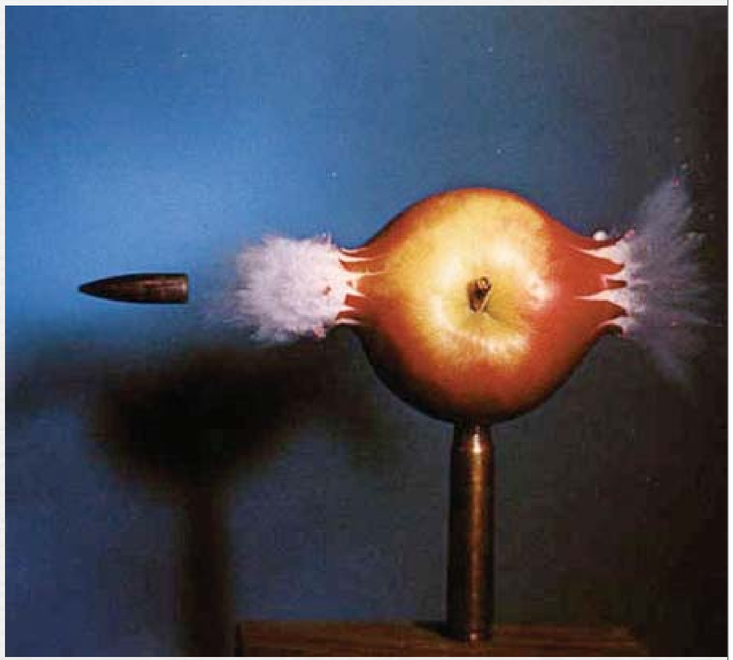


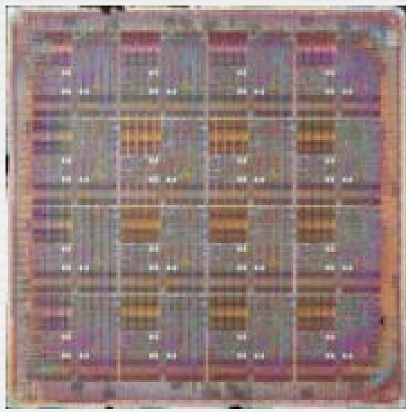
Computational Photography: Coded Blur Removal

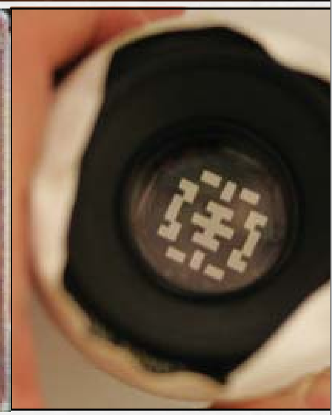
Frédo Durand
MIT CSAIL
with Anat Levin, Peter Sand, Rob Fergus, Taeg Sang Cho, Bill Freeman

This talk: blur removal

- Blur often reduces image quality
  - Motion blur, diffraction, defocus
- Traditional solution:
  - Faster shutter speed, smaller aperture, bigger aperture
  - Often increases noise (gathers less light)
- Today: computational solution
  - Remove blur given **single** image
  - Imaging hardware + software







Motion blur


Most of the scene is static
red bull is moving
from left to right

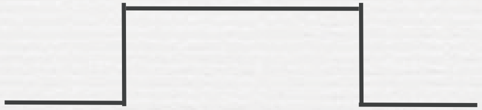
Can we remove the blur?

- Given single image with blur
- Blur is mostly a linear process, just invert it
  - called deconvolution
- But we need to know the exact blur
- But the process needs to be invertible
  - Lose as little information as possible

Deconvolution

Kernel identification

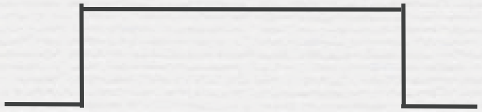
$\stackrel{?}{=}$



$\otimes$

Input blurry image — Correct kernel — Output from correct kernel

$\stackrel{?}{=}$

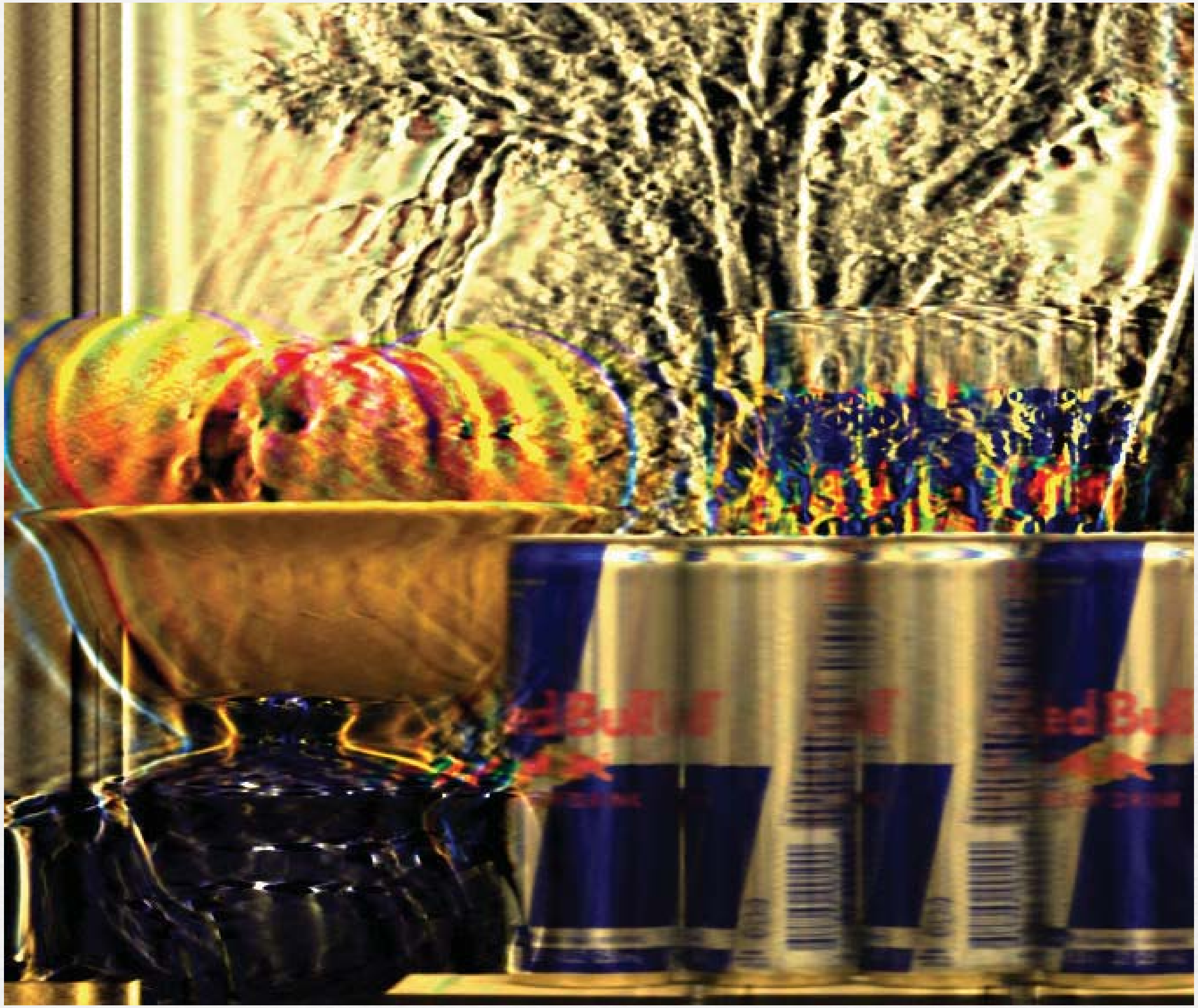


$\otimes$

Input blurry image — Wrong kernel — Output from wrong kernel

Kernel identification

- The kernel is spatially varying



Entire image deblurred with kernel corresponding to the cans' velocity

Challenge with deblurring

- Blur destroys information
    - Often box filter

Deblurring given known blur:

blurred input

deblurred

static input

Challenge with DoF and motion

- Blur destroys information
  - Often box filter
- Kernel identification
  - Spatially varying

This talk: two opposite solutions

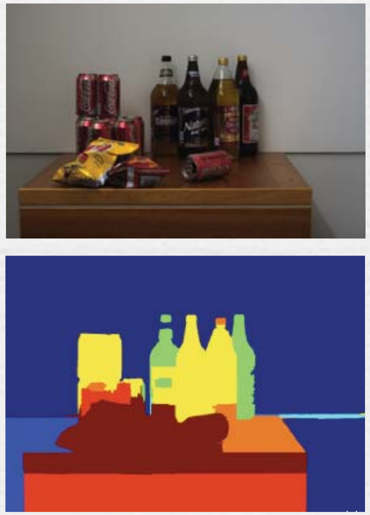
- Make blur very variant
  - Easier to identify
- Make blur invariant
  - No need to identify

- In both cases, we also make the blur easy to invert
  - Preserve information

Outline

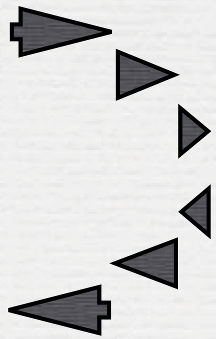
- Natural image prior: help the inversion

- Coded aperture: make blur vary more with depth

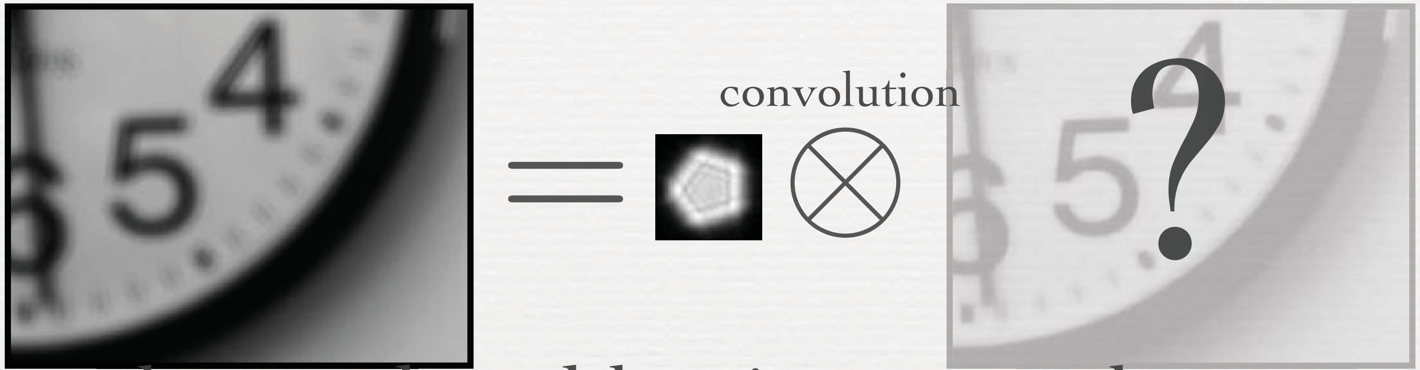


- Motion-invariant photography

(c) Static camera (d) Parabolic camera



Deconvolution with known kernel

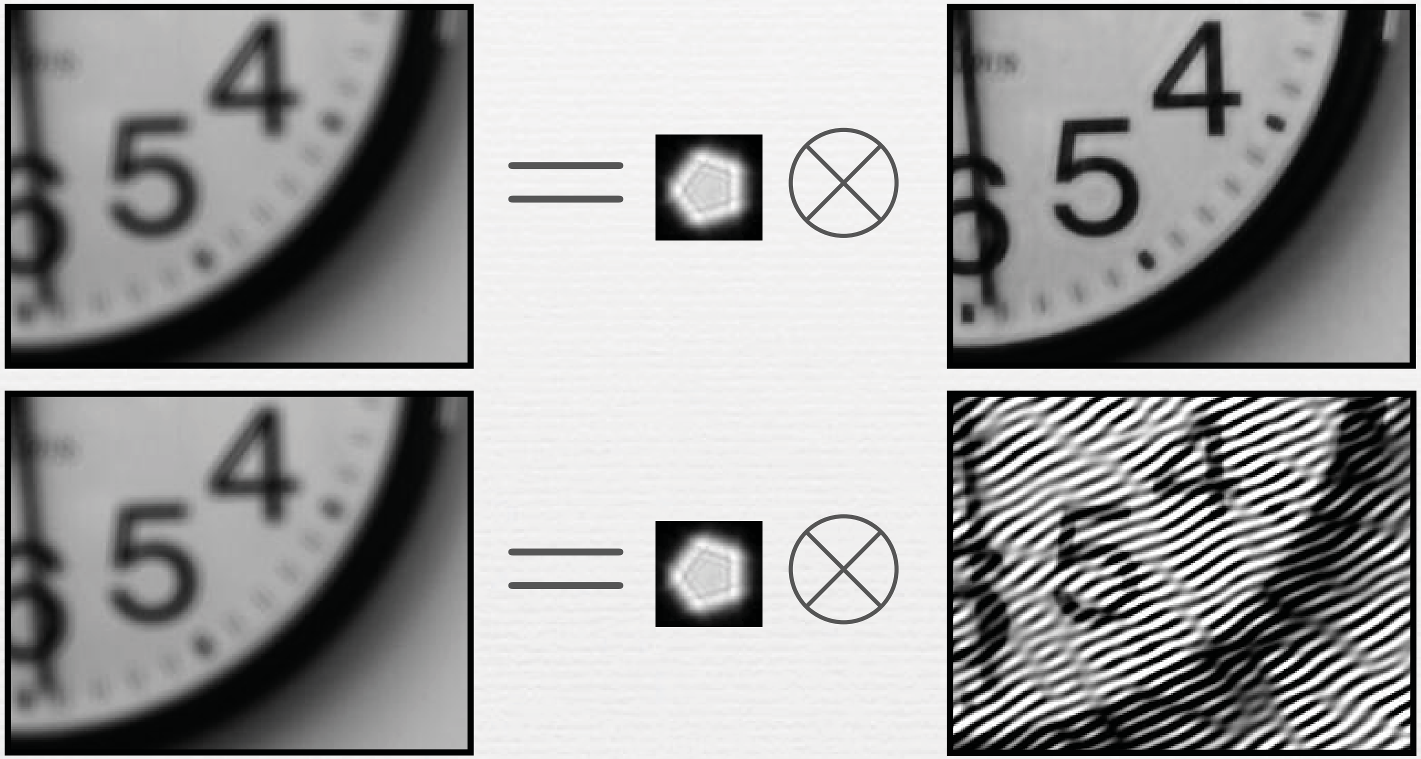


observed blurry image | blurring kernel | unknown sharp image

- Given blurred image and blurring kernel infer sharp image

Deconvolution challenge

- Ill-posed

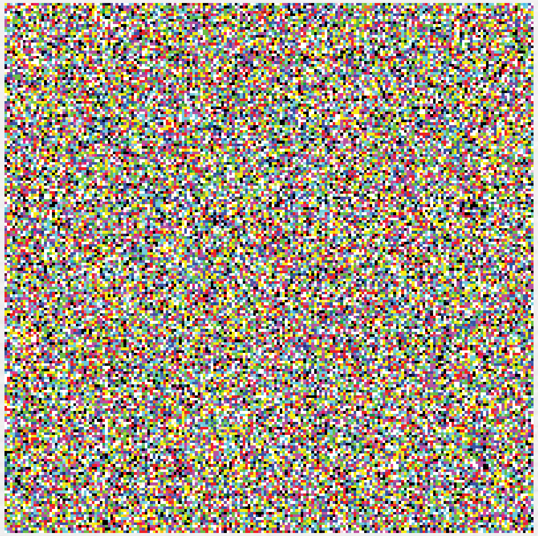


- Traditional algorithms lead to ringing

Richardson-Lucy
deconvolution

Idea 1: Natural image prior

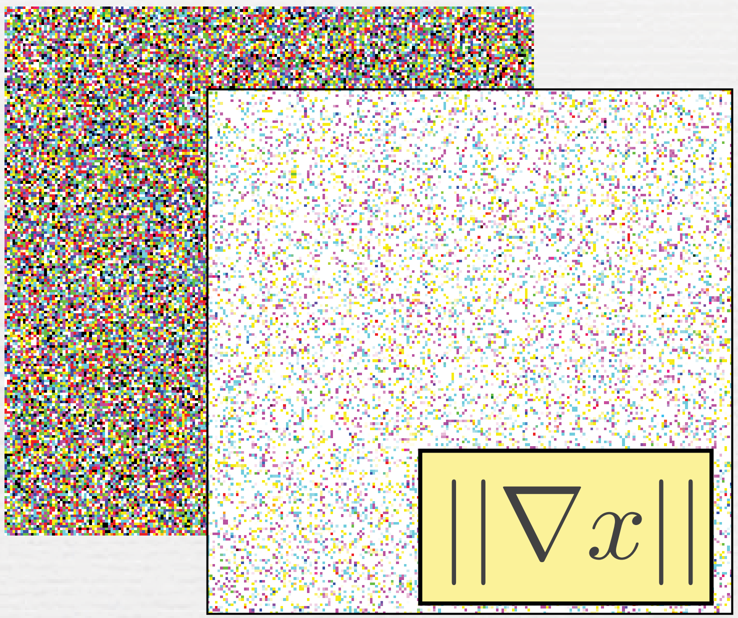
- Random 2D arrays of colors don't look like the world around us

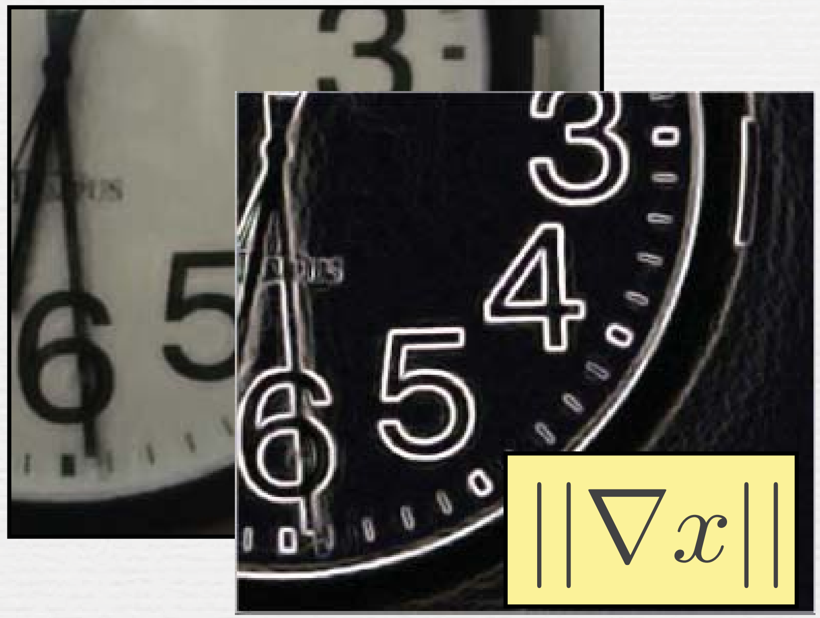


- If we can characterize natural images, we can bias algorithms to output better results

Sparse derivatives prior

- Natural images have sparse derivative
  (the gradient is small almost everywhere)





- Add an optimization term
  (a.k.a. regularization)
  Pay penalty where gradient is non-zero

$$\sum ||\nabla x||^{0.8}$$

Sparsity prior for deconvolution

Input

$$\sum ||\nabla x||^2$$

spread error

$$\sum ||\nabla x||^{0.8}$$

localize error

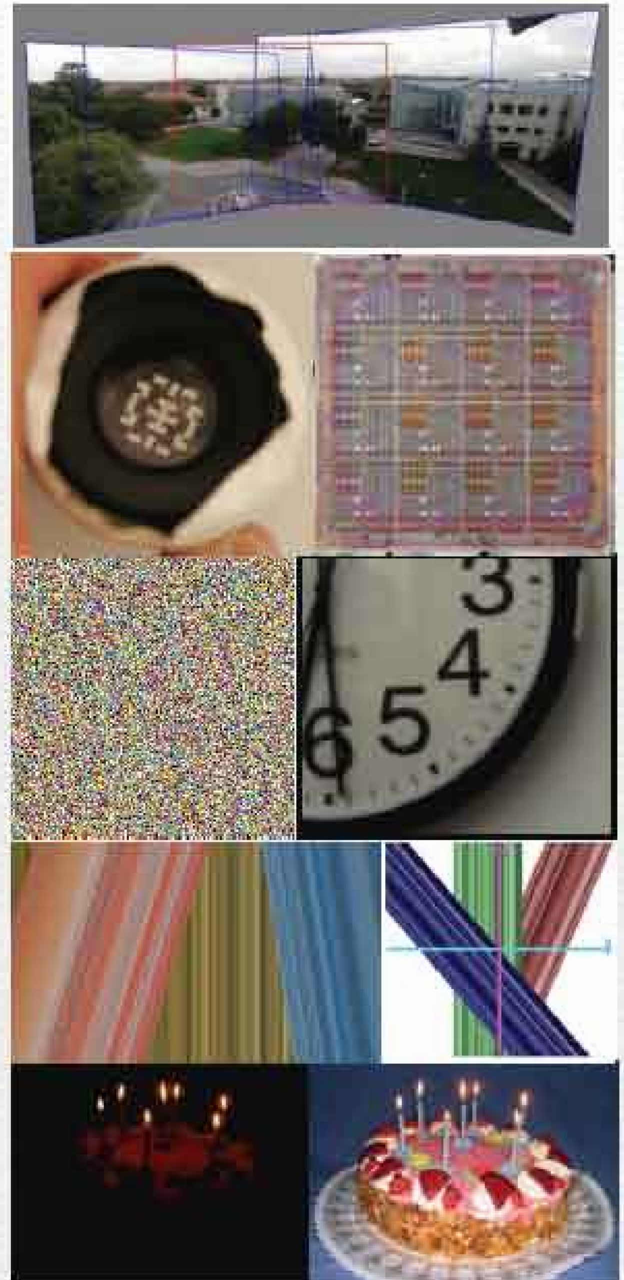
Richardson-Lucy

Gaussian prior

Our sparse prior

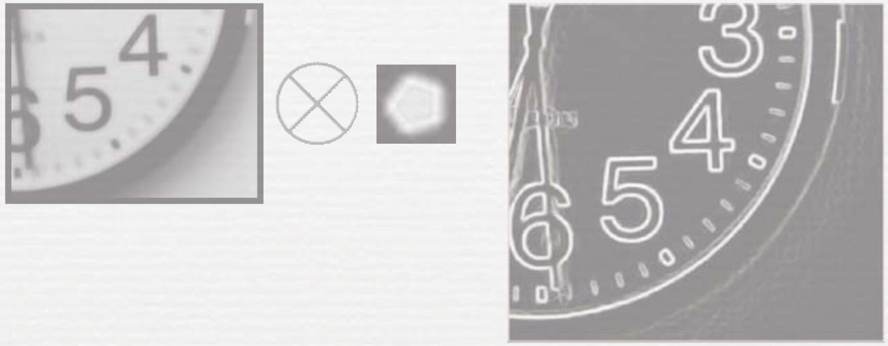
Big ideas in Comp Photo

- Multiple-exposure & multiplexing
- Coded imaging
- **Prior information**
- The raw data is high dimensional
- Active imaging

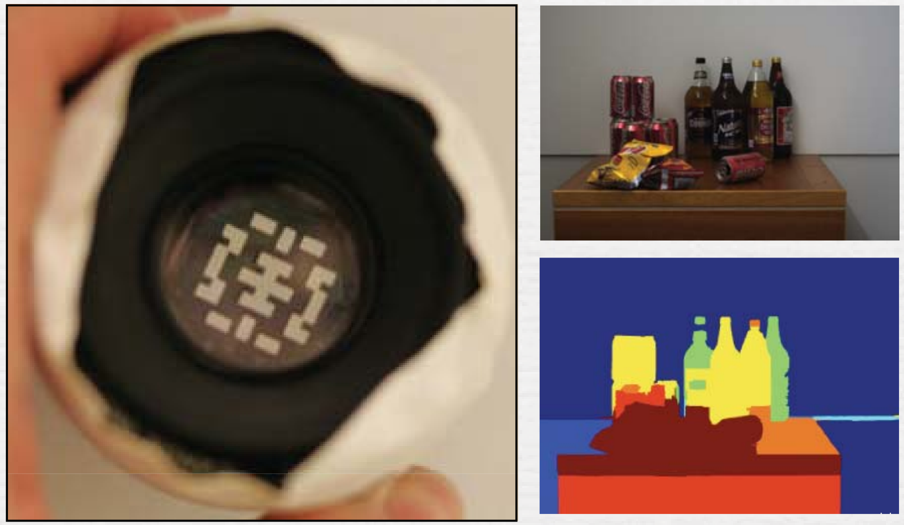


Outline

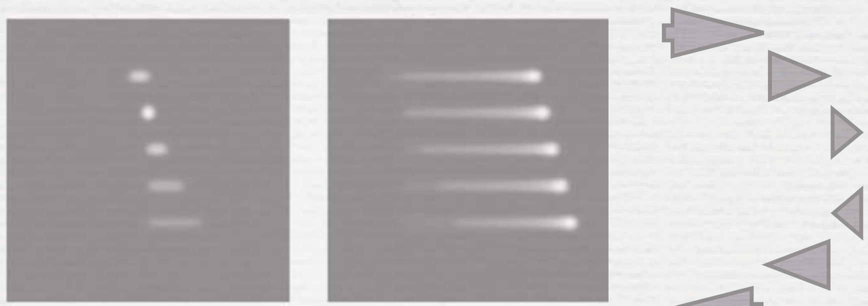
- Natural image prior: help inversion



- **Coded aperture: make blur vary more with depth**



- Motion-invariant photography



(c) Static camera (d) Parabolic camera

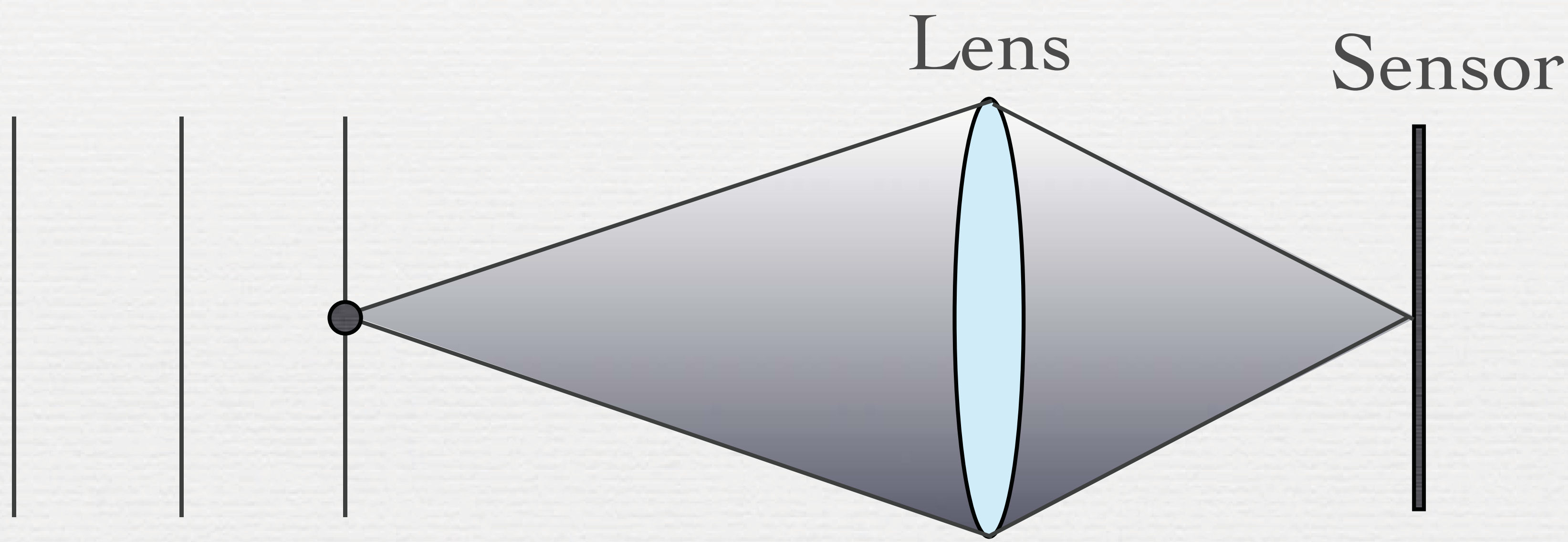
Image and Depth from a Conventional Camera with a Coded Aperture

With Anat Levin, Rob Fergus, Bill Freeman [Siggraph 2007]

RGB & coarse depth from single image

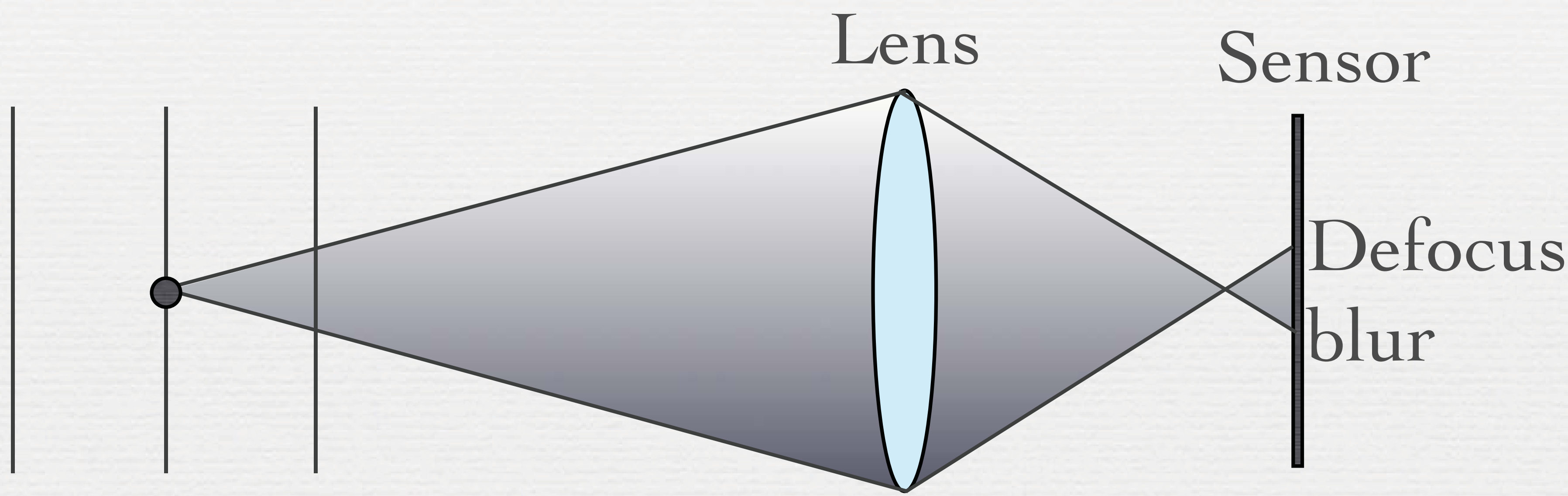
Defocus & depth

- Objects at focusing distance are sharp



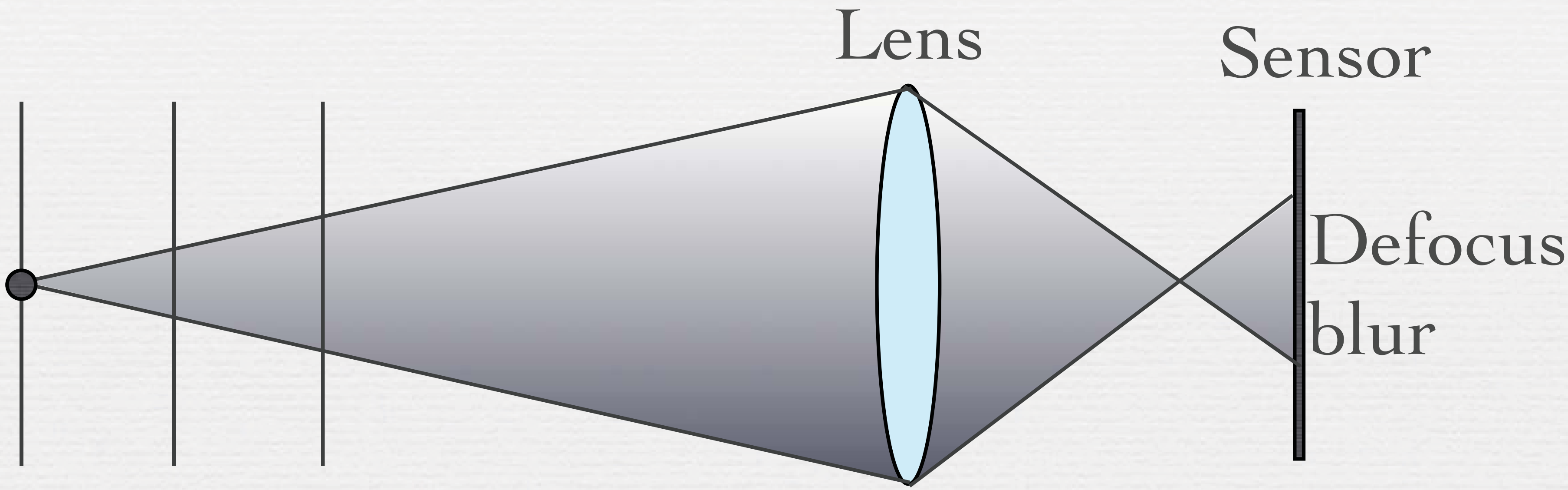
Defocus & depth

- Objects far from focusing distance are blurrier



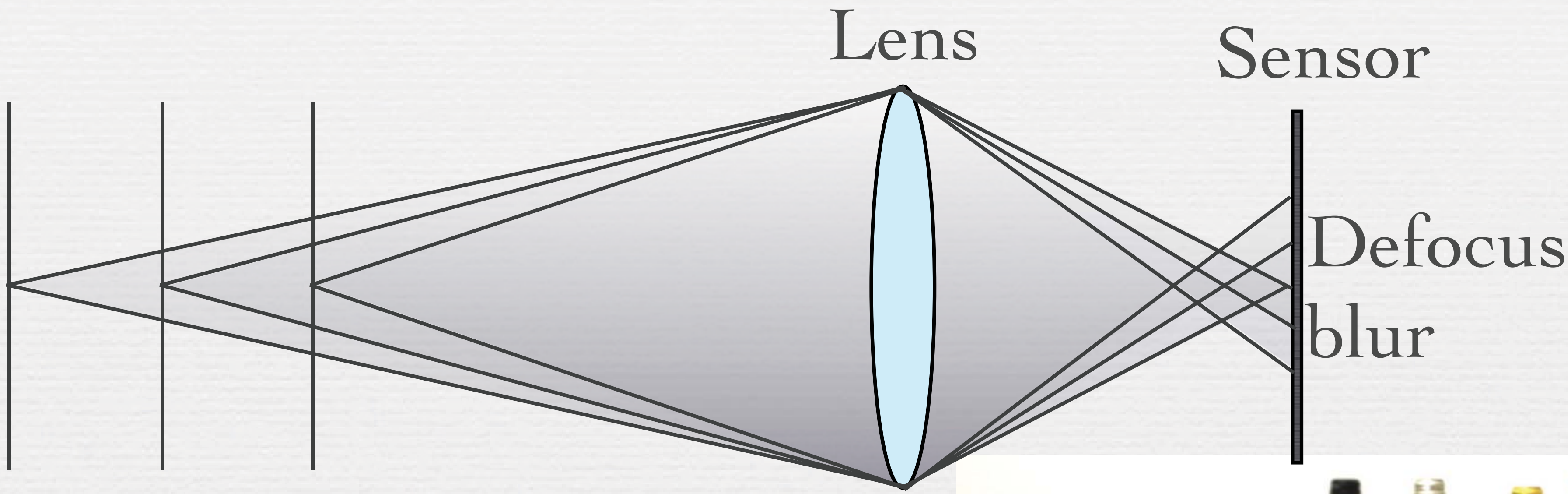
Defocus & depth

- Objects far from focusing distance are blurrier



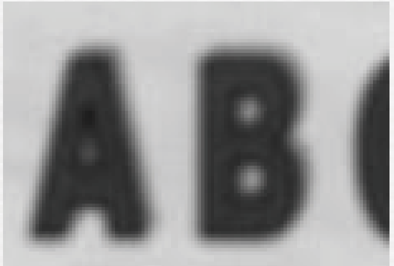
Defocus & depth

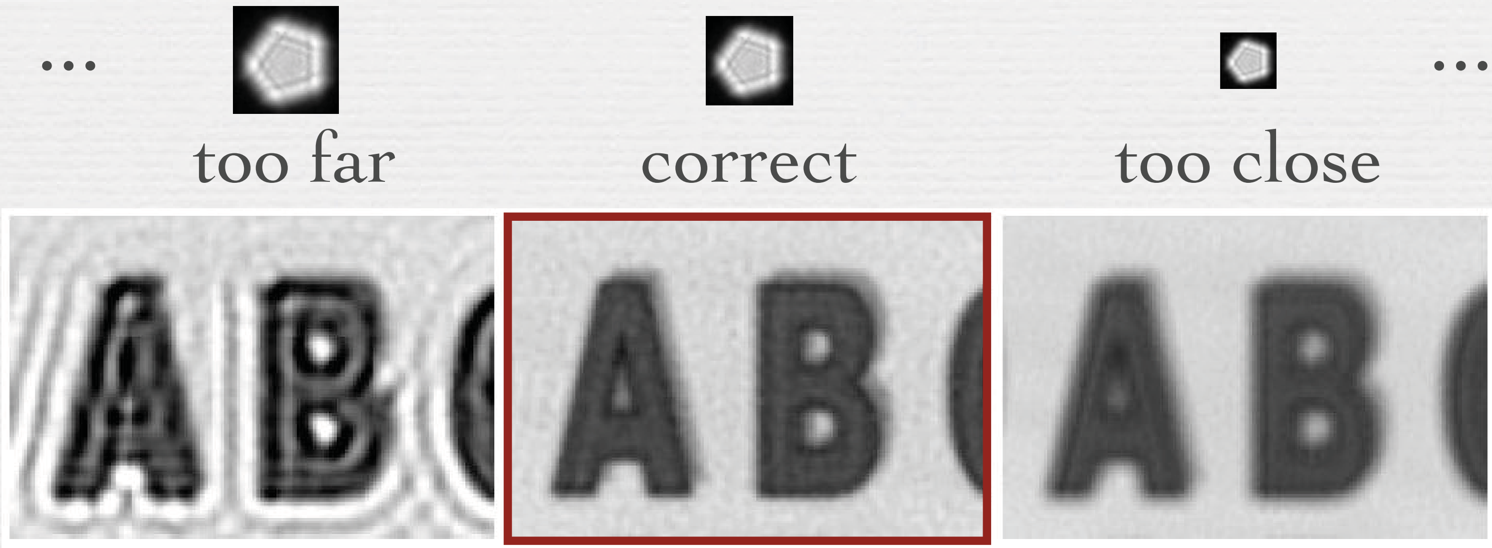
- Objects far from focusing distance are blurrier



- **By inferring blur, we can infer depth**

Strategy

- Input: 
- For each candidate depth,
  try to deconvolve with corresponding kernel

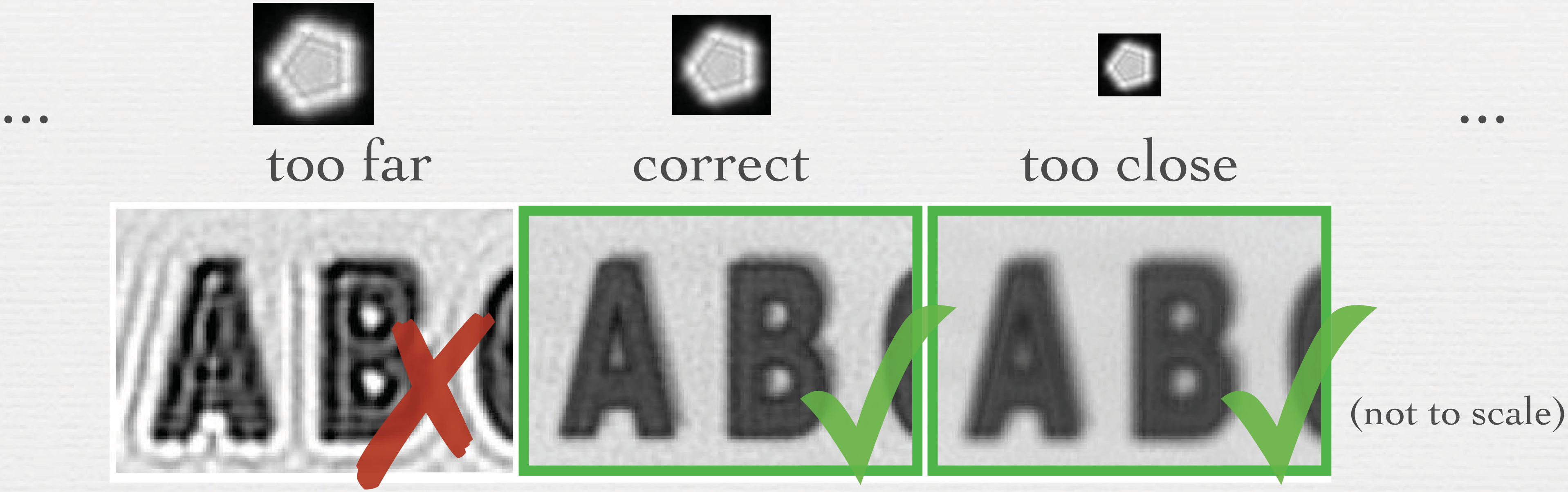


(not to scale)

- For each pixel, keep best kernel/depth

Challenge: hard to infer depth

- For each candidate depth,
  try to deconvolve with corresponding kernel



- **“Too close” not so different from “correct”**

Solution: Coded aperture

- Put a mask (code) on aperture plane (diaphragm)
  - ➡ more structured blur
  - ➡ easier to identify kernel/depth
  - ➡ easier to remove blur

Conventional

Coded

Build your own coded aperture

Open the lens

Open the lens

Open the lens

Open the lens

Open the lens

Open the lens

Open the lens

Now the critical part

Cardboard mask

Cardboard mask

Cardboard mask

Cardboard mask

Cardboard mask

Cardboard mask

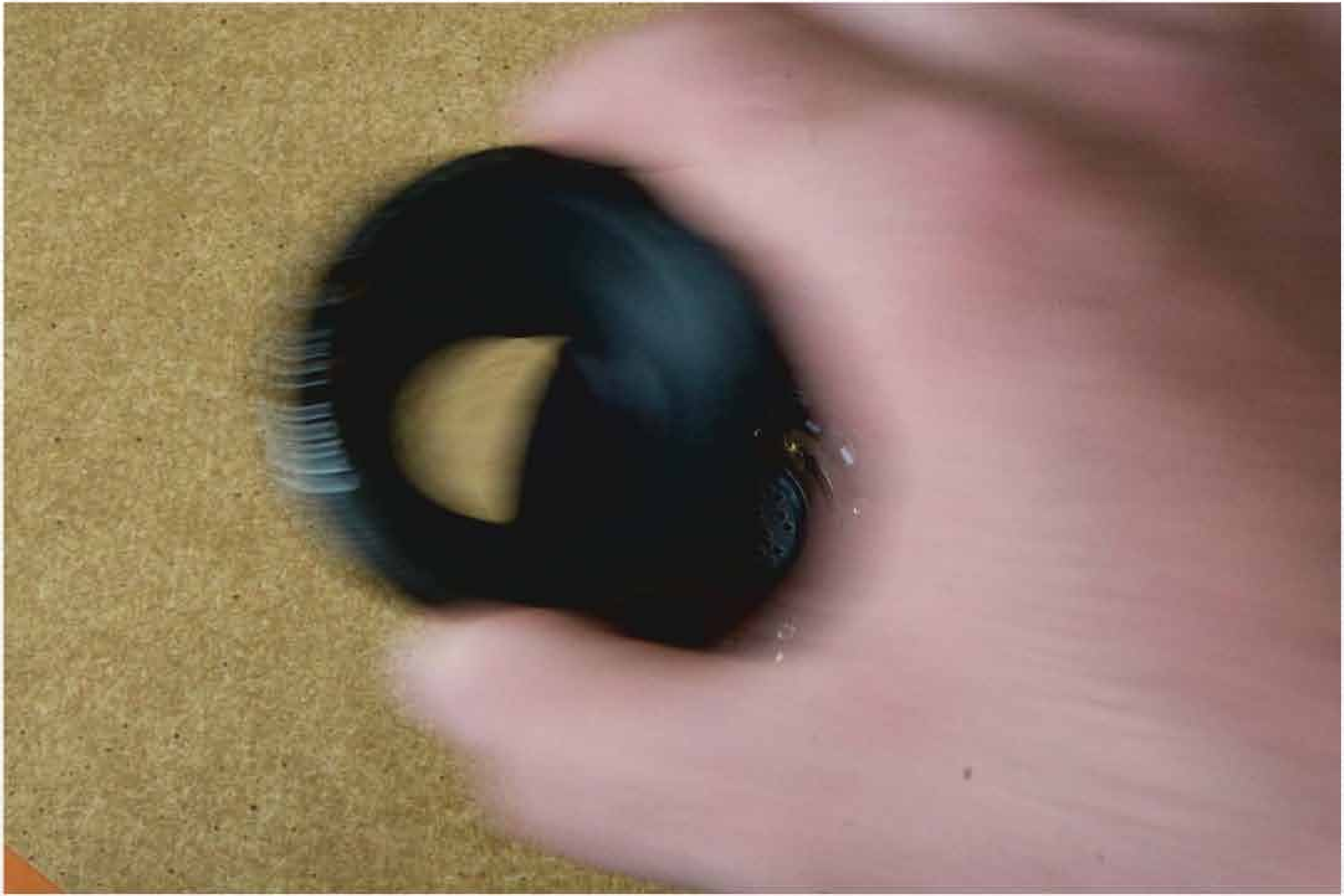
Cardboard mask

Close it up

Close it up

Close it up

Close it up



Close it up

Close it up

Close it up

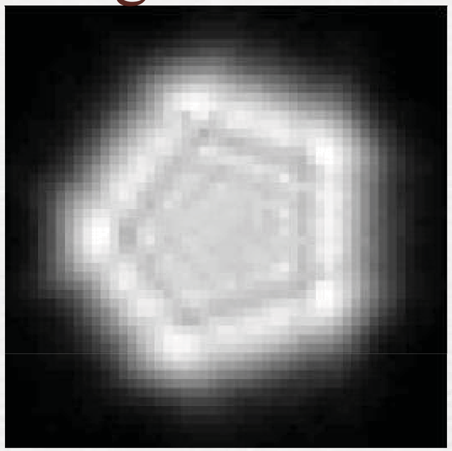
Voilà!

Traditional aperture defocus

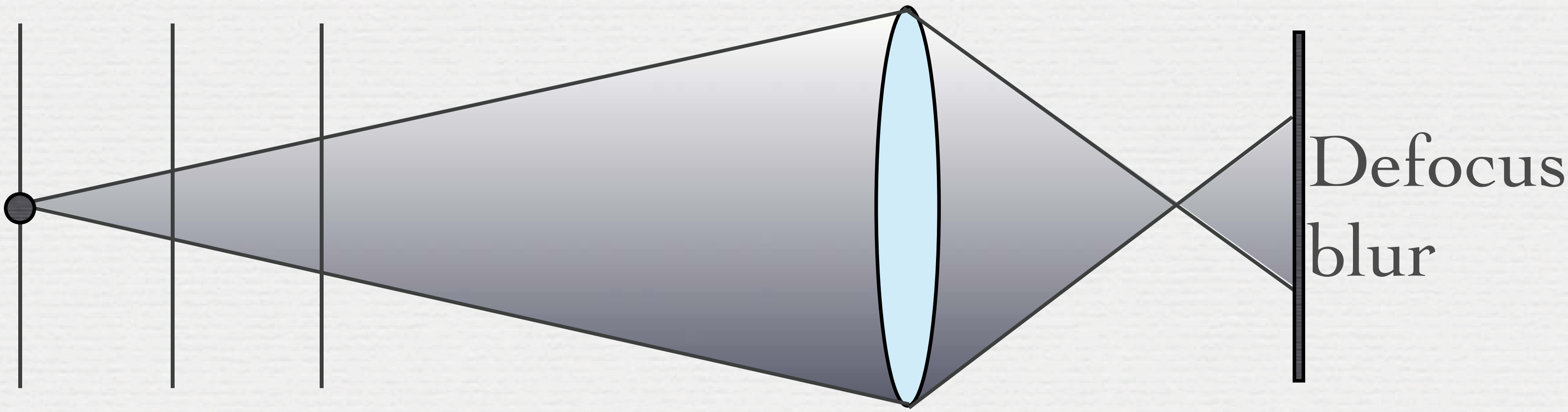
- Shape of mask scaled according to depth

Image of defocused point light

Conventional



Conventional



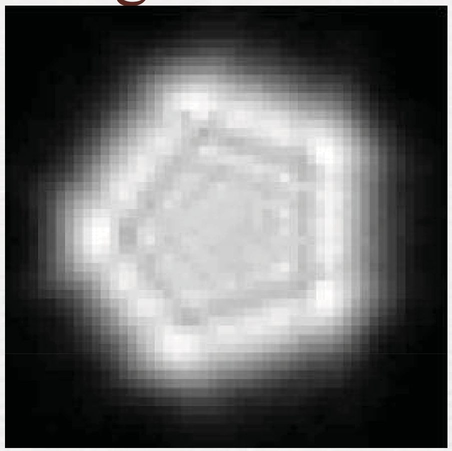
Coded aperture defocus

- Shape of mask scaled according to depth

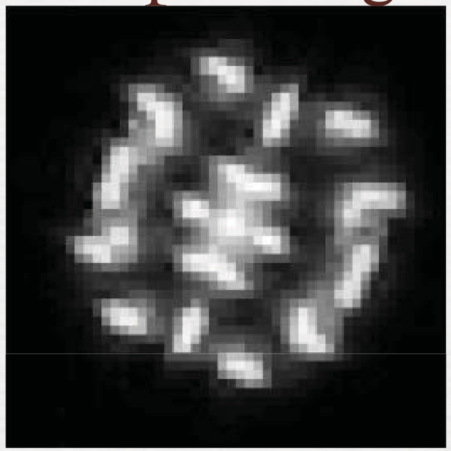
Image of defocused point light

Conventional

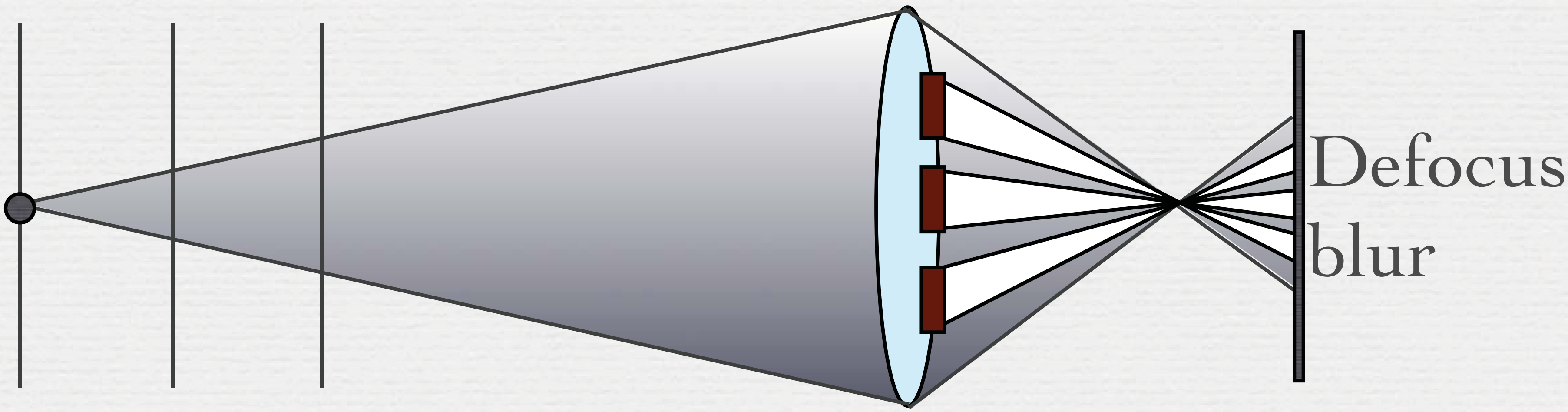
Coded



Conventional

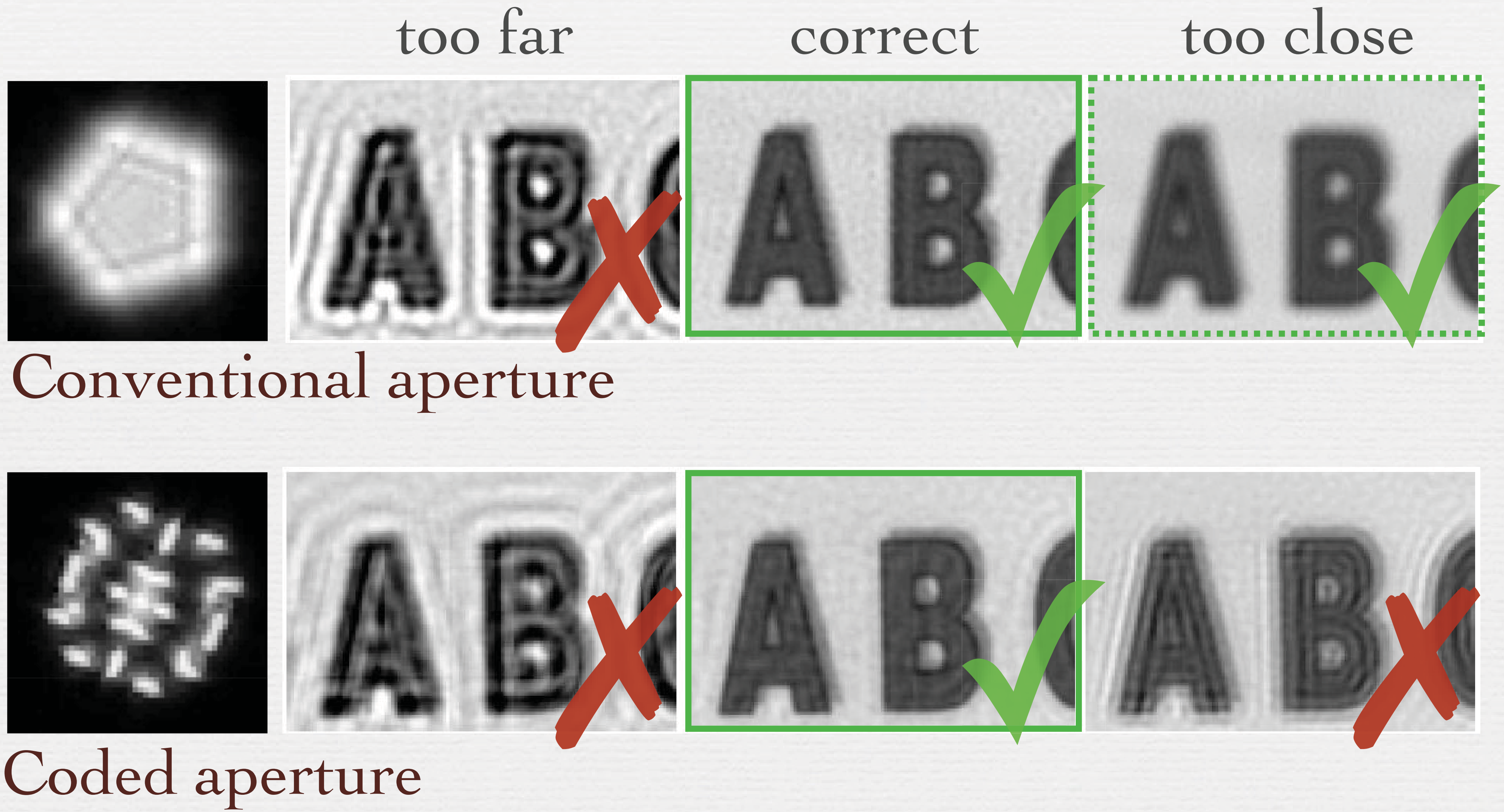


Coded



Why code helps

- Wrong blur is making more mess



Input

Deconvolved (all-focus)

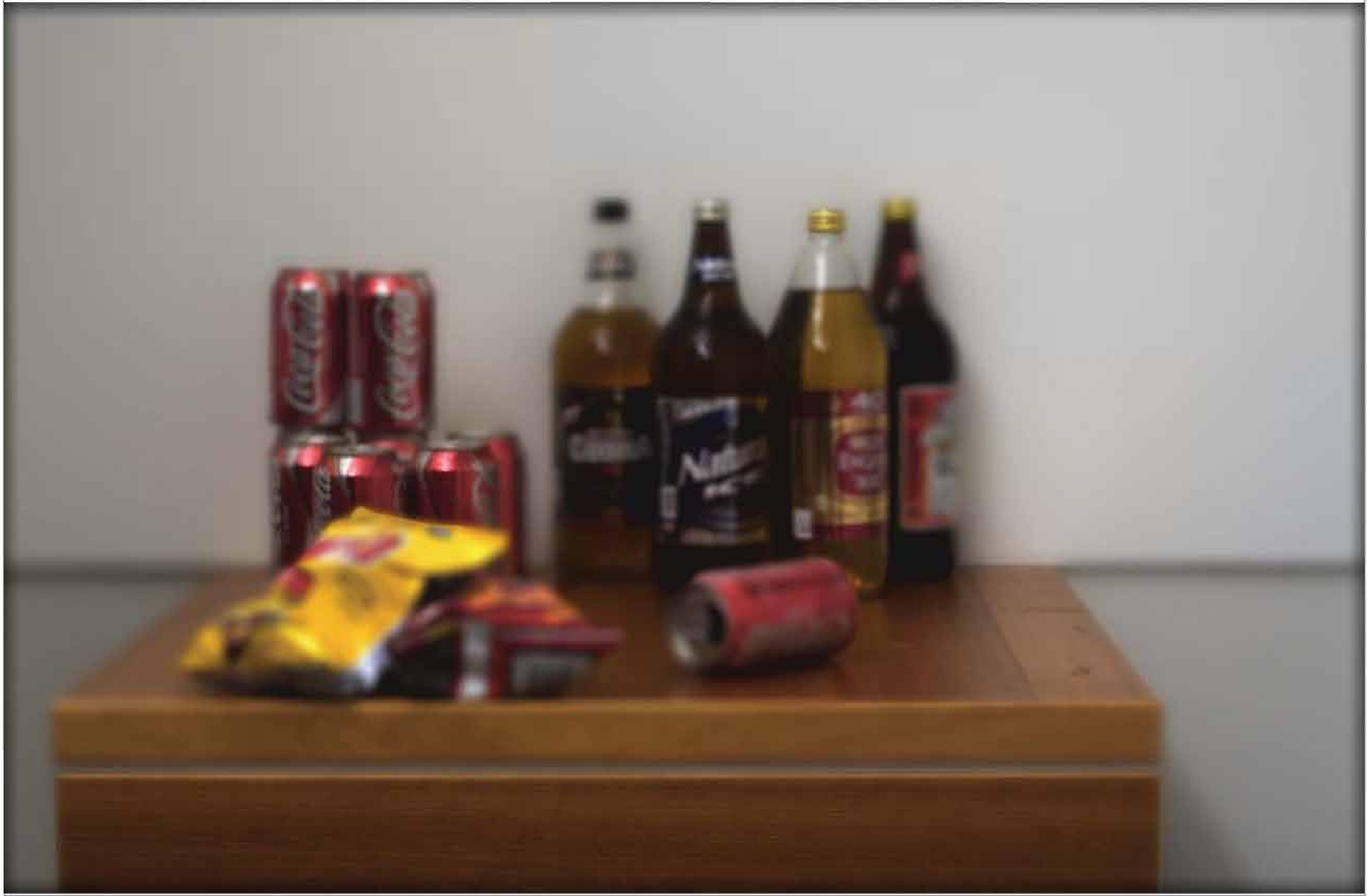
Close up

Depth

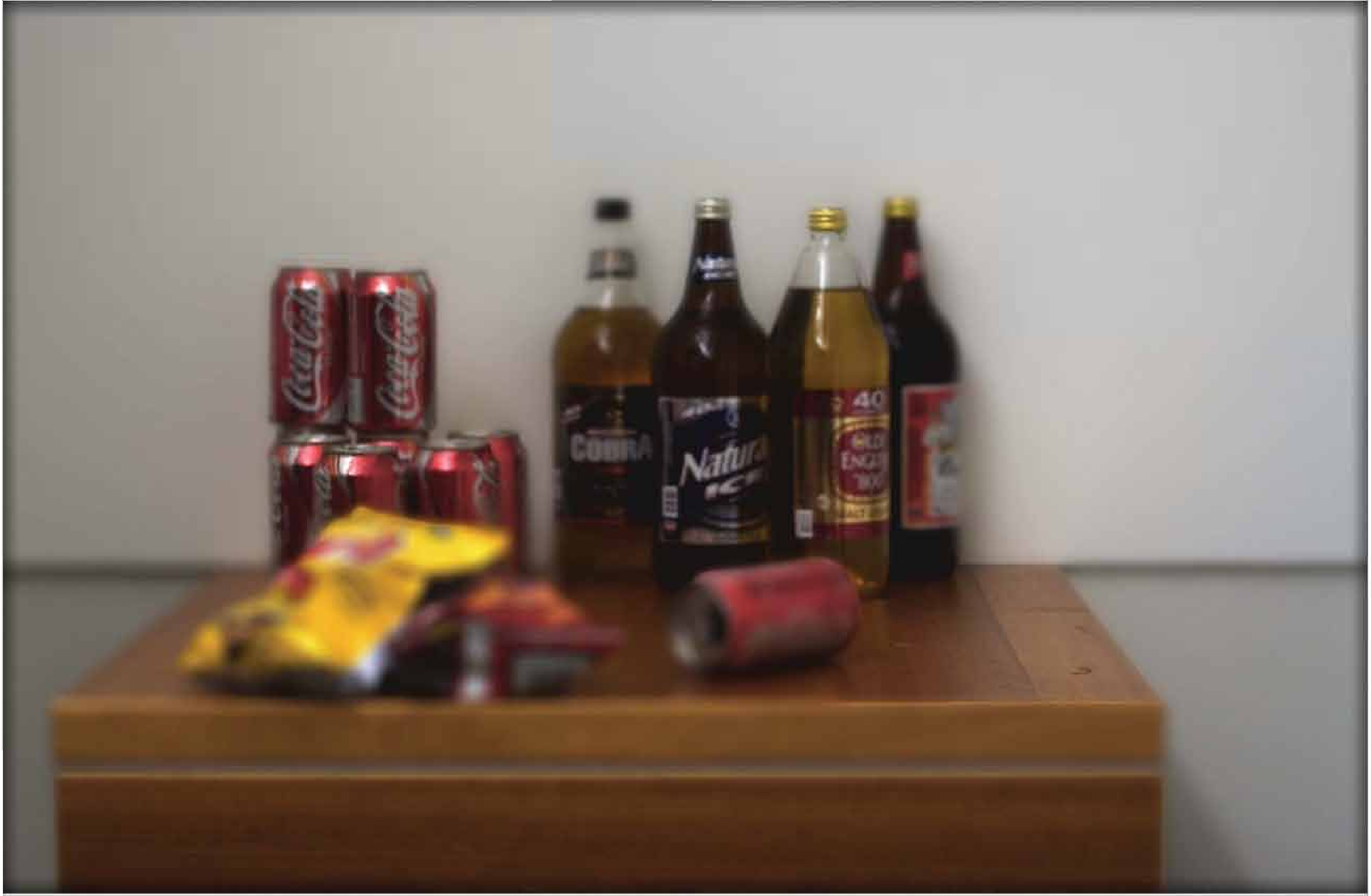
Deconvolved (all-focus)

Refocusing (from single image!)

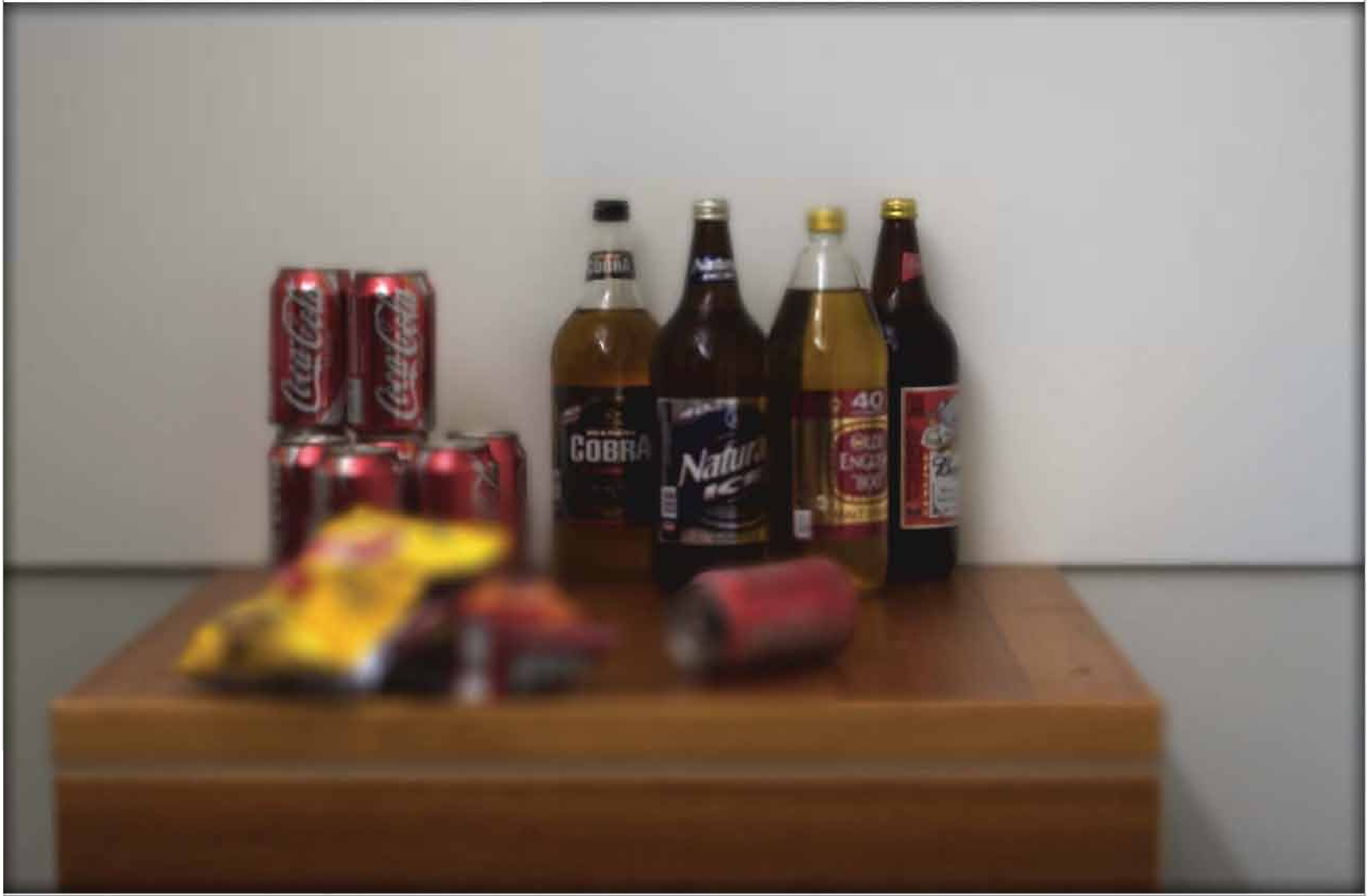
Refocusing (from single image!)



Refocusing (from single image!)



Refocusing (from single image!)

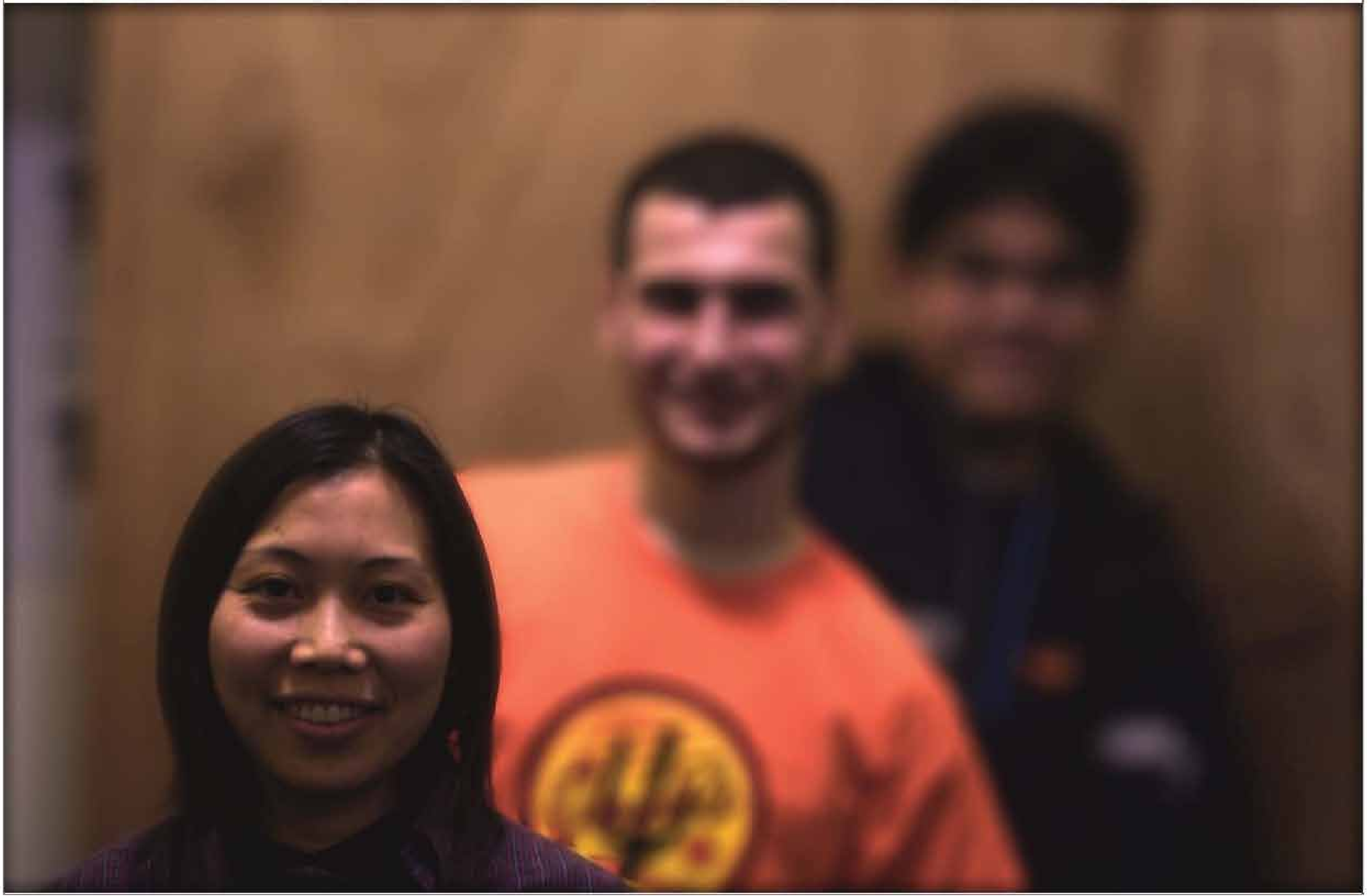


Results

Input

Deconvolved

Refocusing (from single image!)

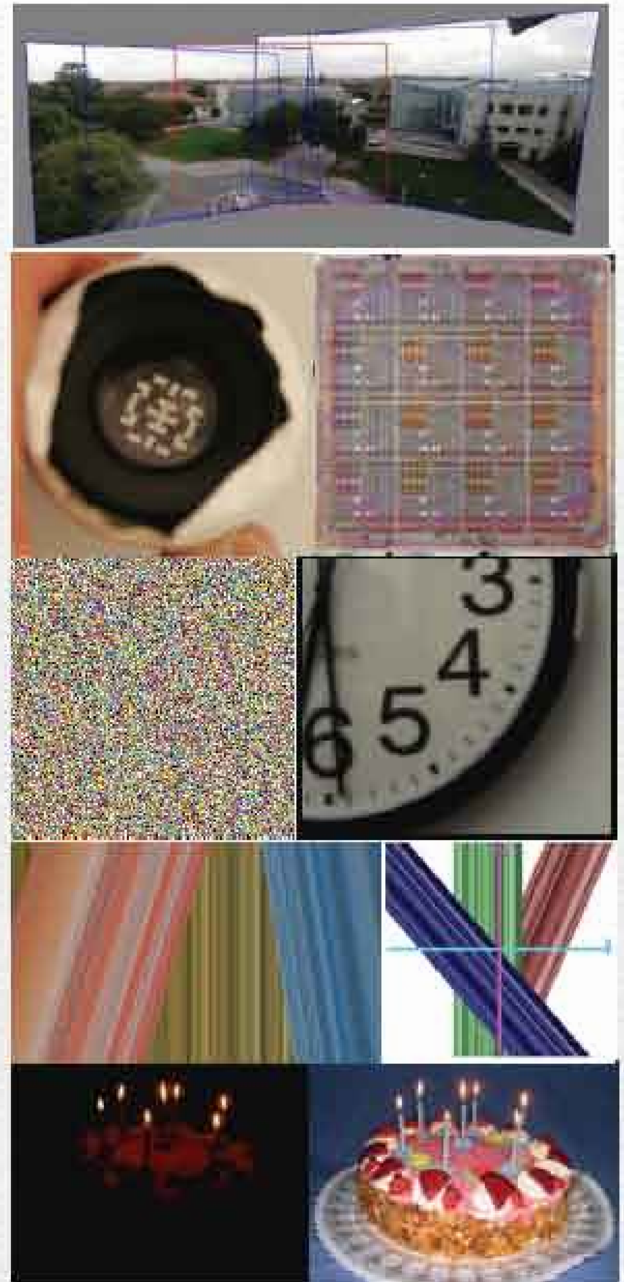


Refocusing (from single image!)

Refocusing (from single image!)

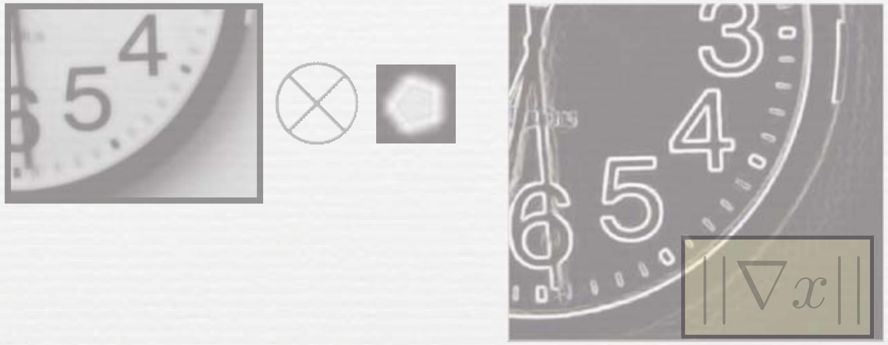
Big ideas in Comp Photo

- Multiple-exposure & multiplexing
- **Coded imaging**
- **Prior information**
- The raw data is high dimensional
- Active imaging

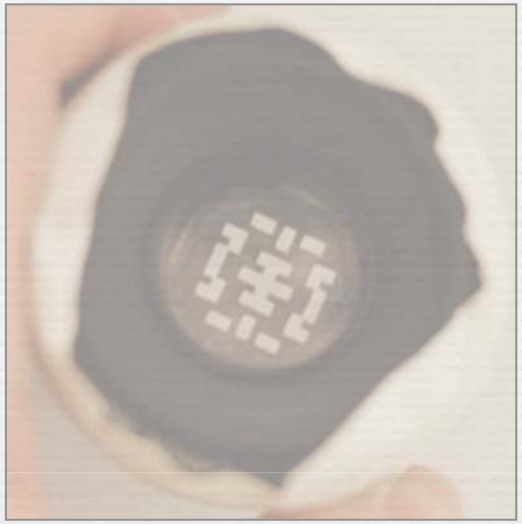


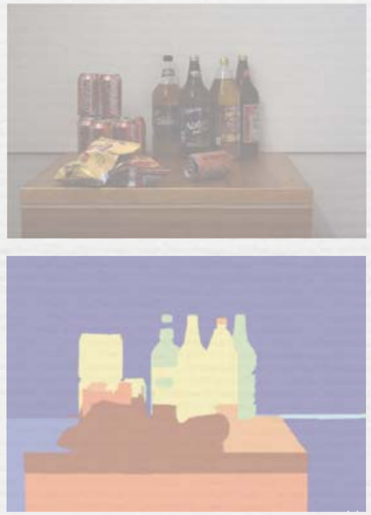
Outline

- Natural image prior



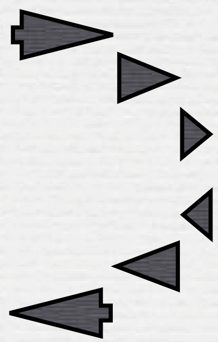
- Coded aperture: make blur vary more with depth





- **Motion-invariant photography**

(c) Static camera (d) Parabolic camera



Counter intuitive solution:

To reduce motion blur, increase it!

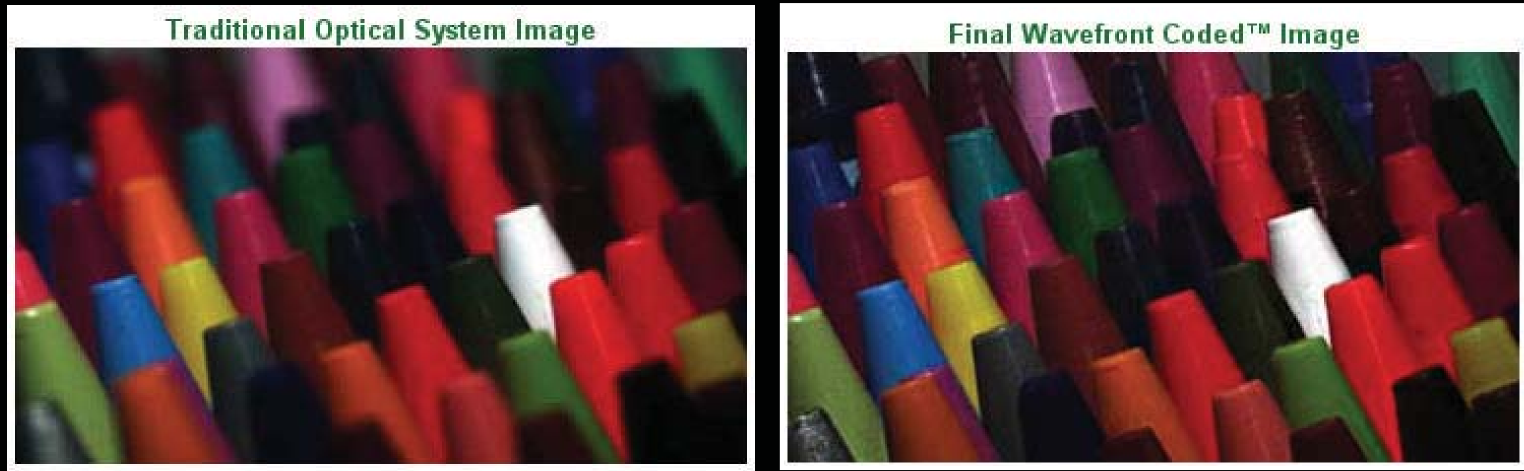
- move camera as picture is taken

- Makes blur invariant to motion- can be removed with spatially uniform deconvolution
  - kernel is known (no need to estimate motion)
  - kernel identical over the image (no need to segment)
- Makes blur easy to invert

Inspiration: depth invariant defocus

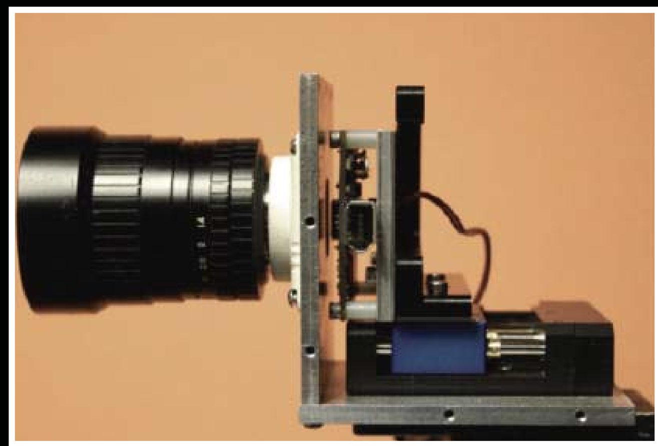
- Wavefront coding - manipulate optical element

    Cathey and Dowski 94



- Vary object/detector distance during integration

    - Hausler 72
    - Nagahara, Kuthirummal, Zhou, Nayar 08



Motion invariant blur- disclaimers:

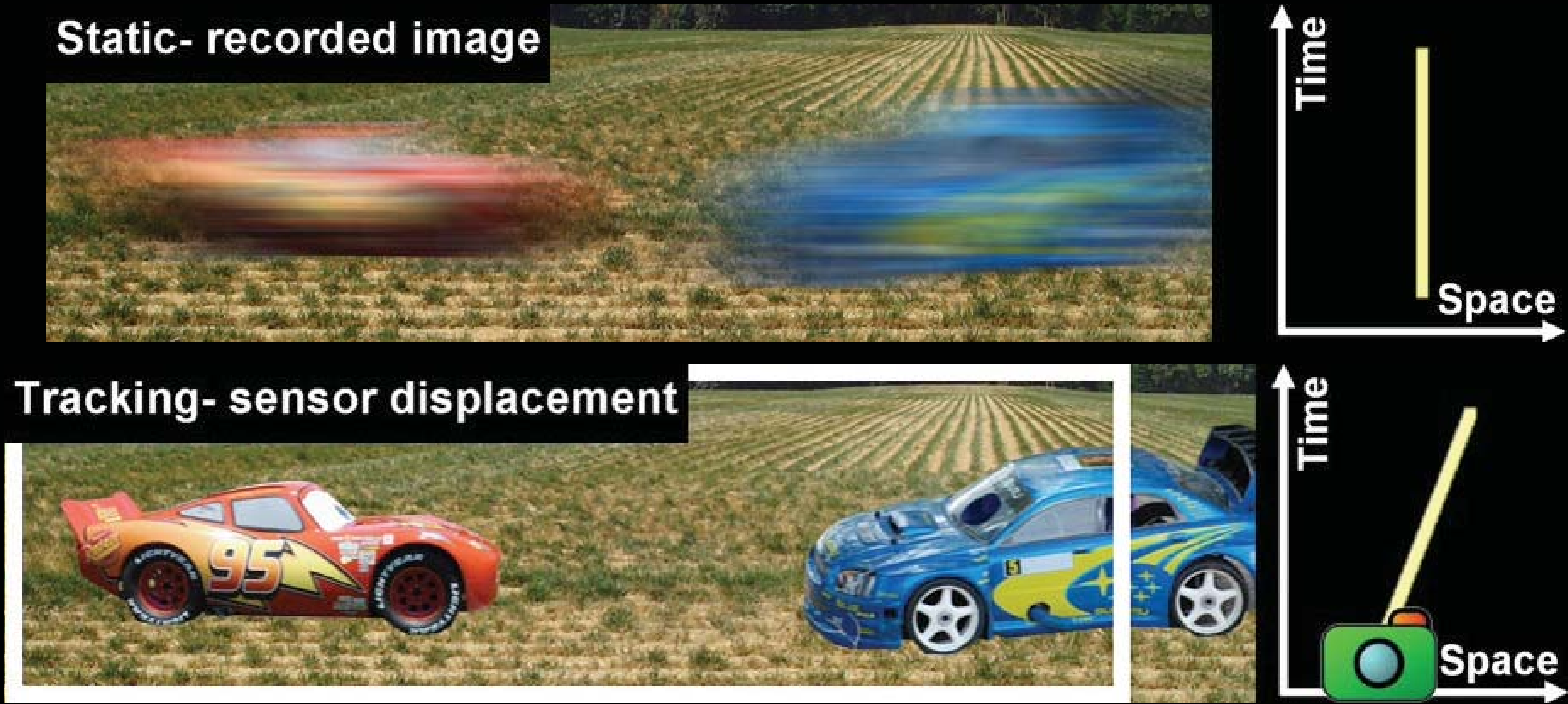
- Assumes 1D motion (e.g. horizontal)
- Degrades quality for static objects

Controlling motion blur

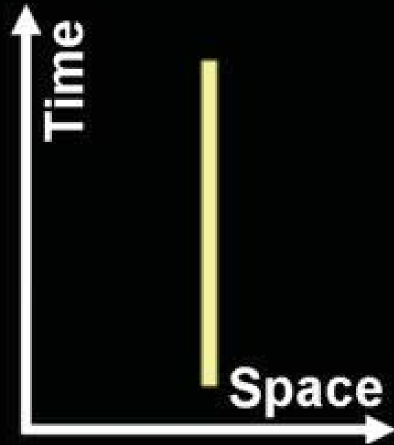
Controlling motion blur

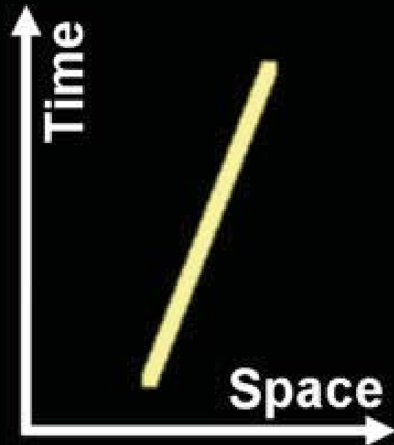
Can we control motion blur?

Controlling motion blur

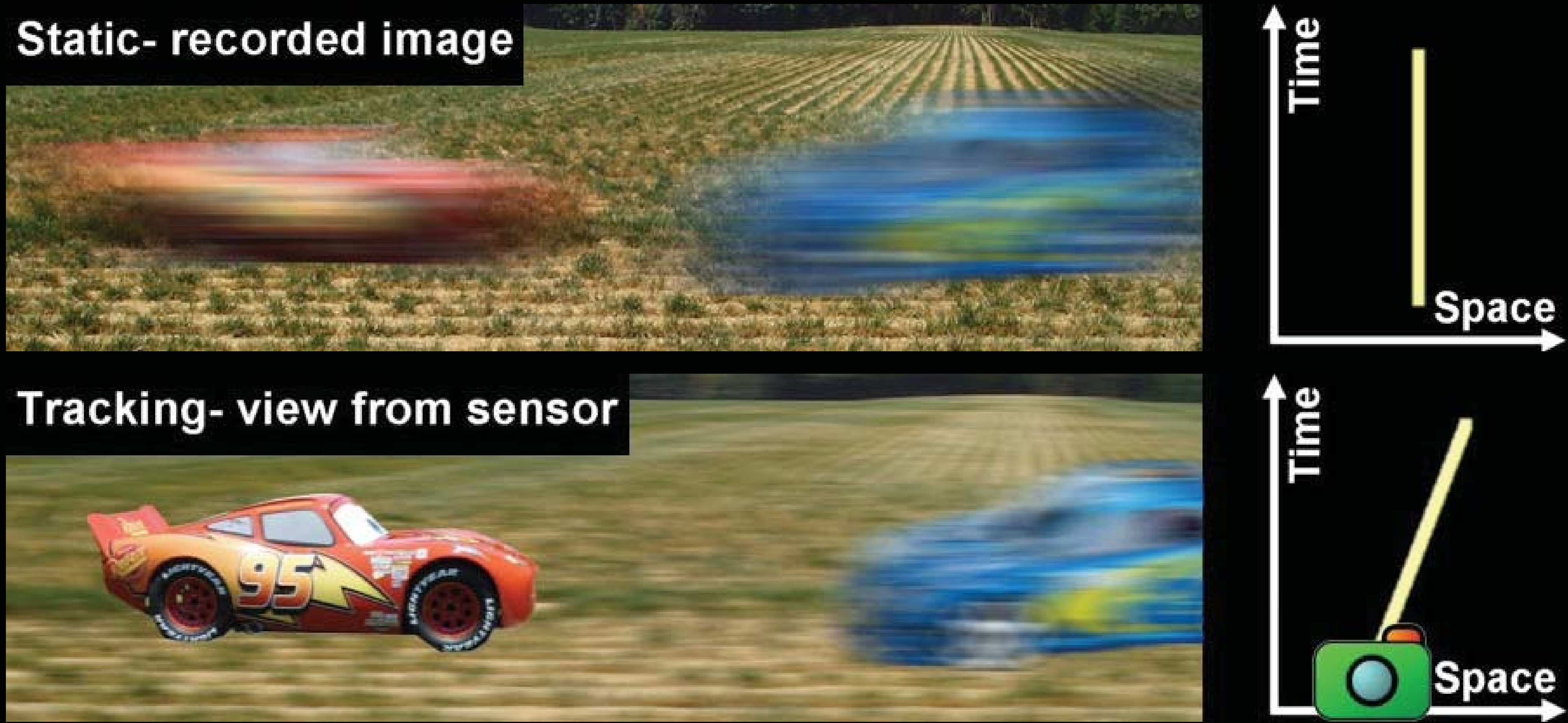


Controlling motion blur

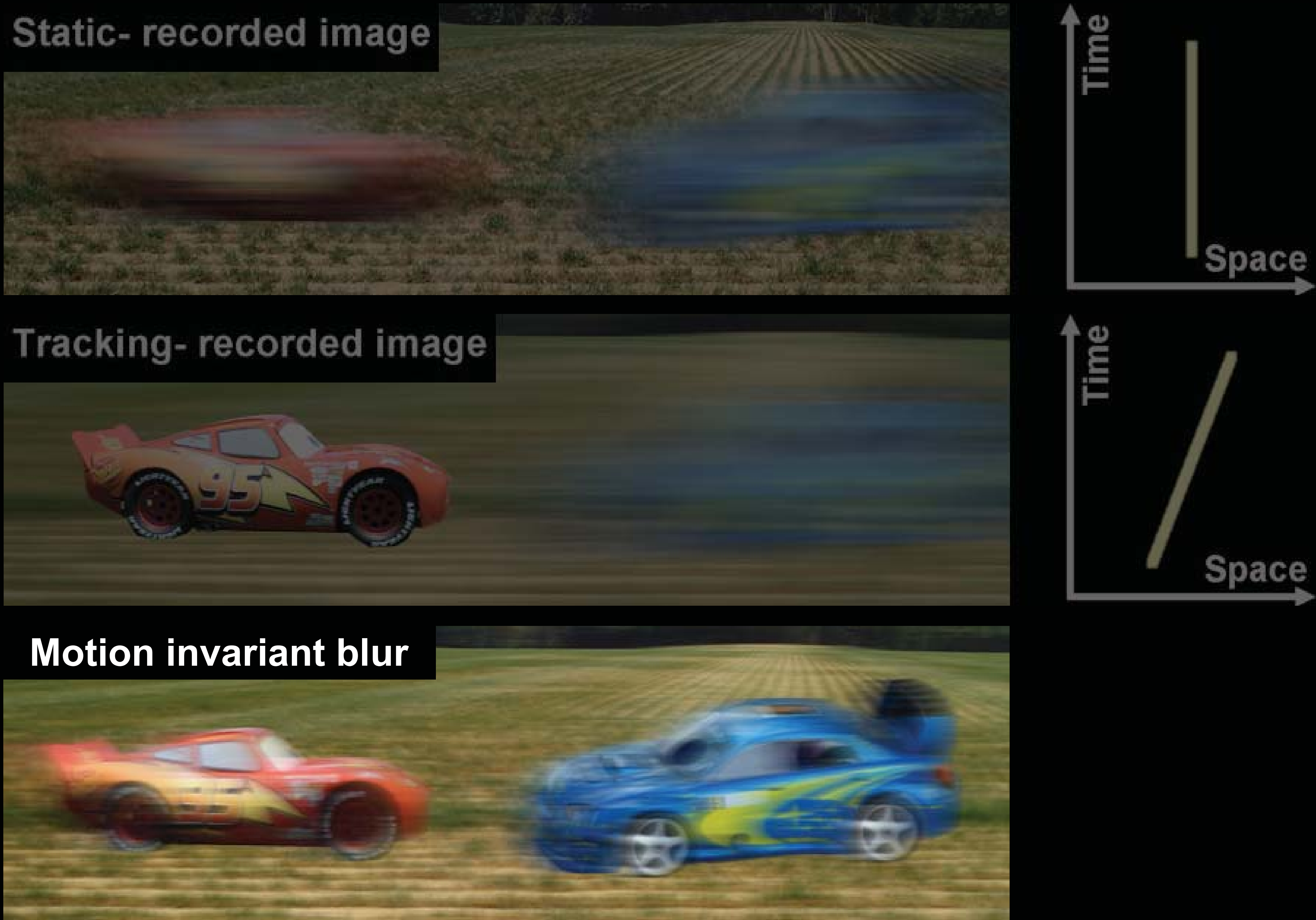




Controlling motion blur



Controlling motion blur



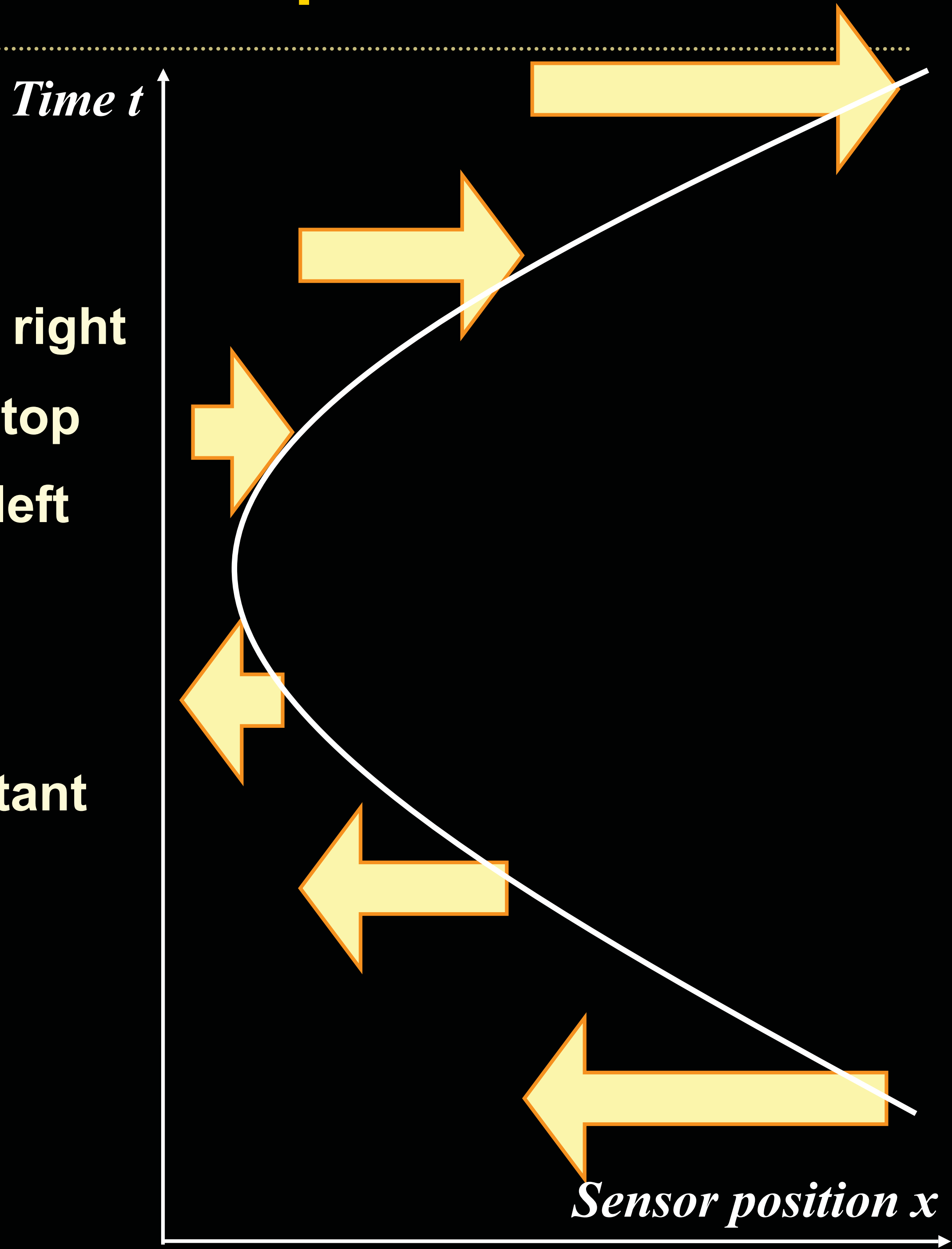
Parabolic sweep

Sensor position $x(t)=a\ t^2$

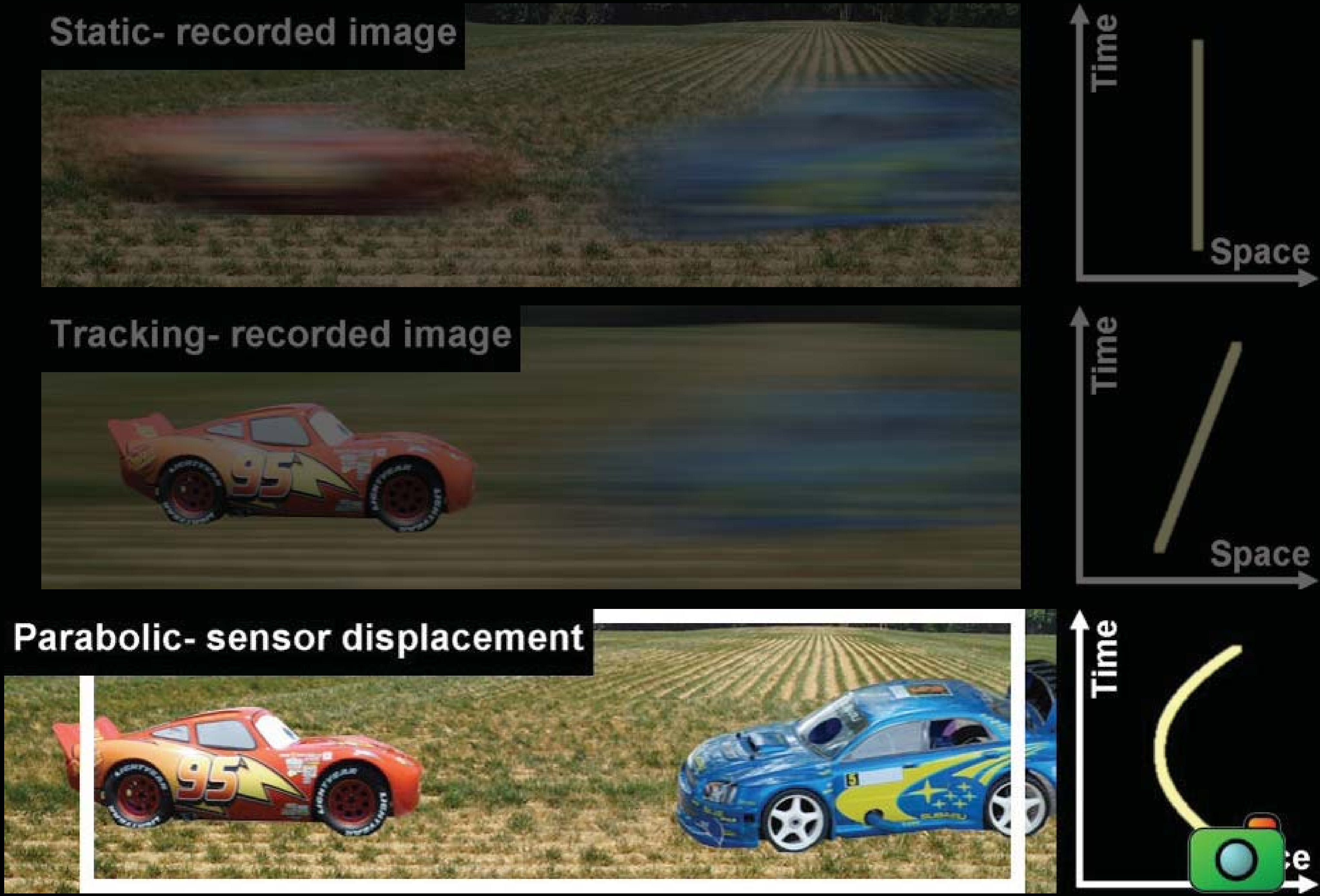
- Start by moving very fast to the right
- Continuously slow down until stop
- Continuously accelerate to the left

Intuition:

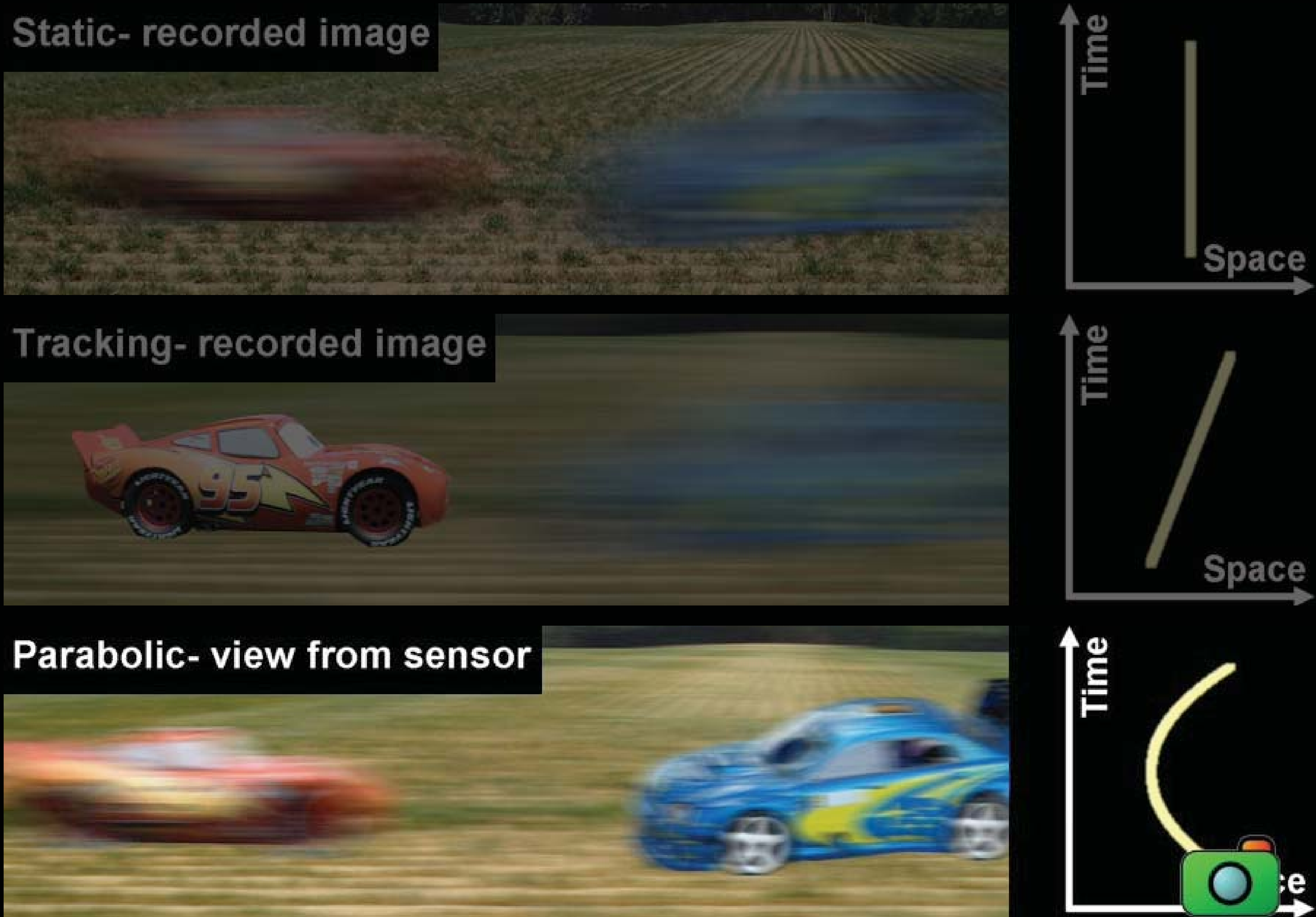
For any velocity, there is one instant where we track perfectly.



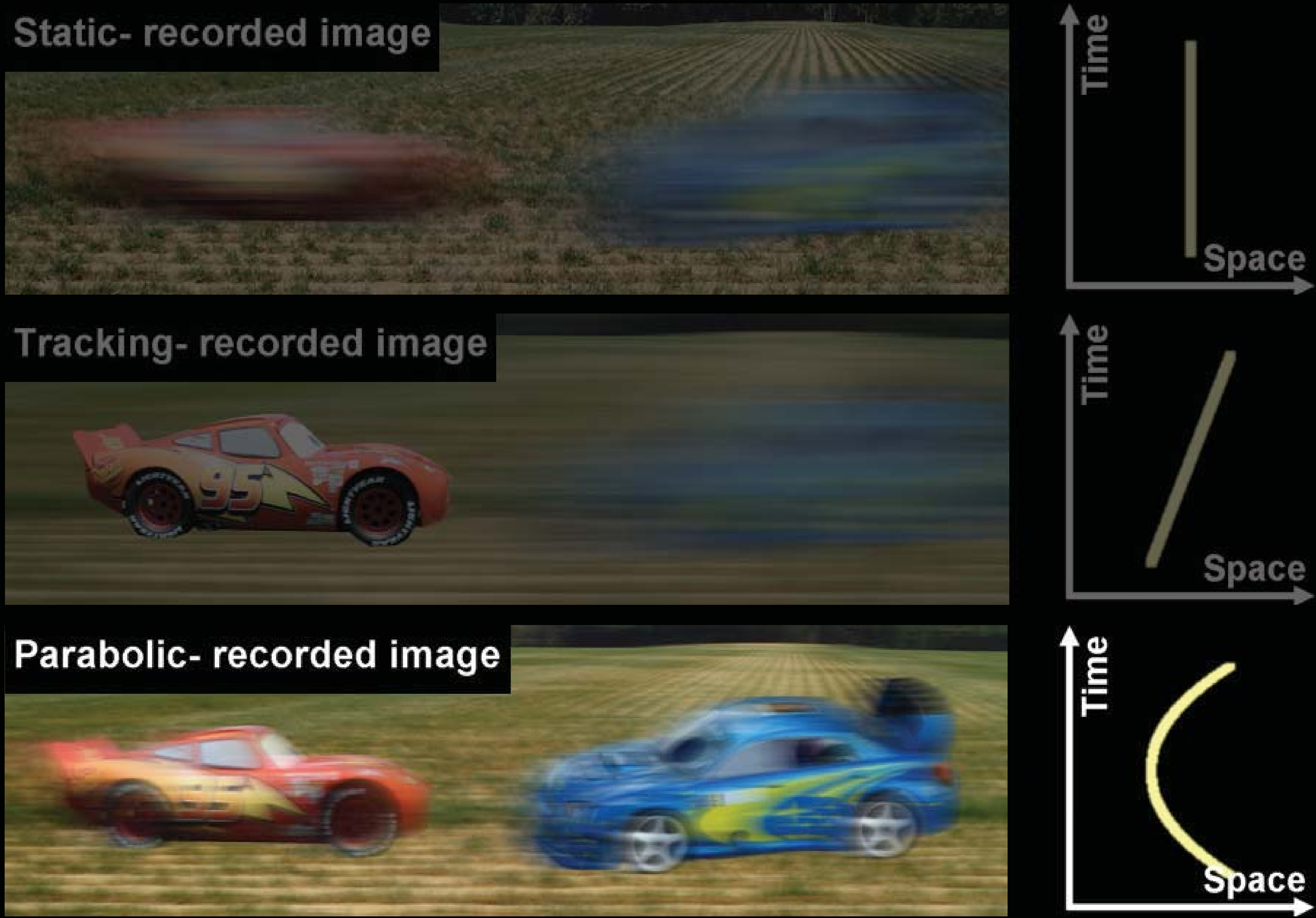
Motion invariant blur

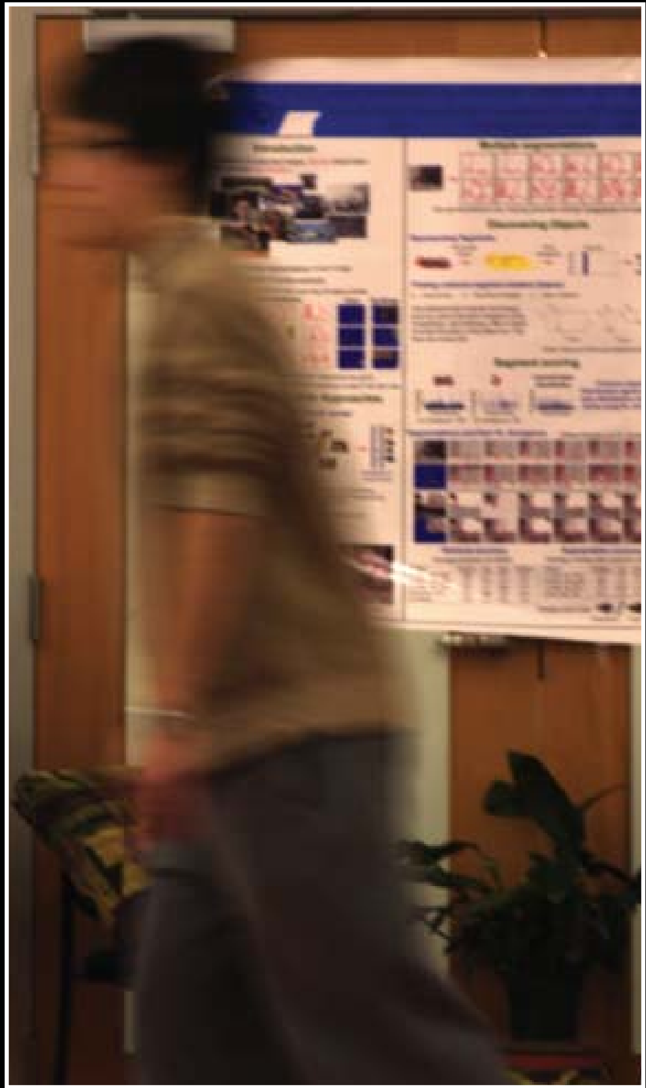


Motion invariant blur



Motion invariant blur



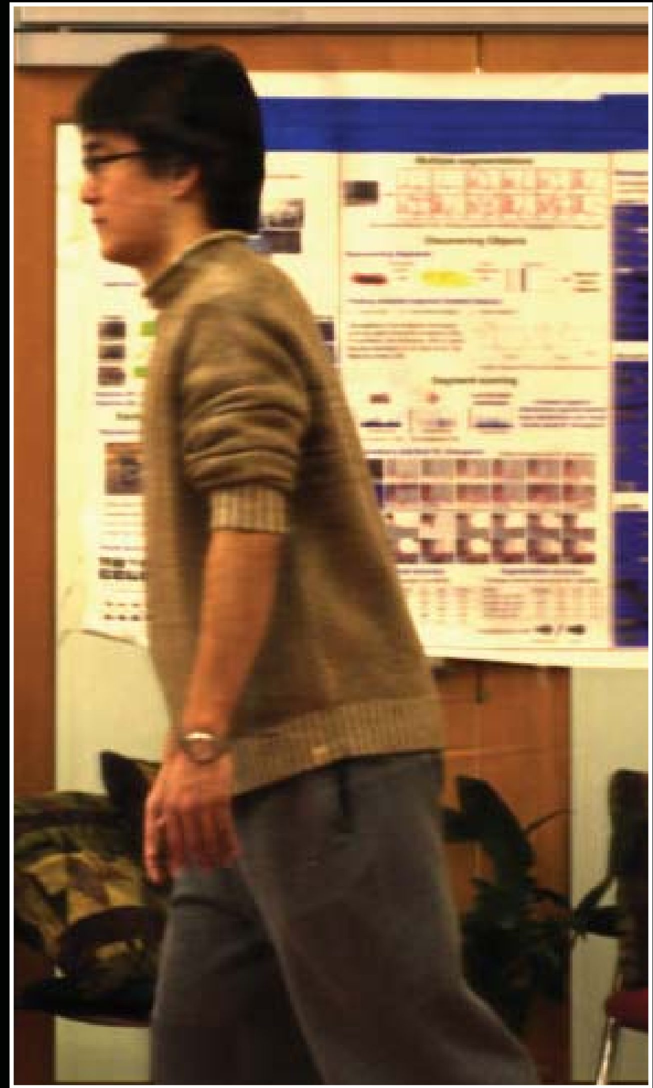


Static camera

Unknown and variable blur kernels

Our parabolic input

Blur kernel is invariant to velocity

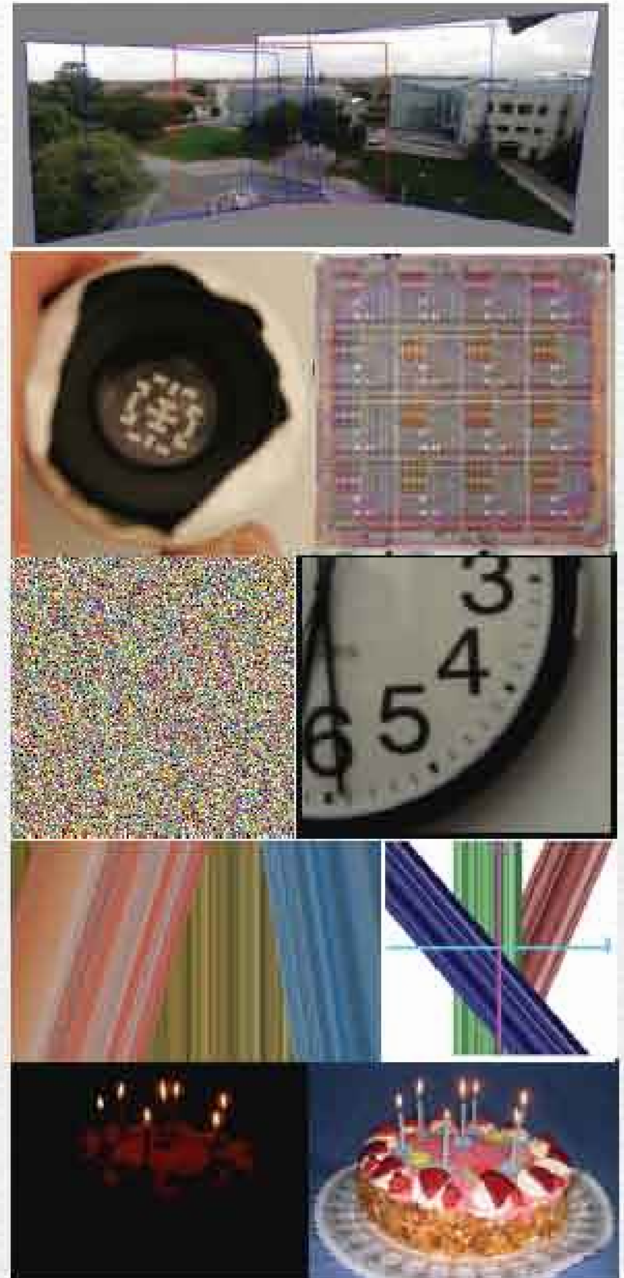


Our output after deblurring

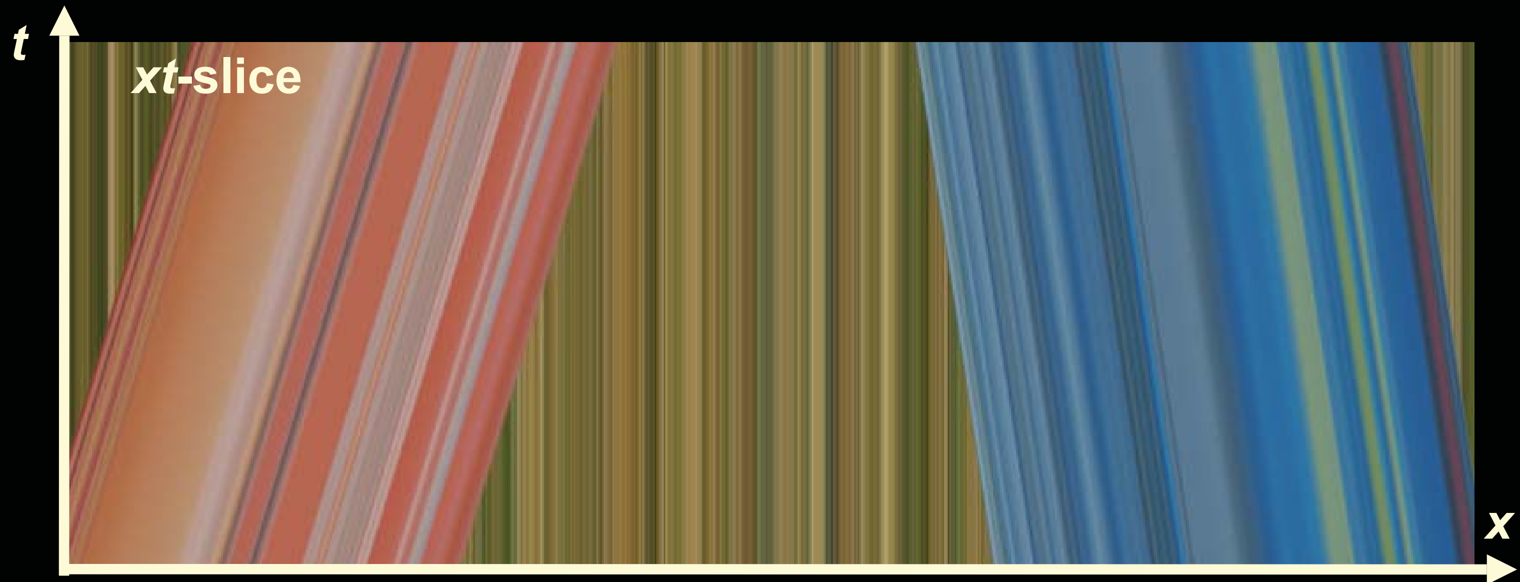
NON-BLIND deconvolution

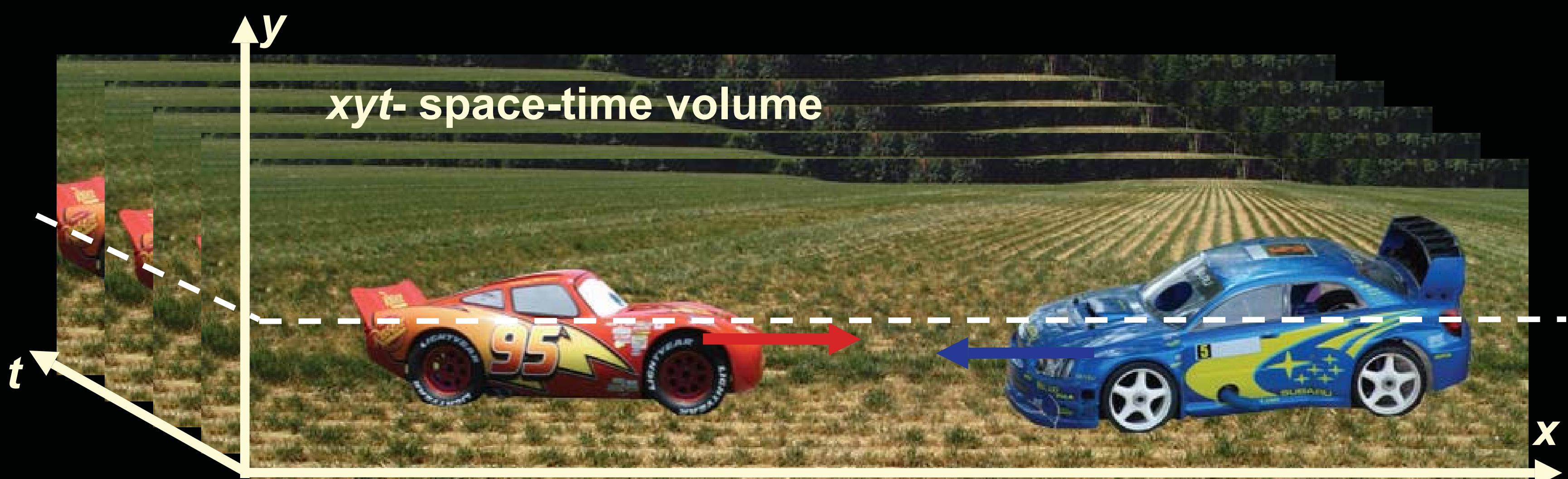
Big ideas in Comp Photo

- Multiple-exposure & multiplexing
- **Coded imaging**
- Prior information
- The raw data is high dimensional
- Active imaging



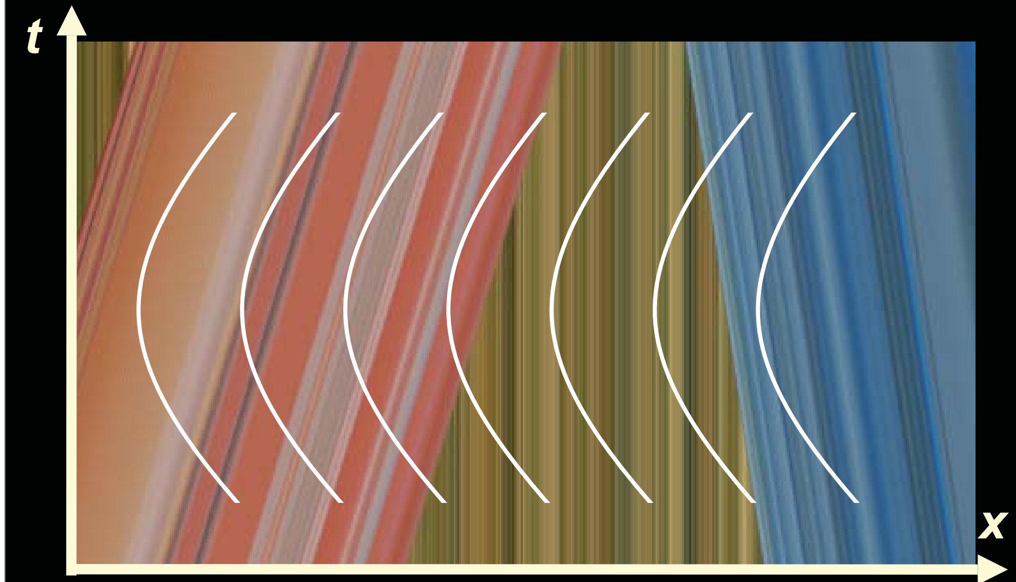
The space time volume



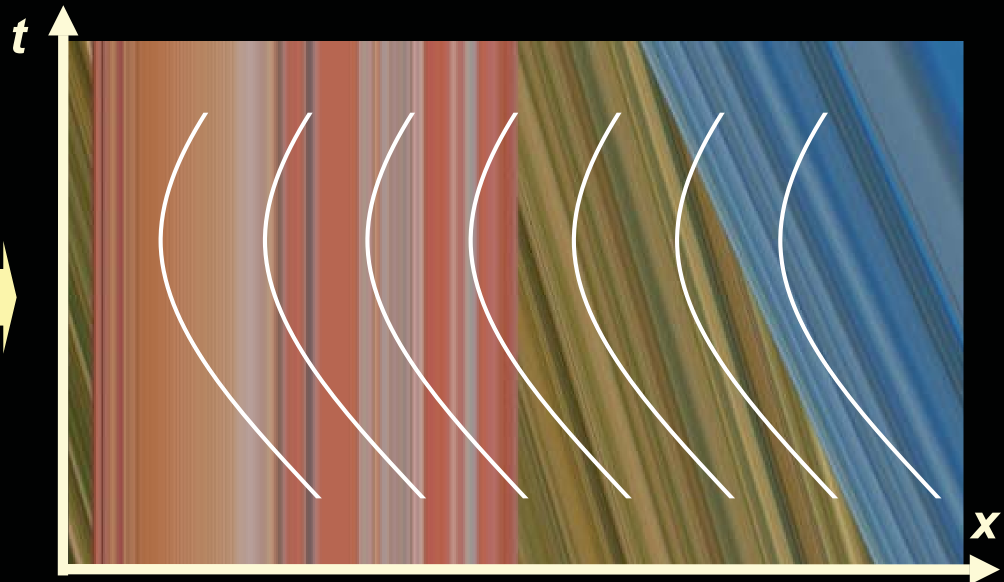


Solution: parabolic curve – shear invariant

Static object coordinates



Moving object coordinates



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

Sheared parabola ⇒ **Shifted parabola**

Deblurring and information loss

Assume: we could perfectly identify blur kernel

Which camera has motion blur that is easy to invert?
- Static? Flutter Shutter? Parabolic?

Prove: parabolic motion achieves near optimal information preservation

blurred input

deblurred

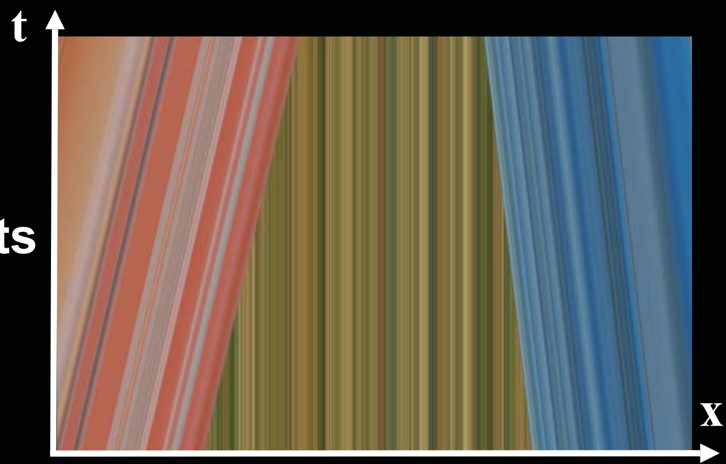
static input

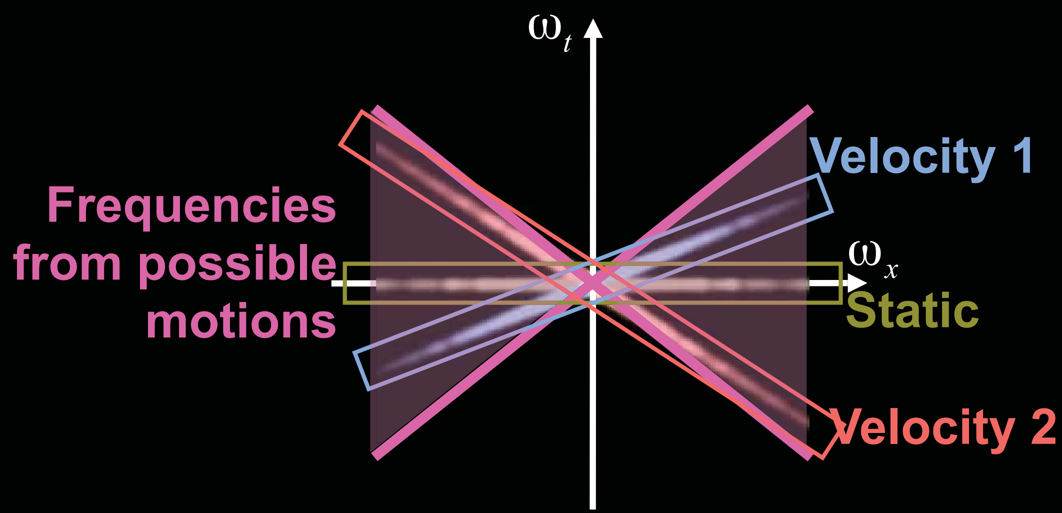
Space-time Fourier domain

Primal Domain

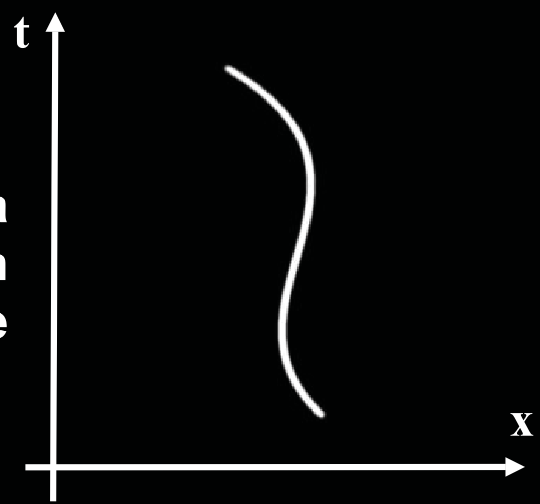
Frequency Domain

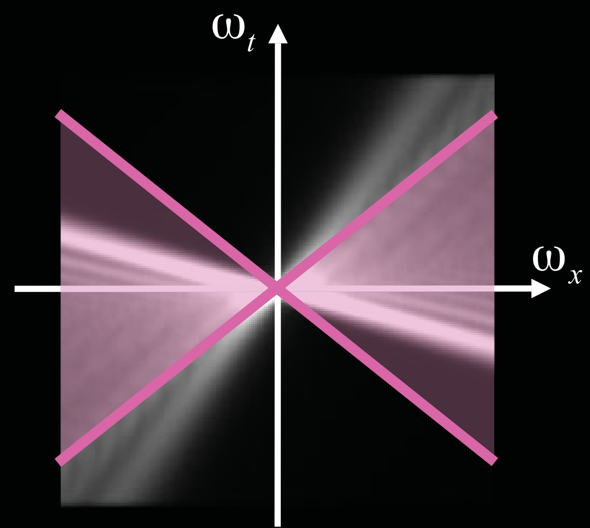
Objects





Camera integration curve



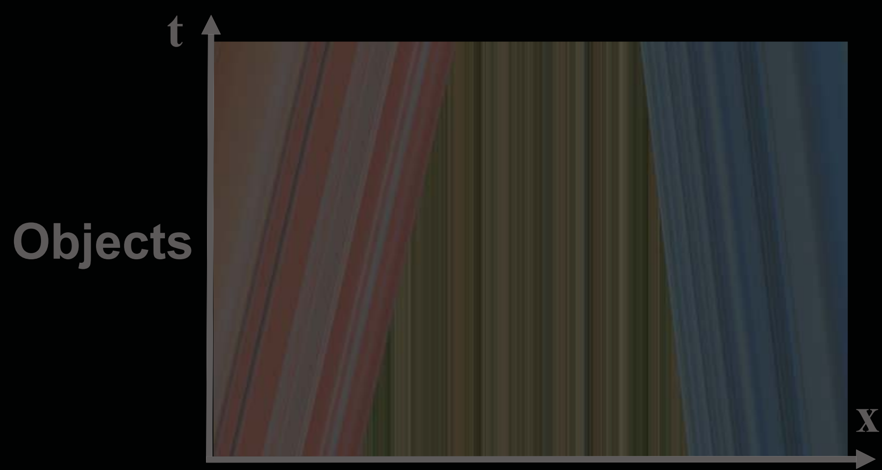


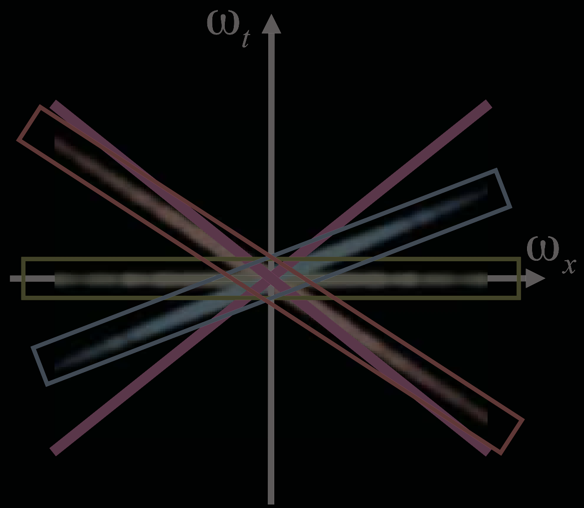
Bounded velocities range=> need to preserve a double wedge in the frequency domain

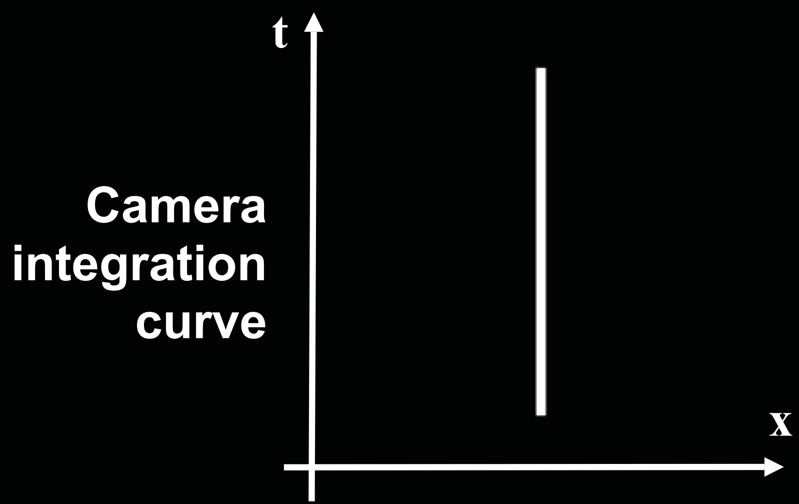
Static camera

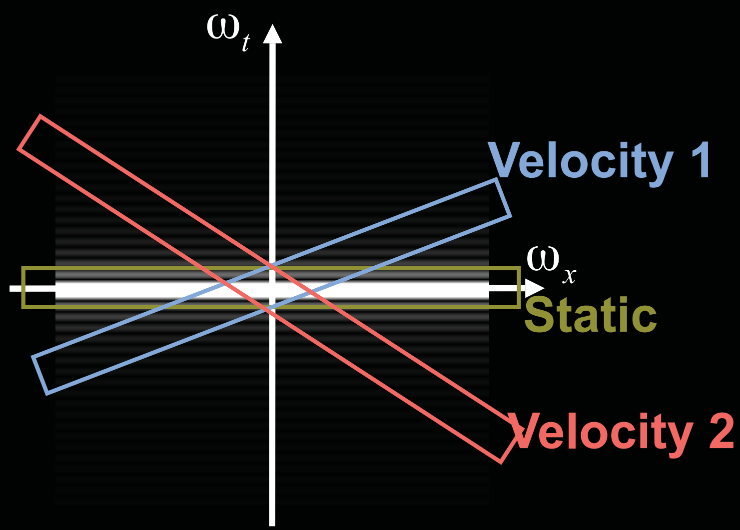
Primal Domain

Frequency Domain





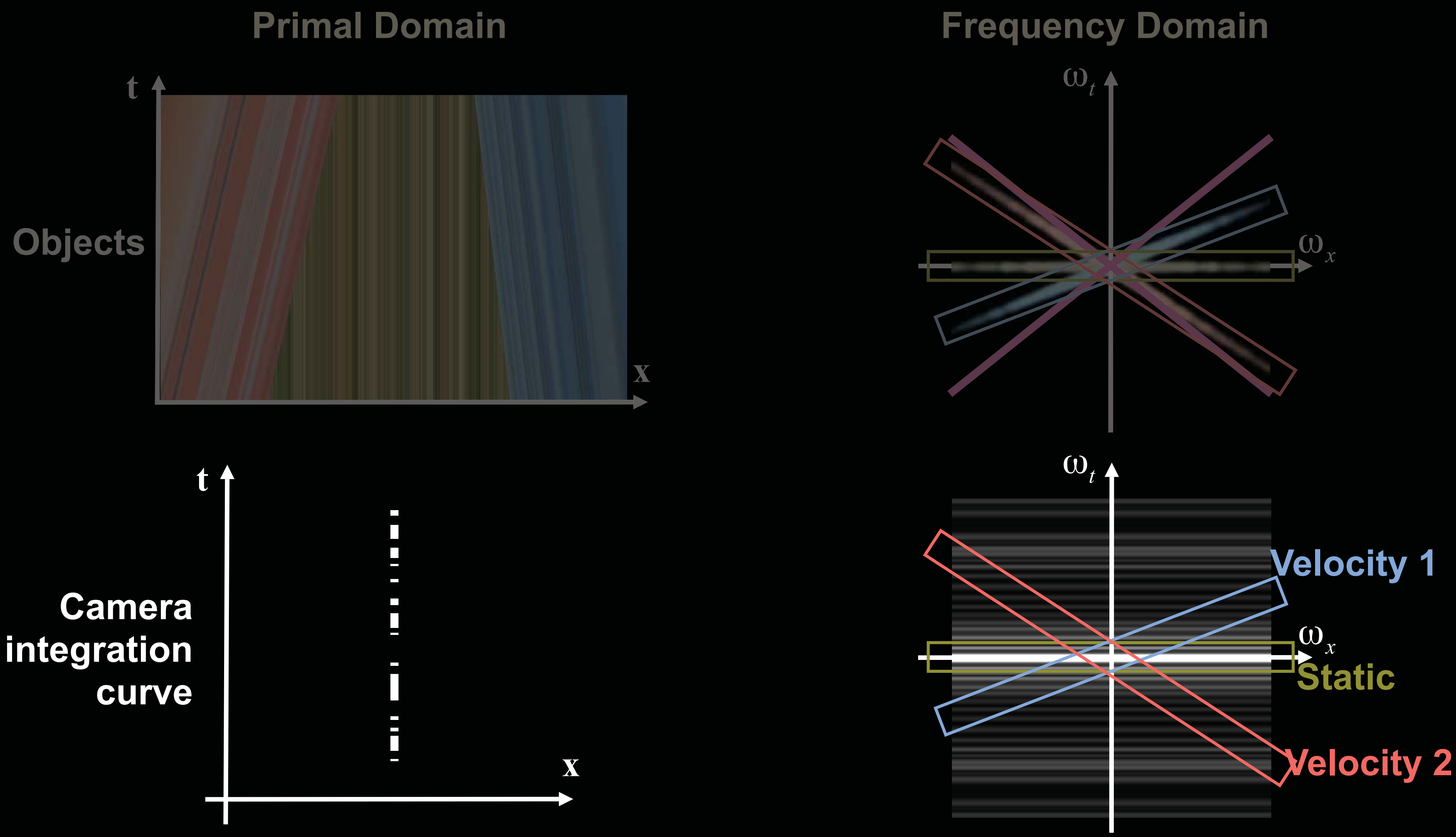




Vertical integration segment **Static object: high response**

Higher velocities: low

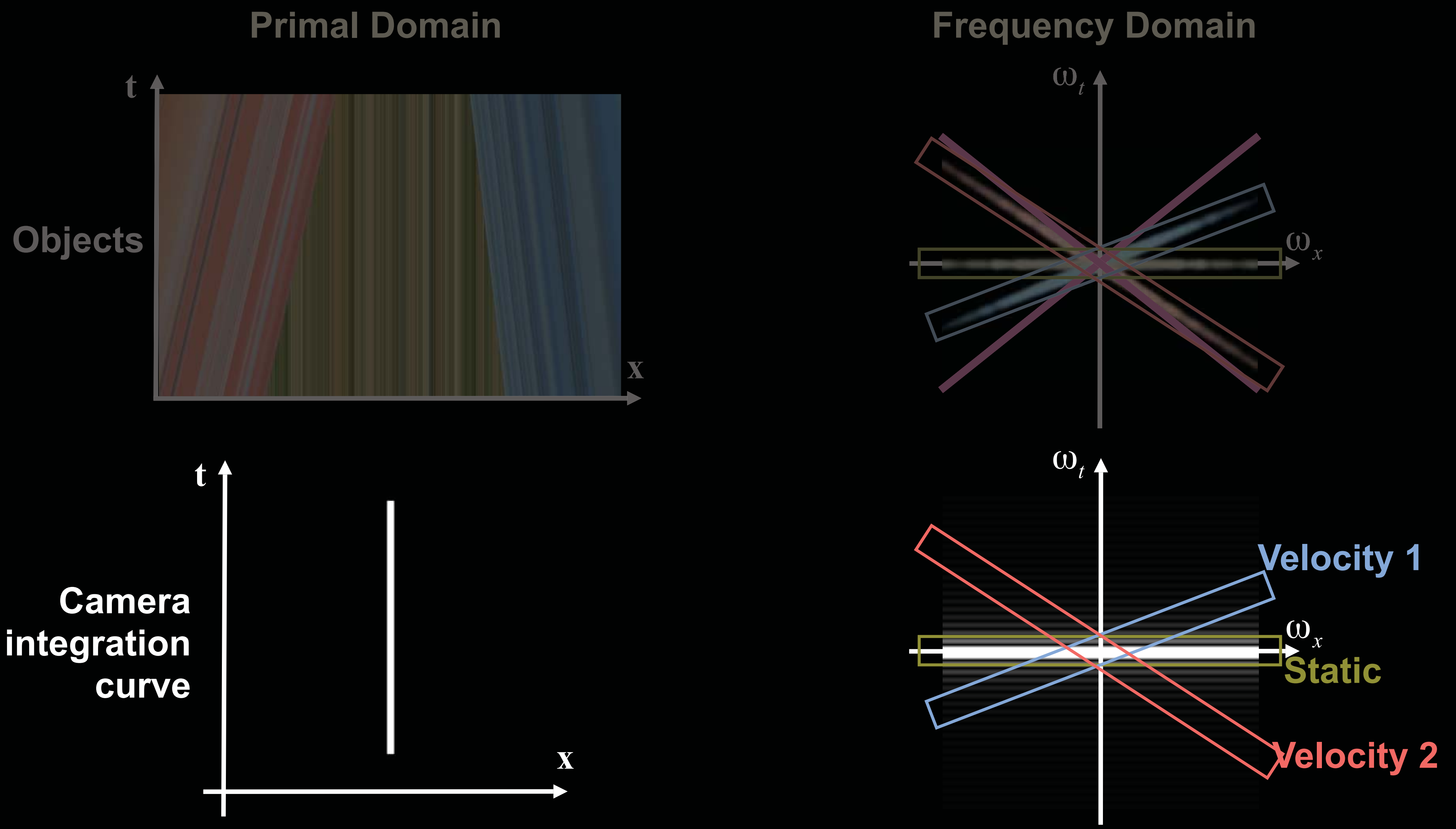
Flutter shutter (Raskar et al 2006)



Vertical but discontinuous integration segment

Higher velocities: better than static camera

Static camera



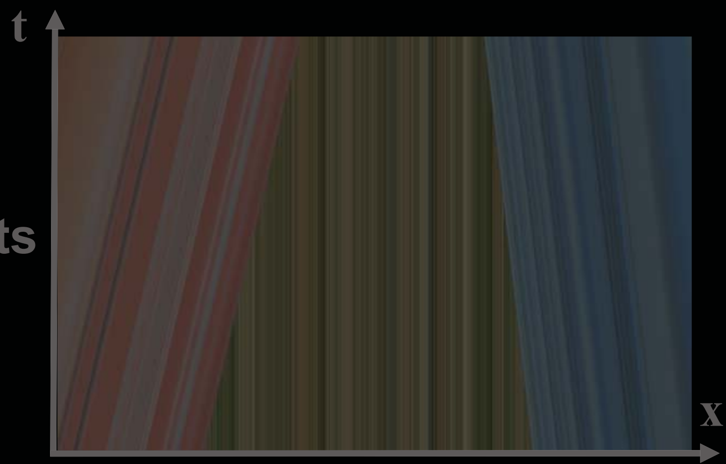
Vertical integration segment **Static object: high response**
Higher velocities: low

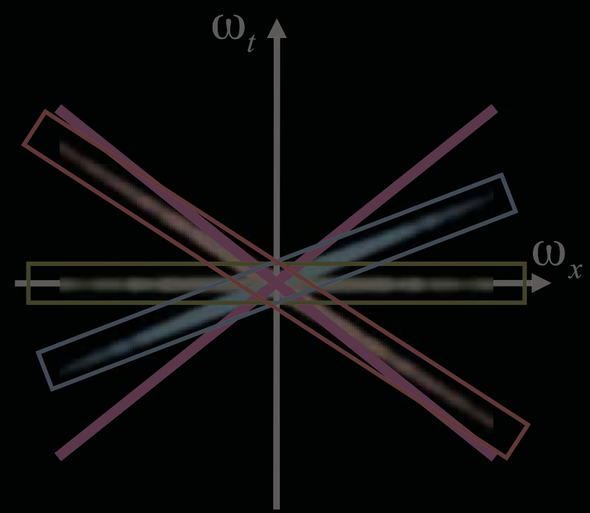
Flutter shutter (Raskar et al 2006)

Primal Domain

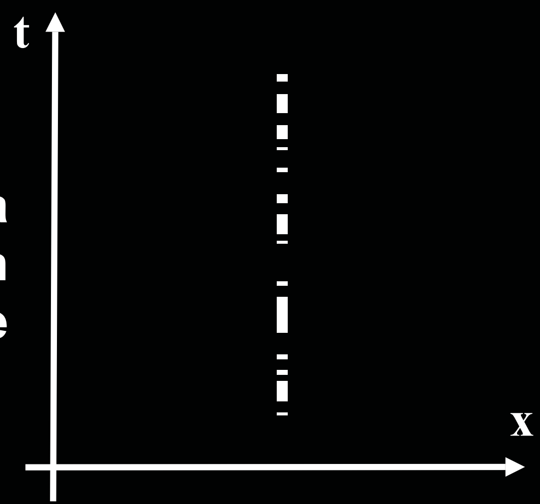
Frequency Domain

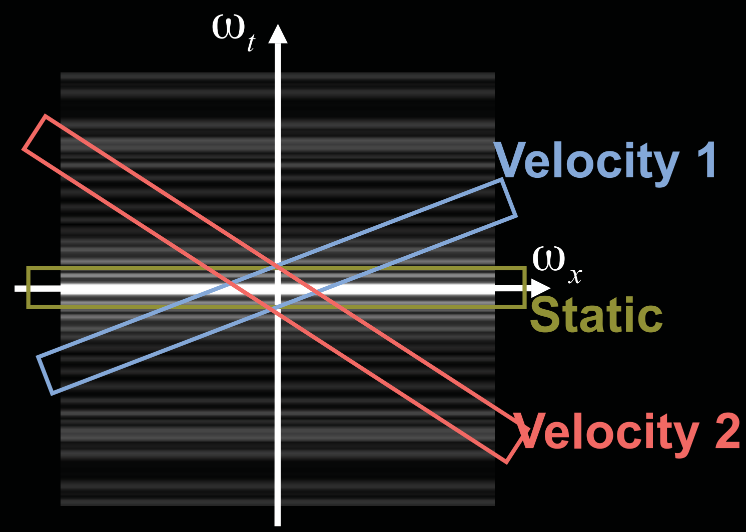
Objects





Camera integration curve

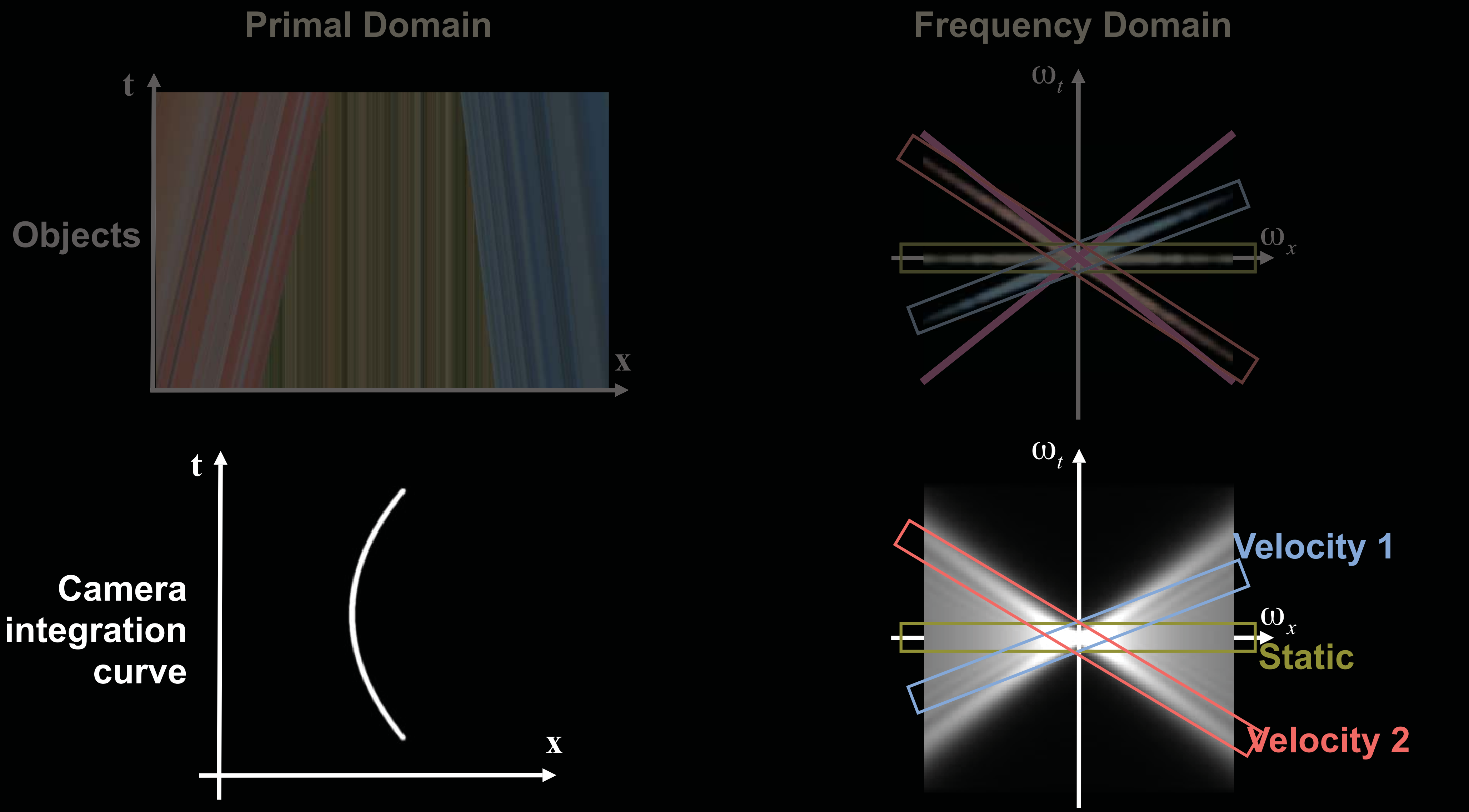




Vertical but discontinuous integration segment

Higher velocities: better than static camera

Our parabolic camera



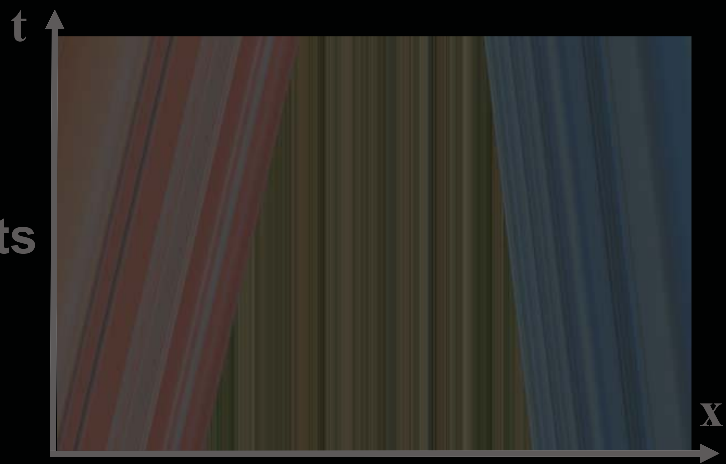
Parabola

Equal high response in all range

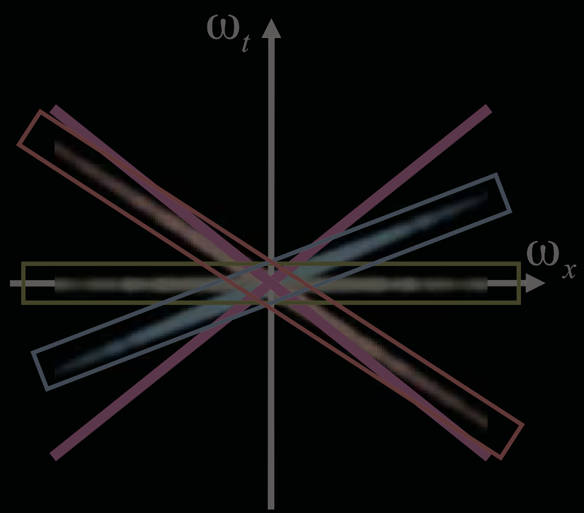
Flutter shutter (Raskar et al 2006)

Primal Domain

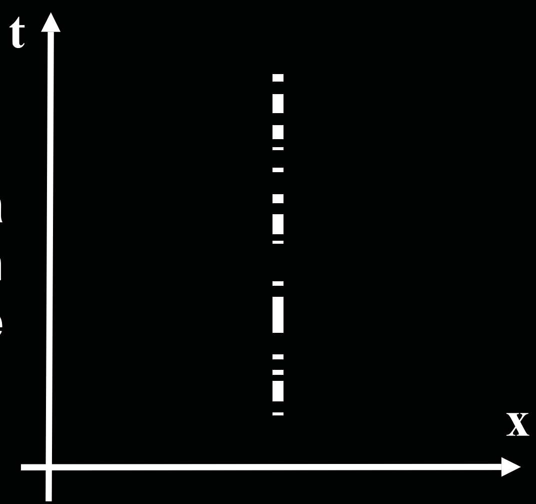
Frequency Domain

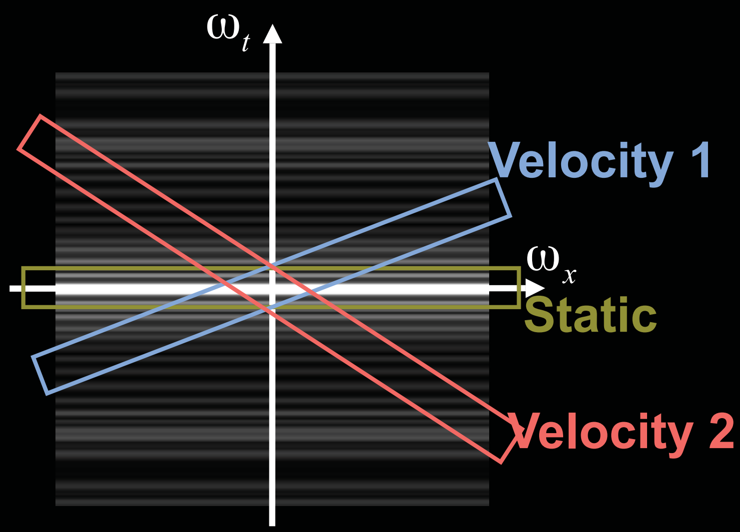


Objects



Camera integration curve

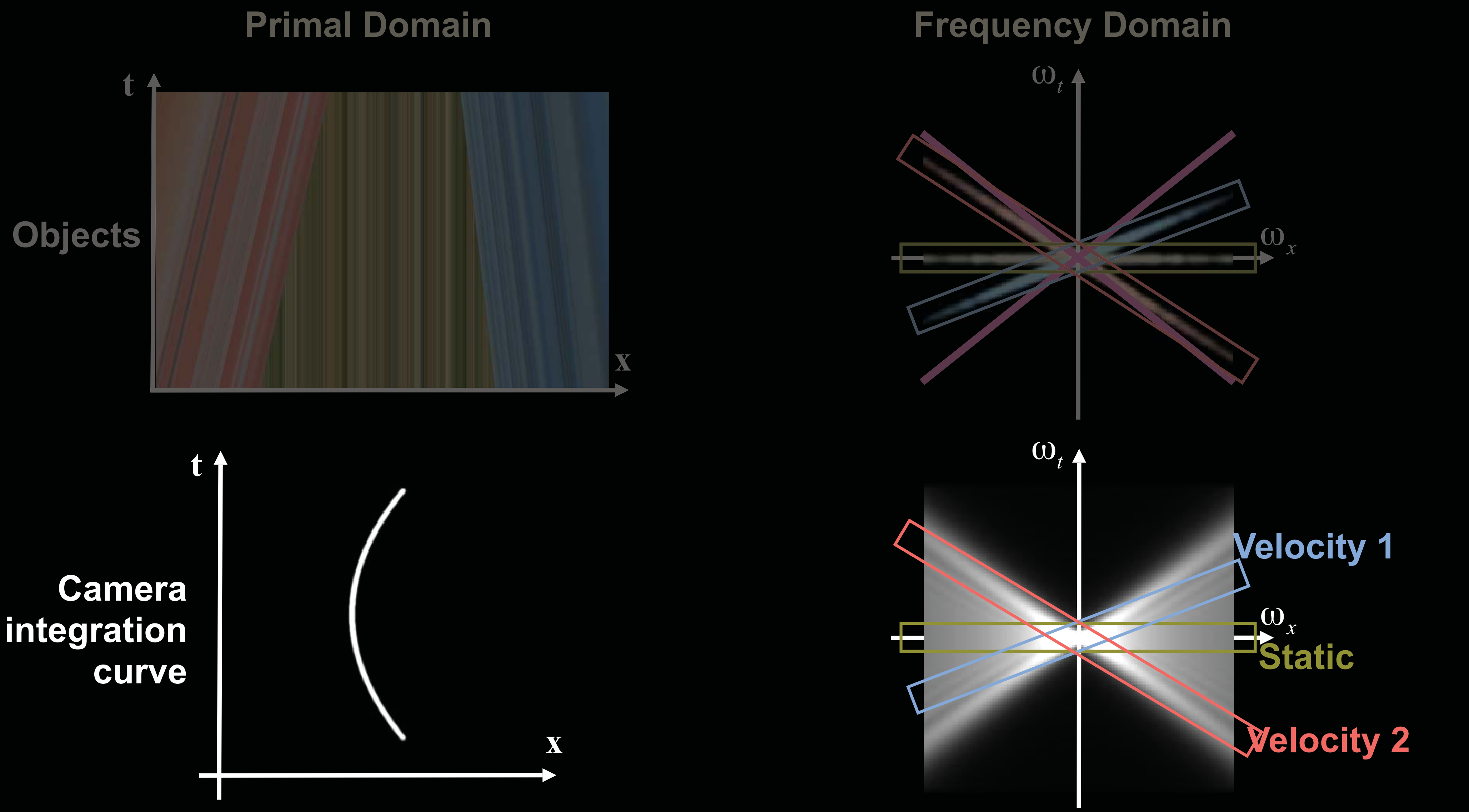




Vertical but discontinuous integration segment

Higher velocities: better than static camera

Our parabolic camera



Parabola

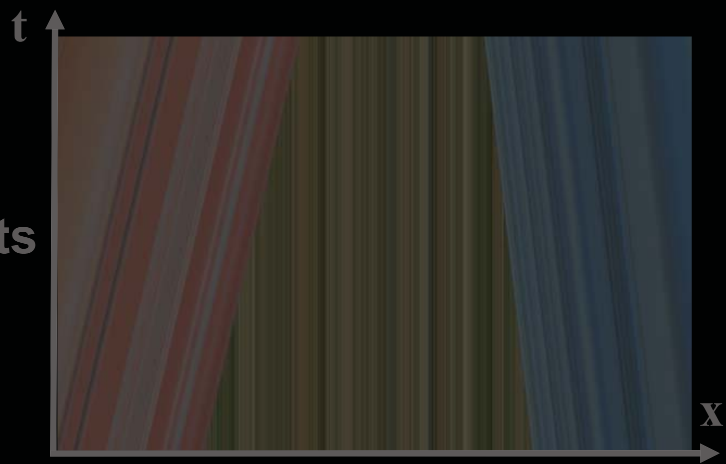
Equal high response in all range

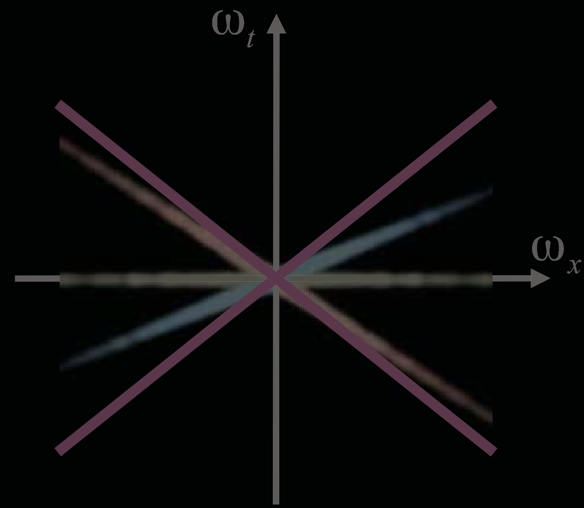
Information budget

Primal Domain

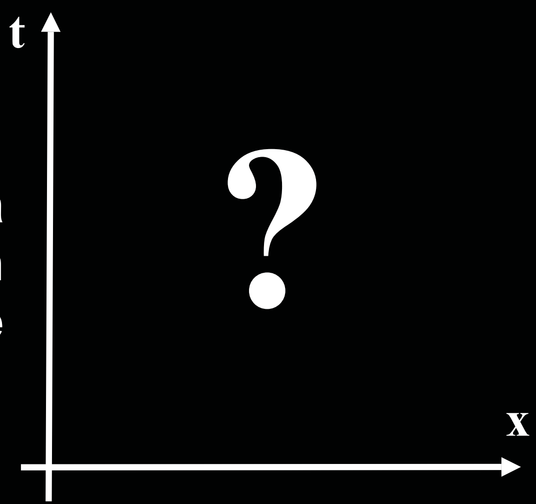
Frequency Domain

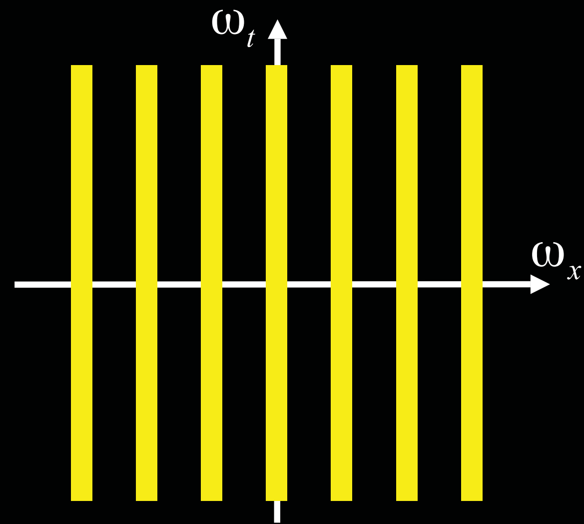
Objects





Camera integration curve



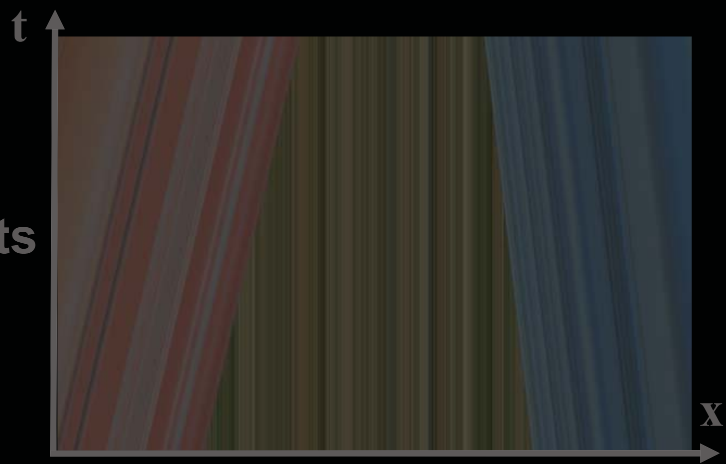


Bounded number of photons ⇨ **Bounded budget per column (norm of power spectrum)**

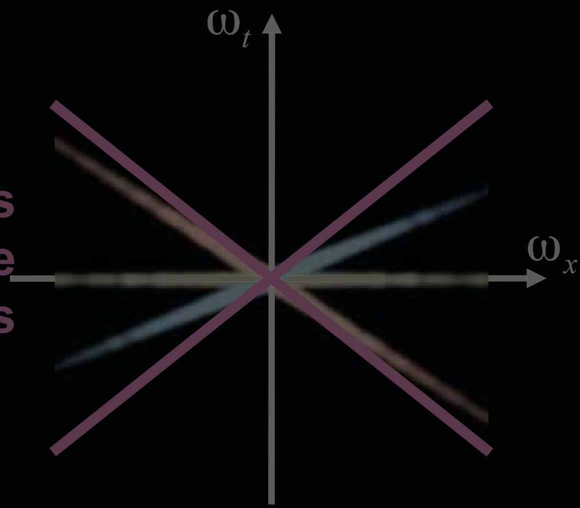
Upper bound given velocity range

Primal Domain | **Frequency Domain**

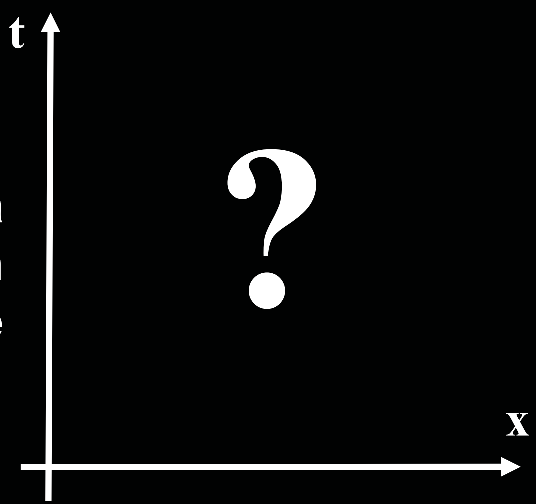
Objects

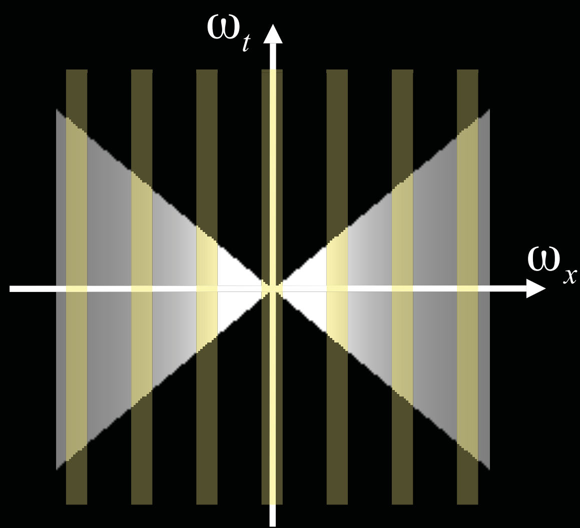


Frequencies from possible motions



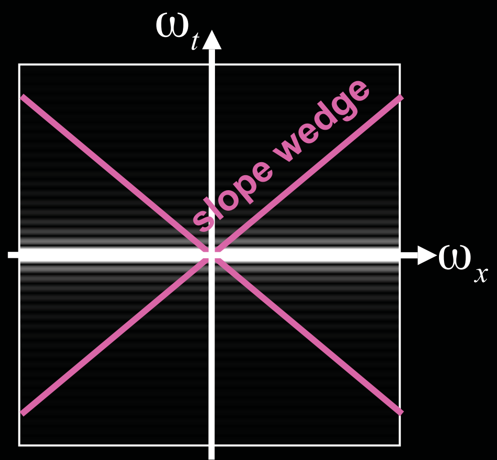
Camera integration curve



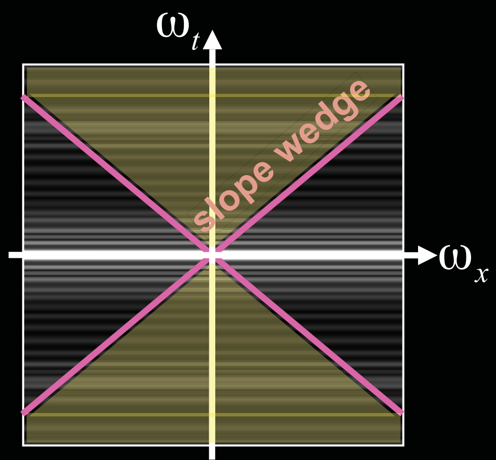


For each column, distribute budget uniformly within wedge $|K(\omega_x,\omega_t)|^2 \le \frac{1}{|\omega_x|}$

Cameras and information preservation



Static

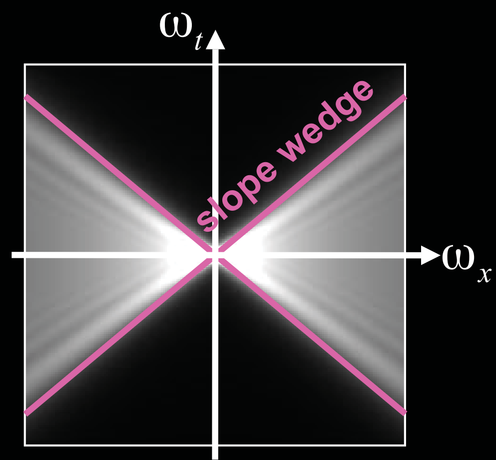


Flutter shutter

Constant horizontally

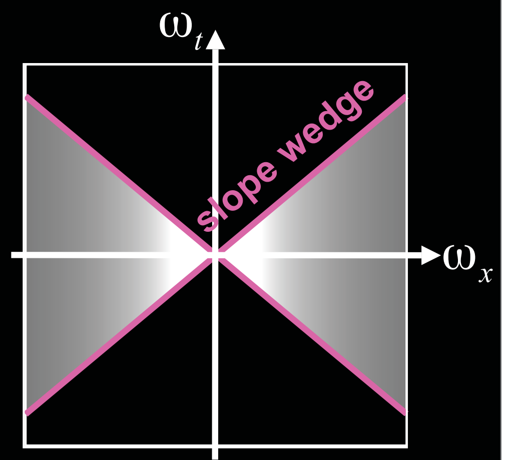
Spends frequency "budget" outside wedge

Handles 2D motion



Parabolic

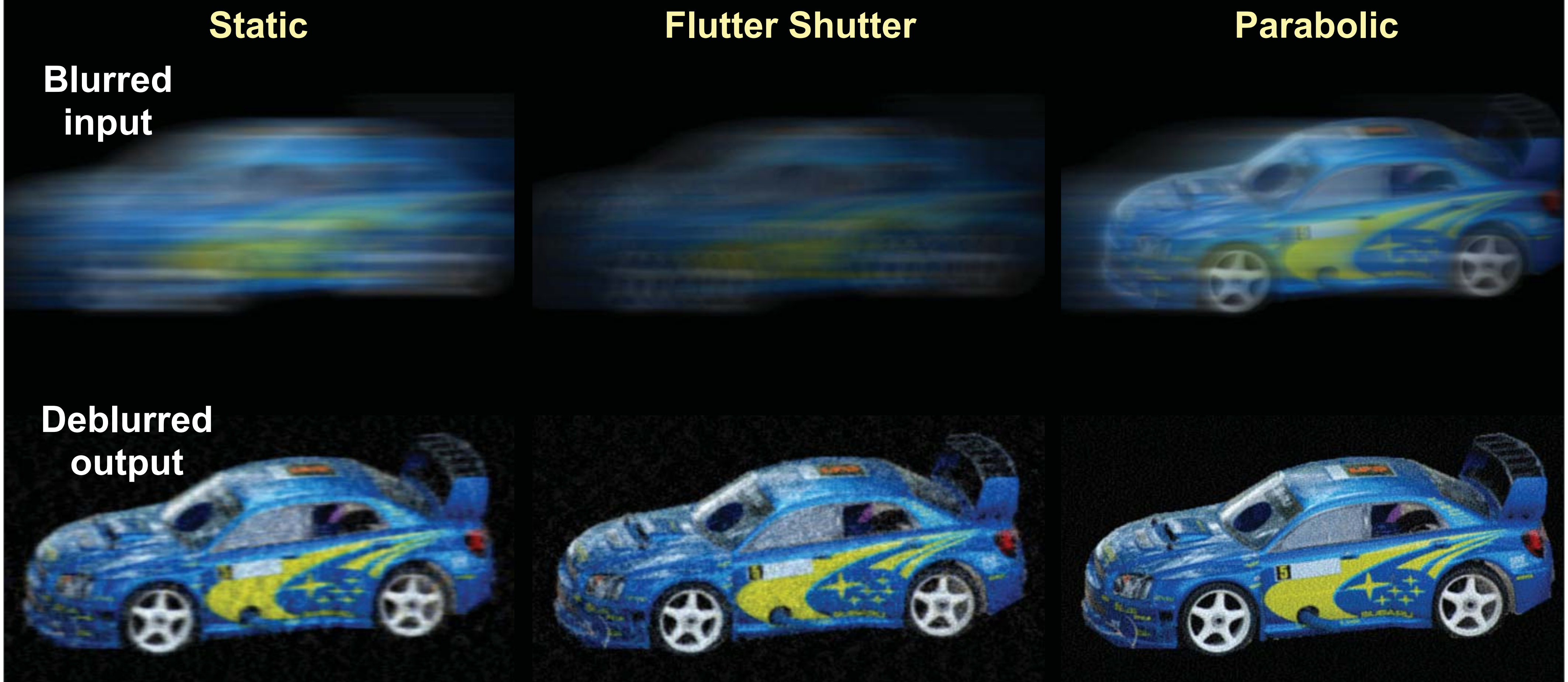
Near optimal "budget" usage at all frequencies



Upper bound

Bounded "budget" per column ω_x

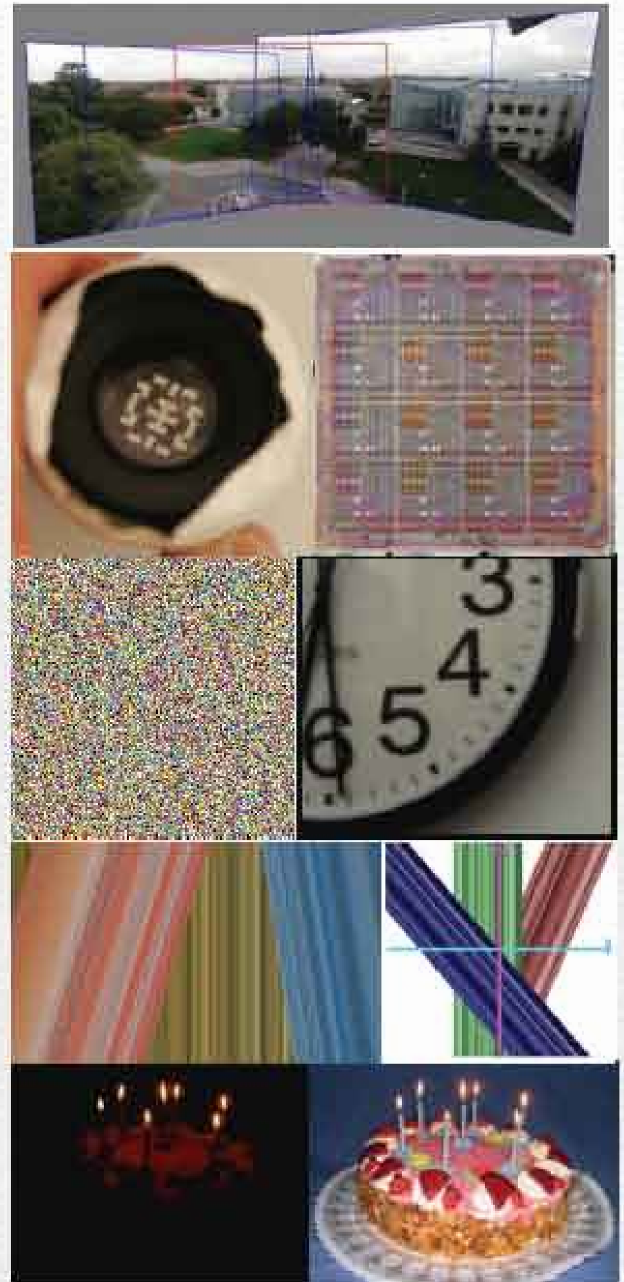
Comparing camera reconstruction



Note: synthetic rendering, exact PSF is known

Big ideas in Comp Photo

- Multiple-exposure & multiplexing
- **Coded imaging**
- Prior information
- **The raw data is high dimensional**
- Active imaging

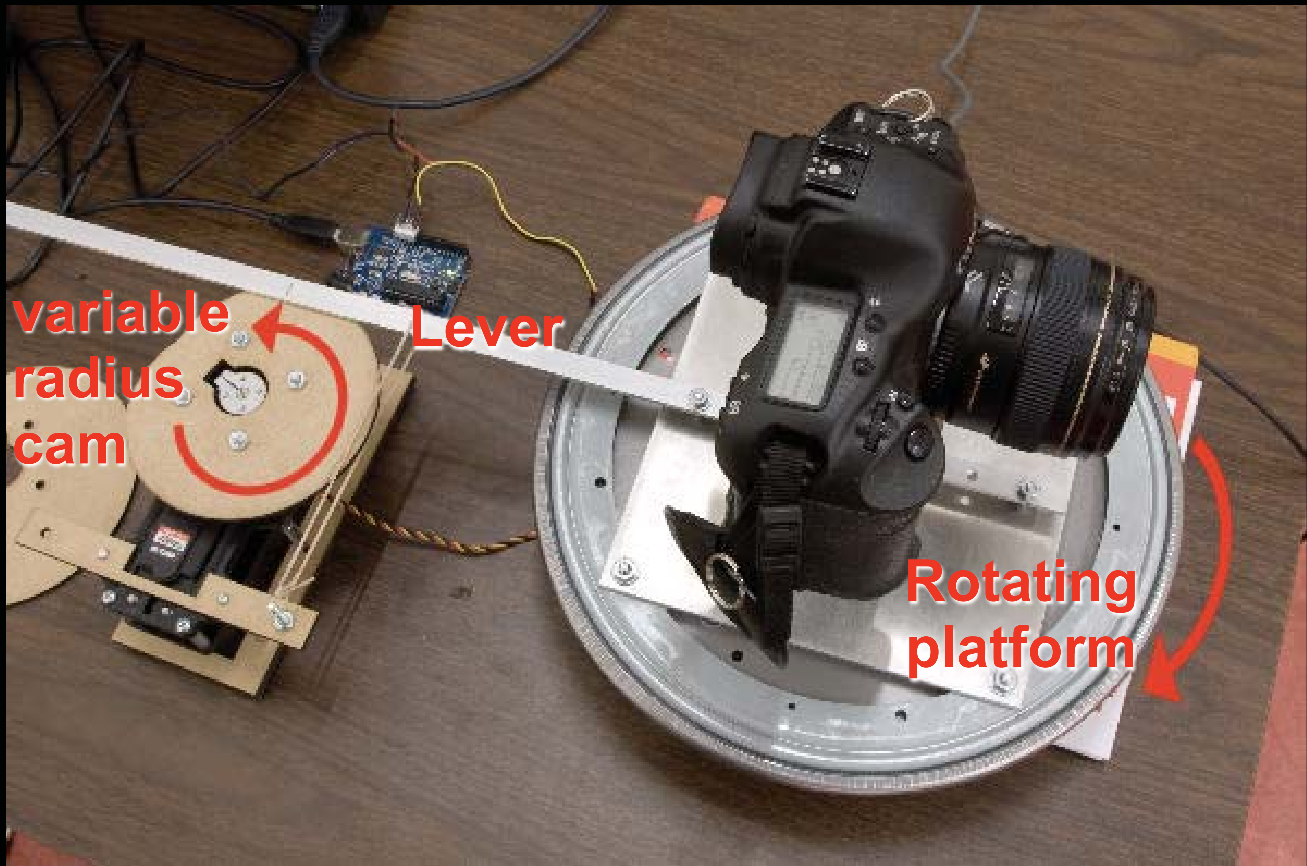


Hardware construction

- Ideally move sensor

  (requires same hardware as existing stabilization systems)

- In prototype implementation: rotate camera



Linear rail

Static camera input-
Unknown and variable blur

Our parabolic input- Blur
is invariant to velocity

Linear rail

Static camera input-
Unknown and variable blur

Our output after deblurring-
NON-BLIND deconvolution

Human motion– no perfect linearity

Input from a static camera

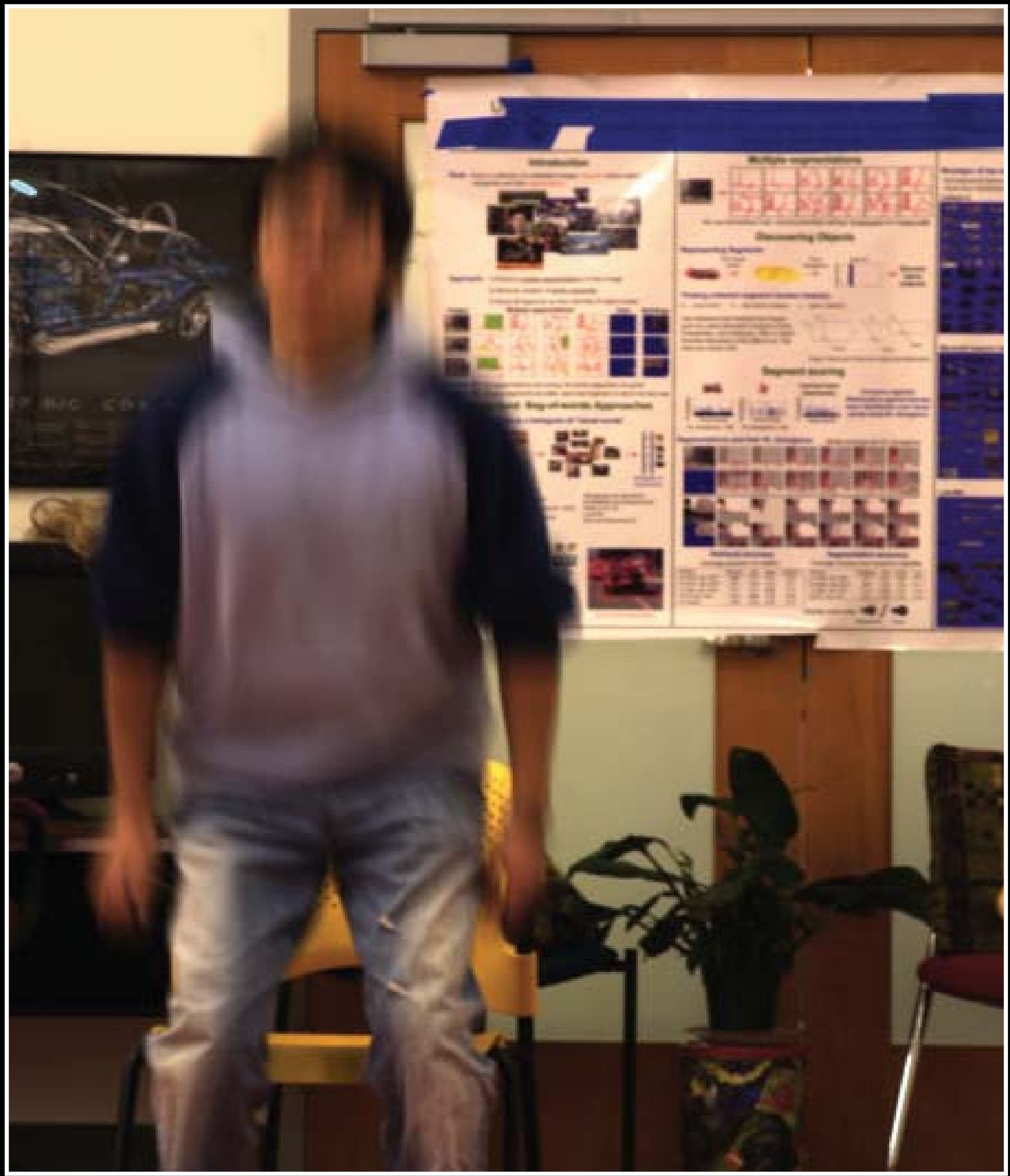
Deblurred output from our camera

Violating 1D motion assumption– forward motion

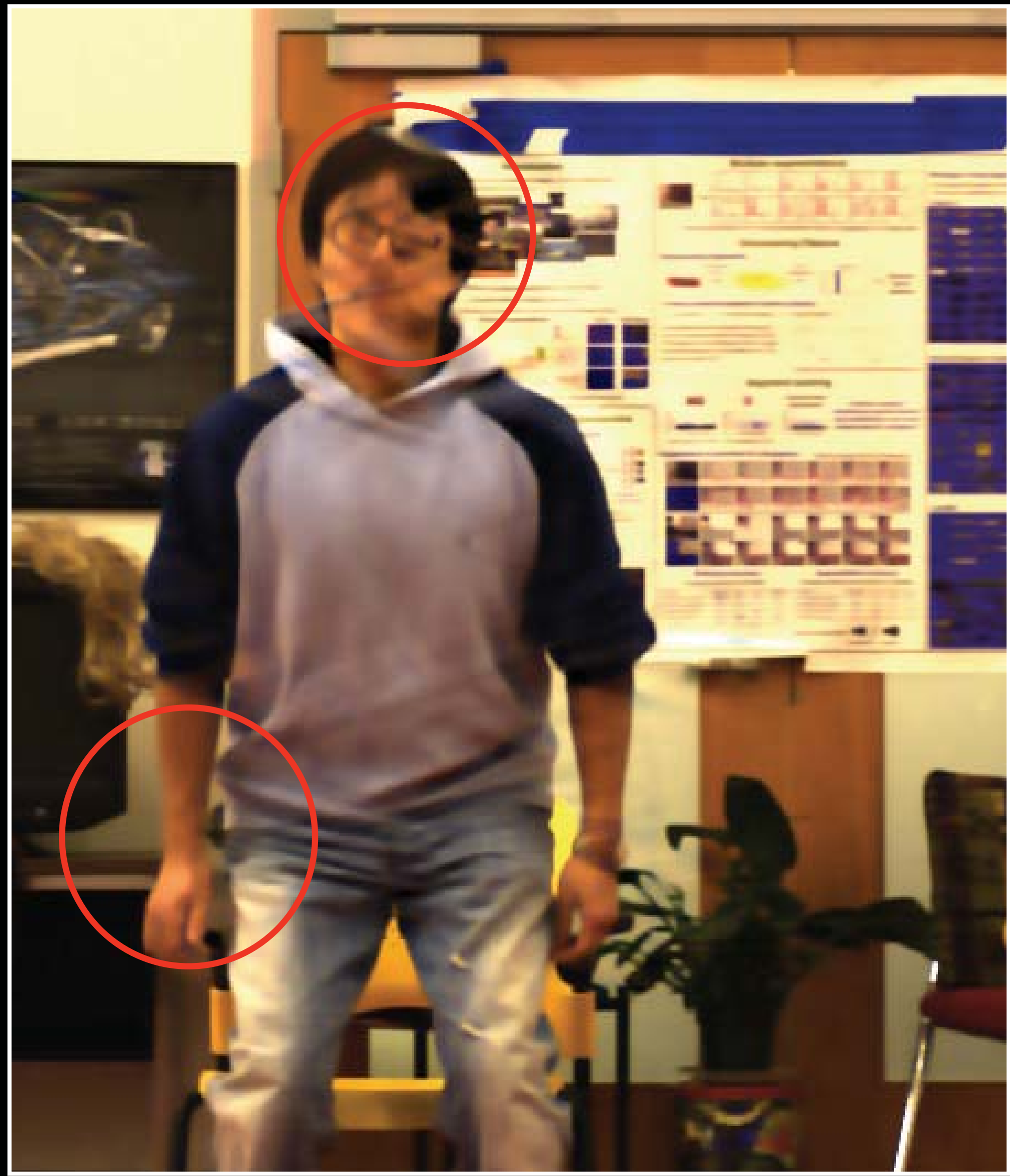
Input from a static camera

Deblurred output from our camera

Violating 1D motion assumption- stand-up motion



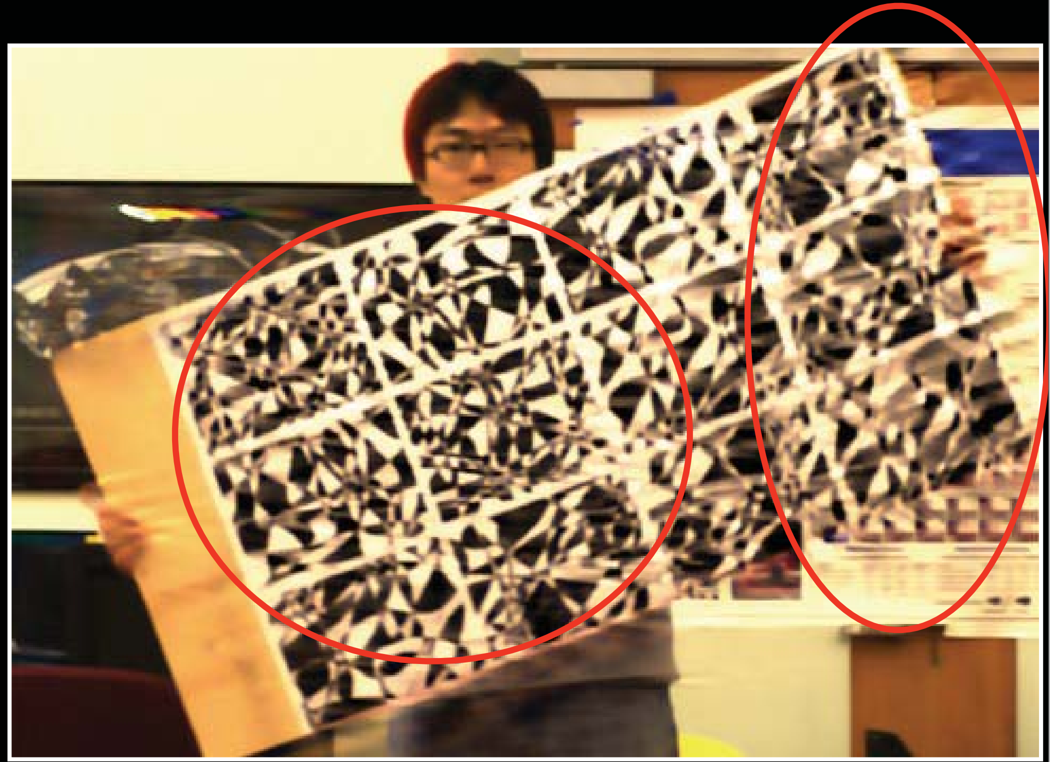
Input from a static camera



Deblurred output from our camera

Violating 1D motion assumption- rotation

Input from a static camera



Deblurred output from our camera

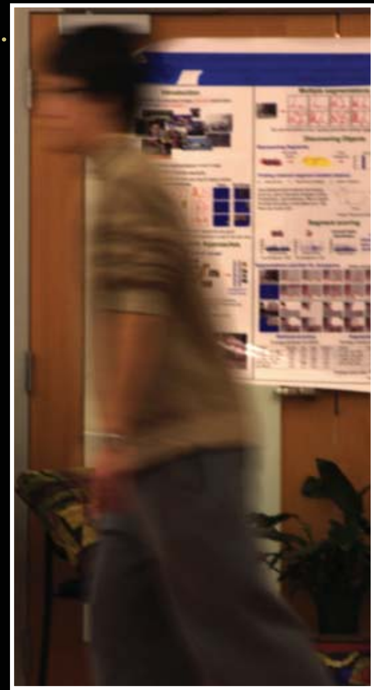
Parabolic curve – issues

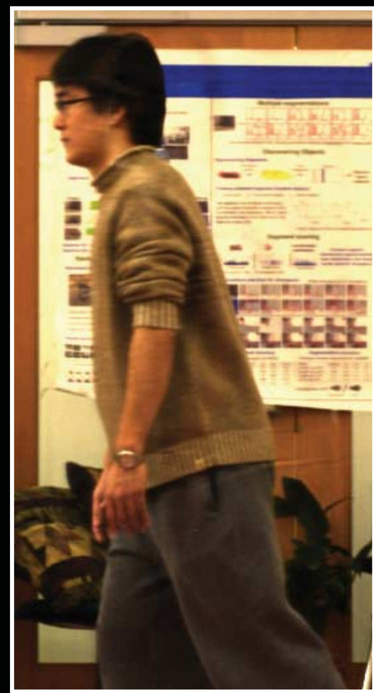
- Spatial shift- but does not affect visual quality in deconvolution
- Parabola tail clipping: not exactly the same blur
- Motion boundaries break the convolution model
- Assumes: Object motion horizontal

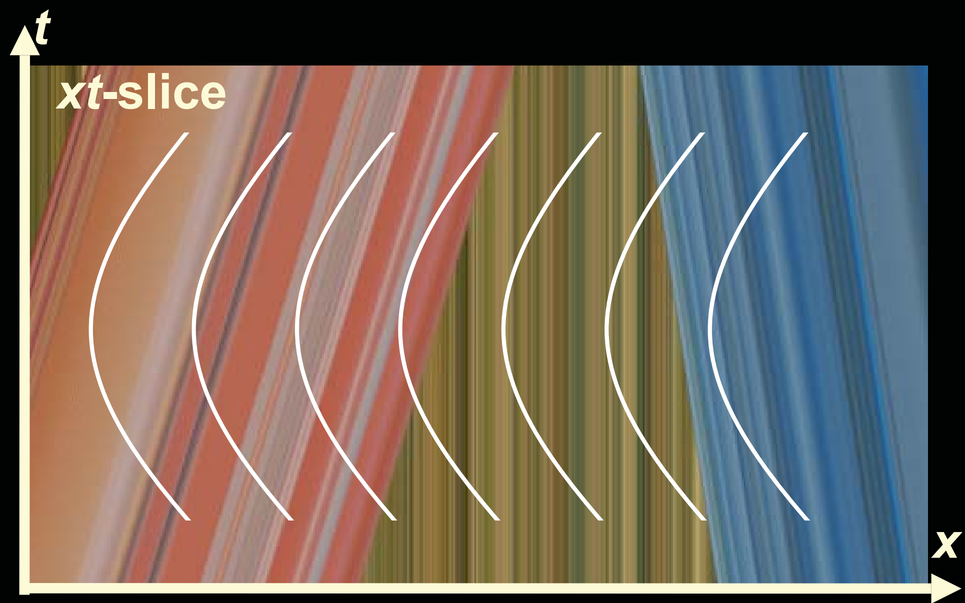
  Object motion linear up to 1st order approximation

Conclusions

- **Camera moved during exposure, parabolic displacement**
- **Blur invariant to motion:**
  - **Same over all image (no need to segment)**
  - **Known in advance (no kernel identification)**
- **Easy to invert (near optimal frequency response)**
- **For 1D motion**
  - **Somewhat robust to 1D motion violation**
  - **Future work: 2D extensions**





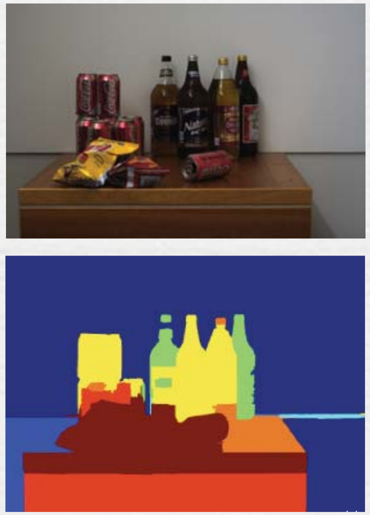


Acknowledgments:
NSF CAREER award 0447561
Royal Dutch/Shell Group NGA
NEGI-1582-04-0004
Office of Naval Research MURI
MSR New Faculty Fellowship
Sloan Fellowship

Summary

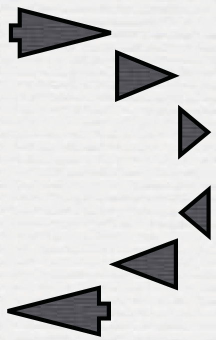
- Natural image prior

- Coded aperture : make kernel easier to identify
  - benefit:
    large depth of field + depth



- Motion-invariant photography
  - For 1D motion, blur invariant to velocity

(c) Static camera (d) Parabolic camera



Big ideas in Comp Photo

- Goals:
  - Beat physics, better image quality/quantity
  - More data (depth, etc.)
  - New visual media
  - Seeing the unseen
- Multiple-exposure imaging & multiplexing
- Coded imaging
- Prior information
- Edges matter
- The raw data is high dimensional
- Active imaging

Challenges

- Theory, frameworks, comparisons, optimality
- Diffraction, wave optics
- Putting it all together
  (engineering, system, applications)
- Better priors
  - Kernel identification
  - High-quality inversion
- Video
- Real-time enhancement (microscope)
- Applied visual perception
- Intrinsic images
- Matting
- Scene and object recognition
- 3D reconstruction

Understanding camera trade-offs through a Bayesian analysis of light field projections

Anat Levin[1], Bill Freeman[1,2], Fredo Durand[1]

Computer Science and Artificial Intelligence Lab (CSAIL),

[1]Massachusetts Institute of Technology

and [2]Adobe Systems

Cameras, old and new

Traditional camera: Lens forms final 2D image

Computational camera: Recorded data is not the final output.

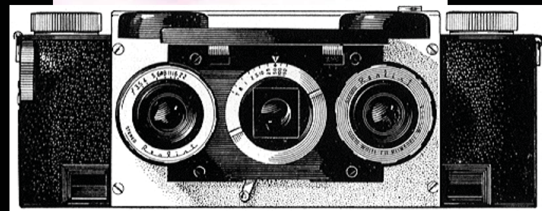
- Visual array estimated from sensor measurements.
- Extra design degree of freedom.
  Beyond 2D images--acquisition of light field or depth.
  Post-exposure re-synthesis of image.

An explosion of cameras

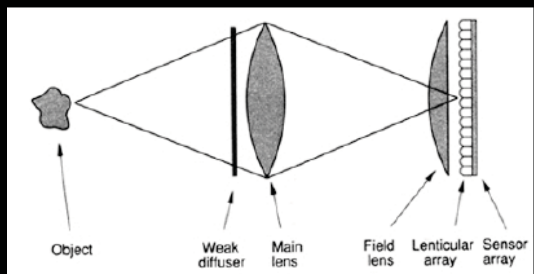
- Best way to capture image and depth: Stereo? Plenoptic camera? Coded aperture? or...?
- What aspects of these cameras contribute to their performance?
- Can we design new cameras with improved reconstruction performance?

Conventional single-lens cameras

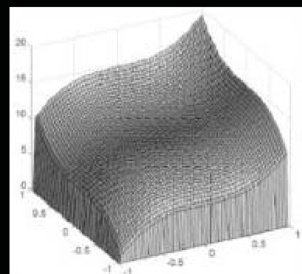
Stereo and trinocular cameras



Plenoptic cameras



Coded aperture

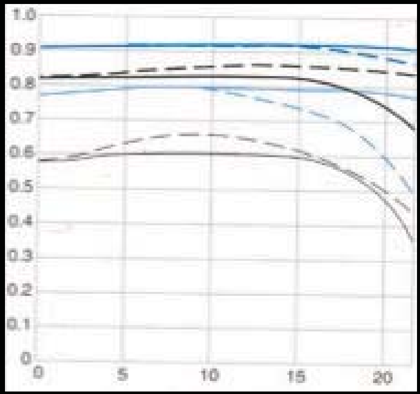


Wavefront coding

Camera evaluation, old and new

Traditional optics evaluation:

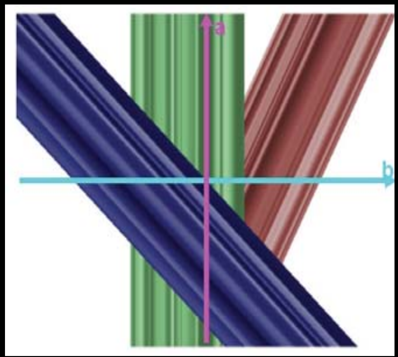
2D image sharpness (eg, Modulation Transfer Function)



contrast vs. spatial frequency

Our modern camera evaluation:

How well does the recorded data allow us to <u>estimate</u> the visual world - the lightfield?



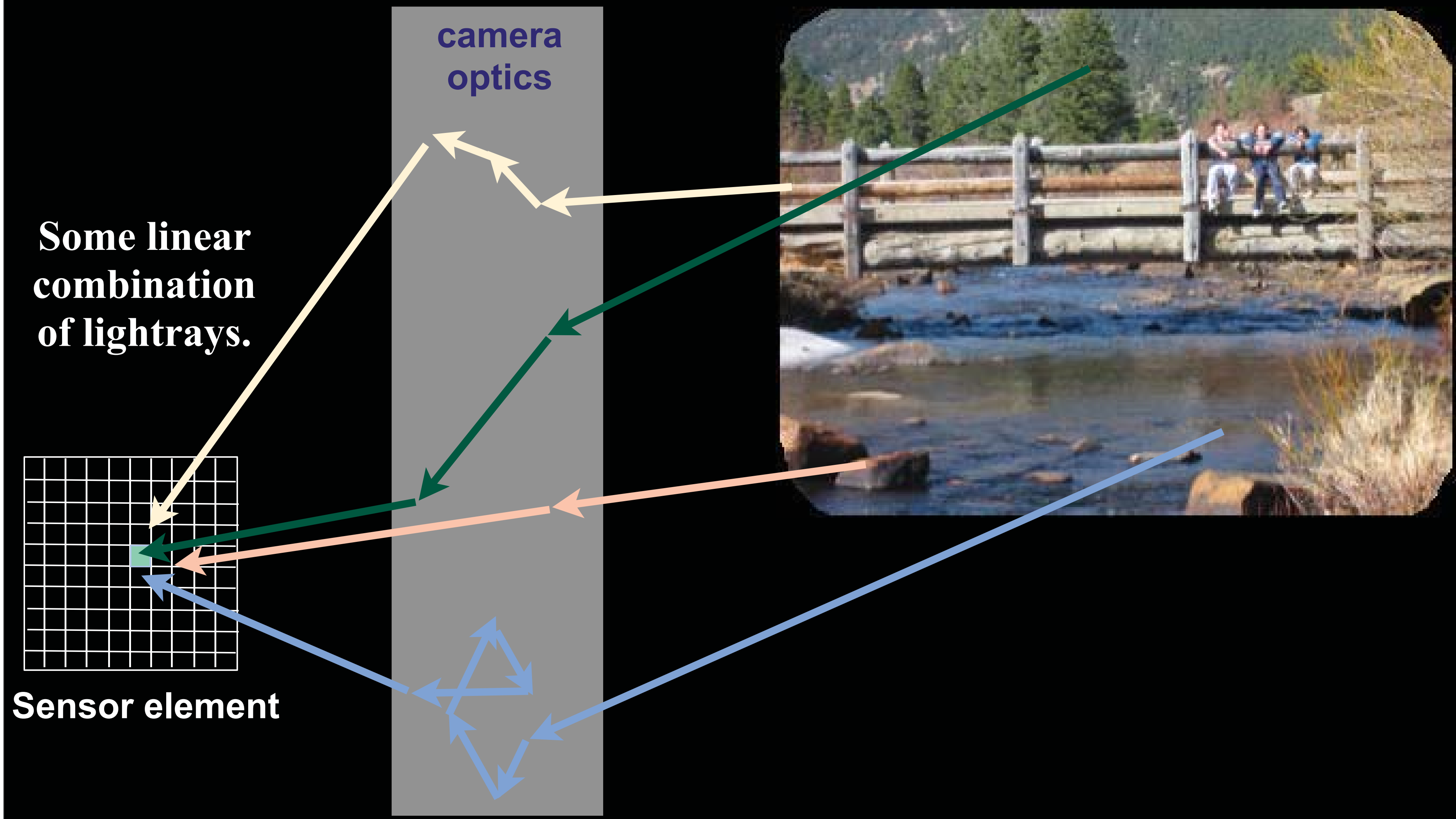
lightfield reconstruction

Computational photography camera evaluation: an estimation problem

- Characteristics of the signal to be estimated.
- Projection functions of various cameras.
- Bayesian lightfield analysis
  - Reconstructing the lightfield from camera data.
  - Comparing performance tradeoffs of different cameras.

so let’s talk about lightfields and cameras

What does a camera sensor element record?



Sensor element data

$$y_i = T_i x + n_i$$

datum

The camera
4D->2D linear projection

The lightfield
(4D)

noise

What is a camera?

Camera: all-positive linear projection of a 4D lightfield

$$\mathbf{y} = \mathbf{T} \quad \mathbf{x} + \mathbf{n}$$

noise

camera data

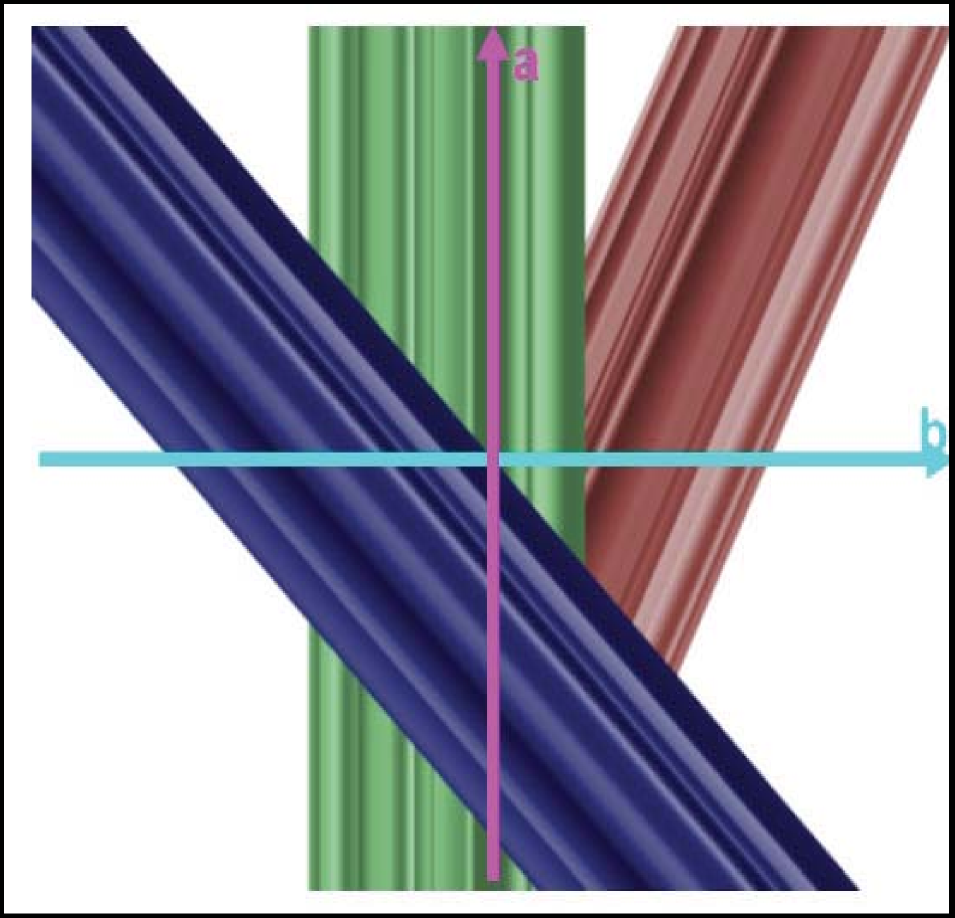
The camera
4D->2D linear projection

The lightfield
(4D)

A more revealing parameterization of the lightfield

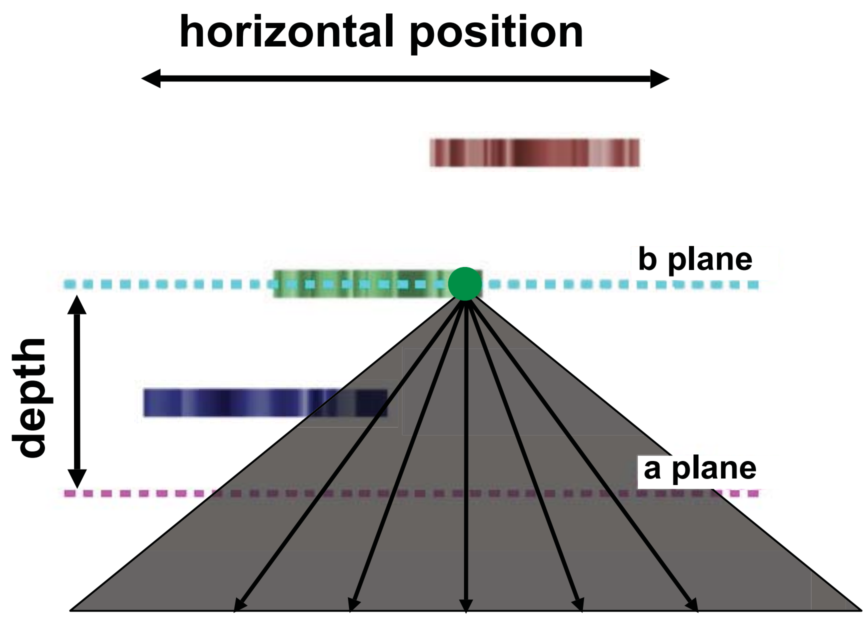
Light field: parameterization of the 4D space of light rays in the world

Provides a convenient way to model different lenses and cameras designs

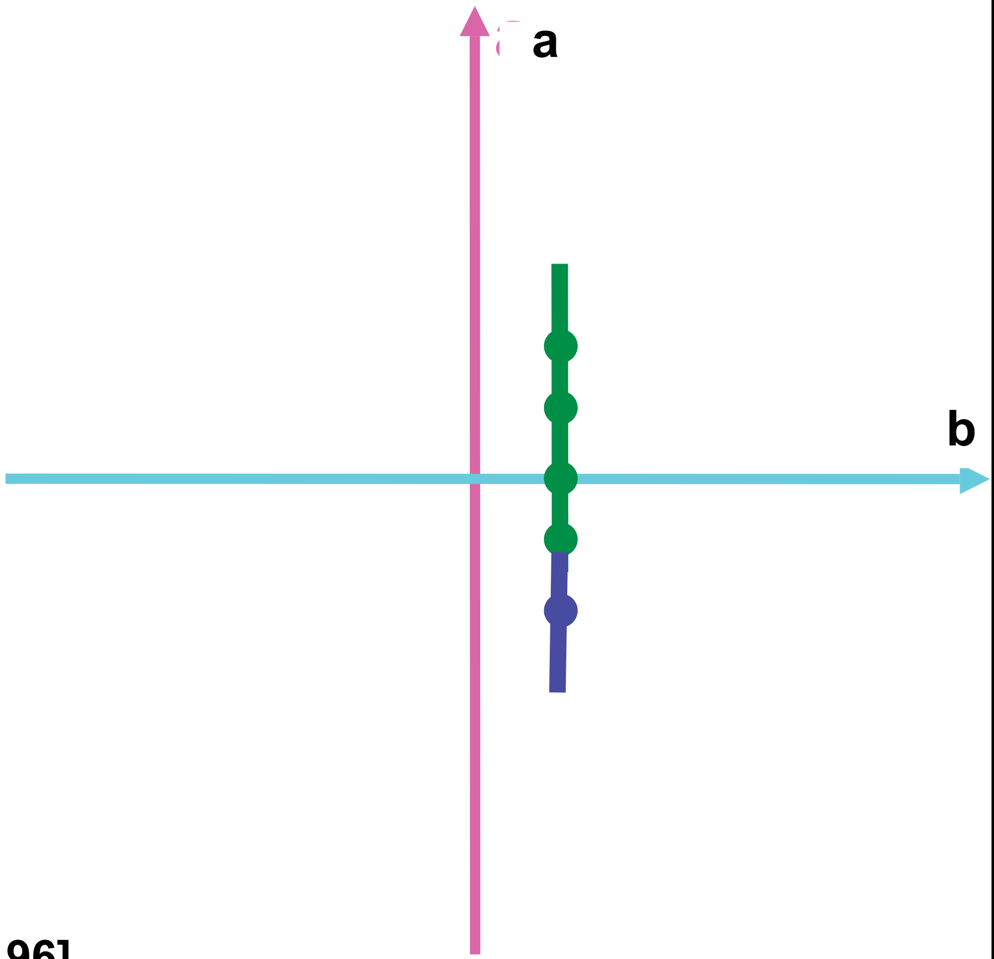


Lightfield tutorial

flatworld 1D scene



2D lightfield

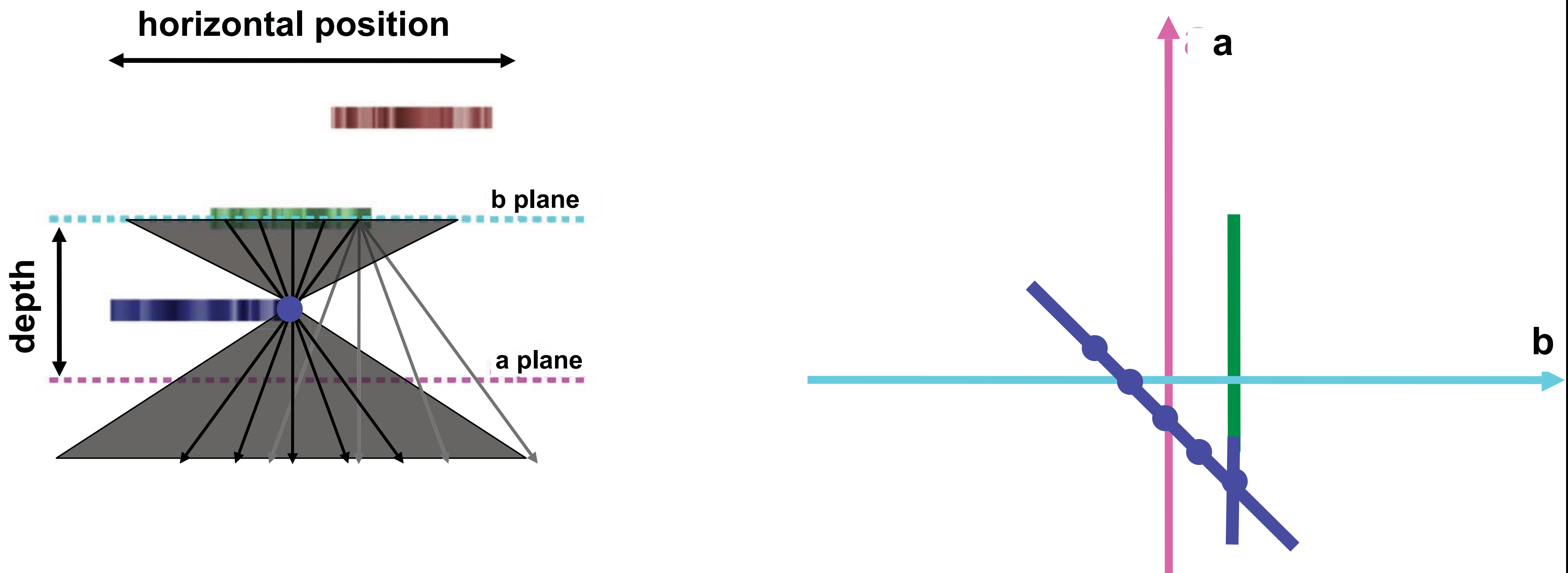


2 plane parameterization [Levoy and Hanrahan 96]

Lightfield tutorial

flatworld 1D scene

2D lightfield

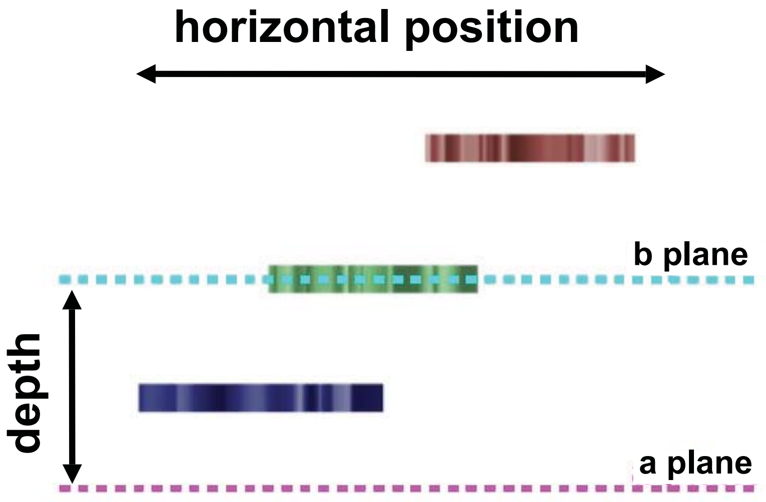


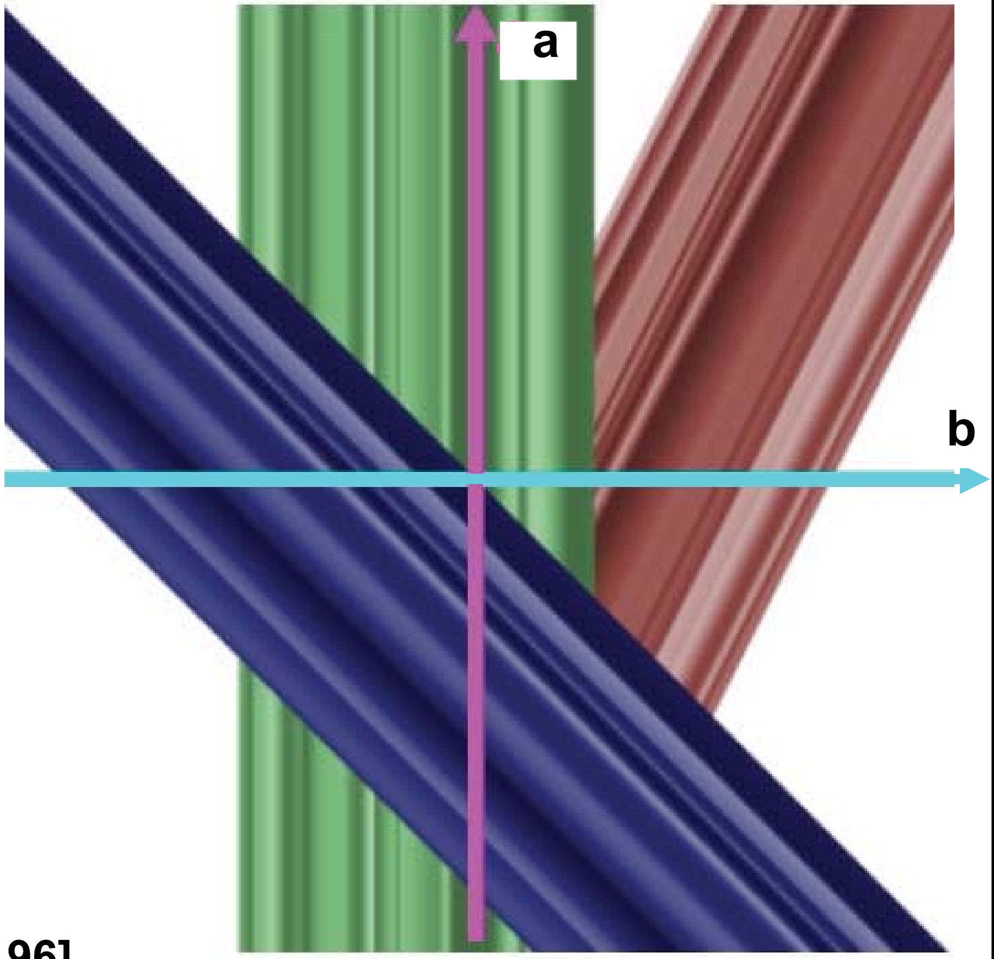
2 plane parameterization [Levoy and Hanrahan 96]

Lightfield tutorial

flatworld 1D scene

2D lightfield



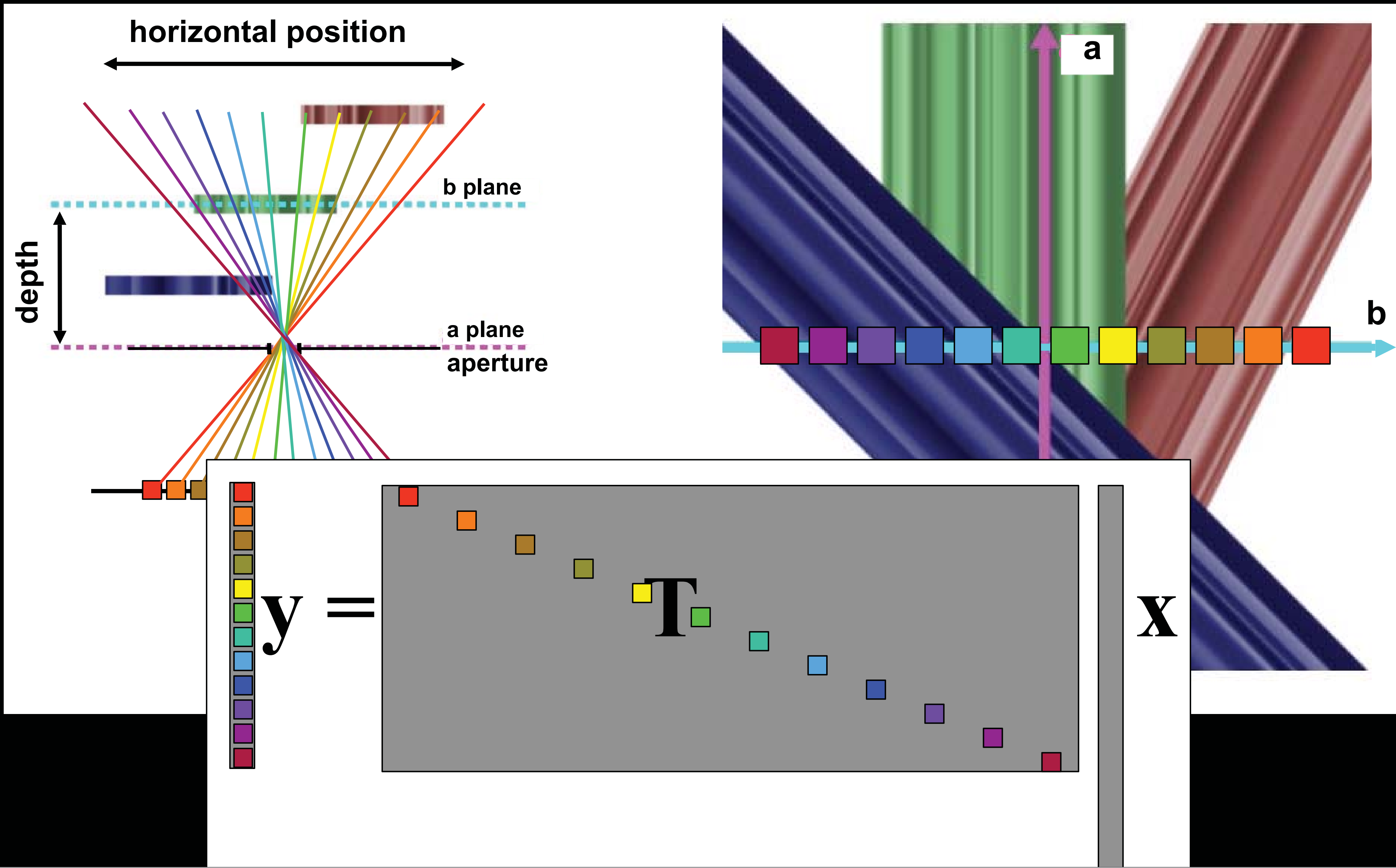


2 plane parameterization [Levoy and Hanrahan 96]

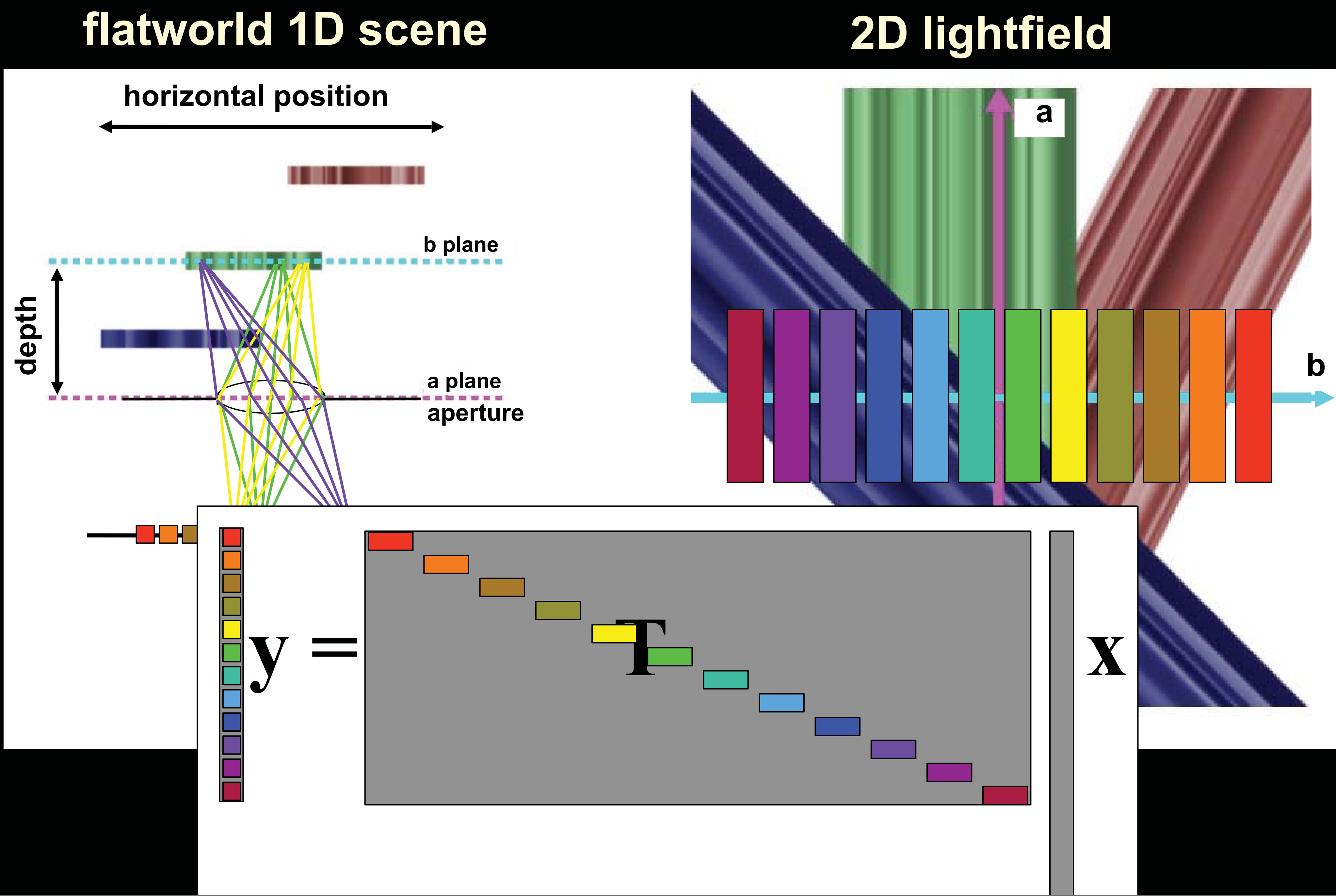
Pinhole camera

flatworld 1D scene

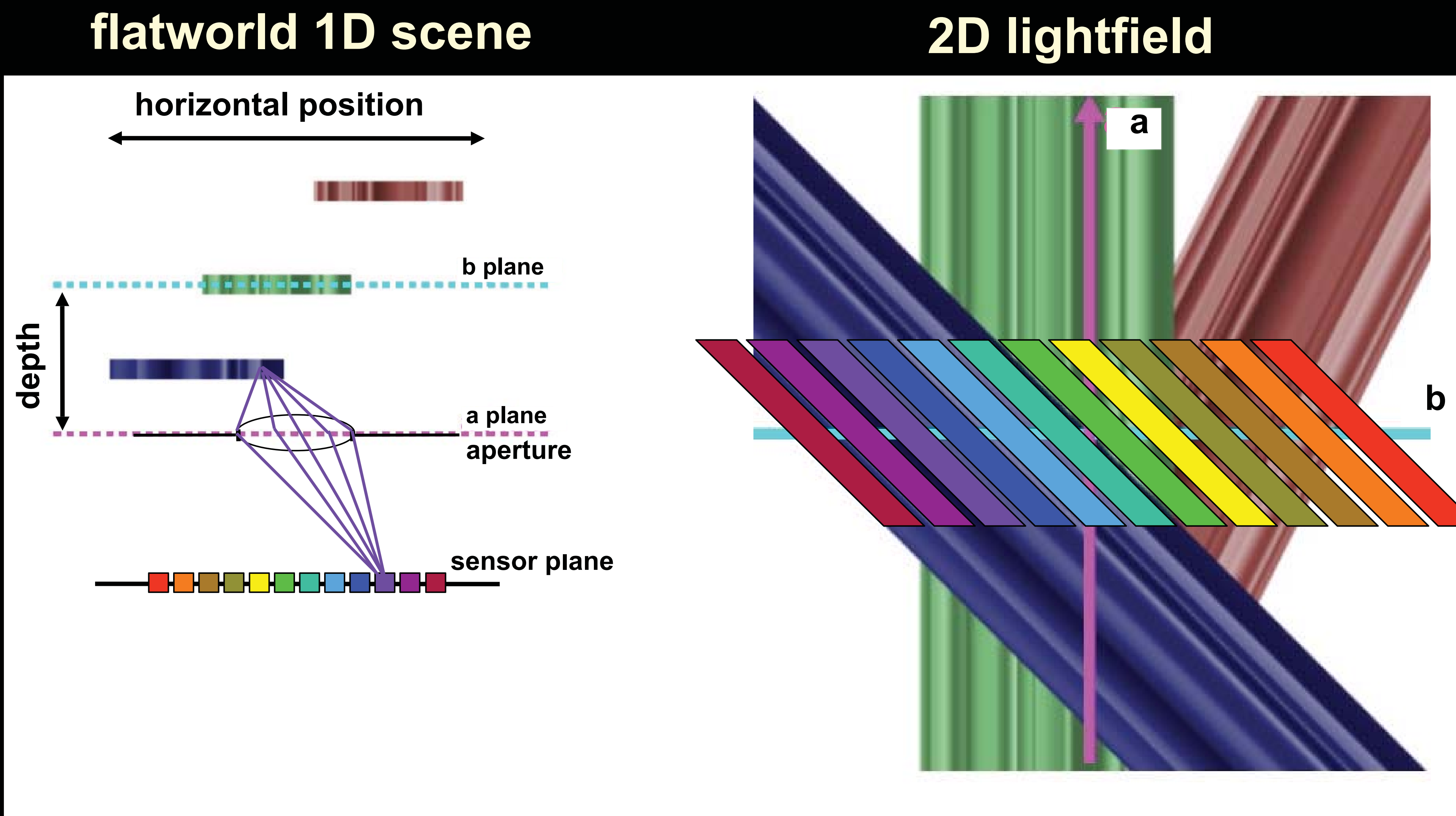
2D lightfield



Lens, focused at green object

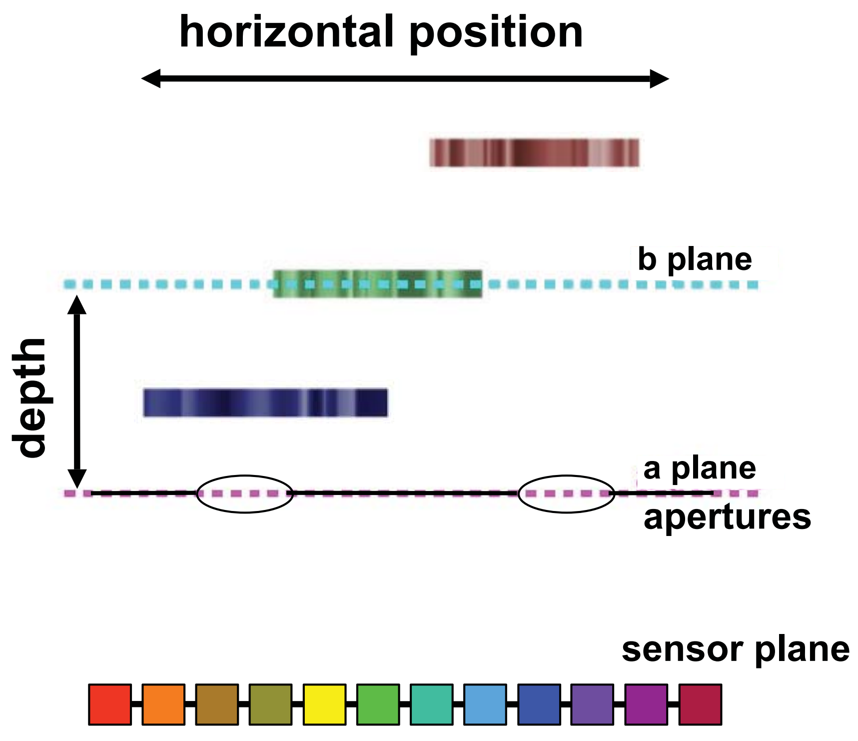


Lens, focused at blue object

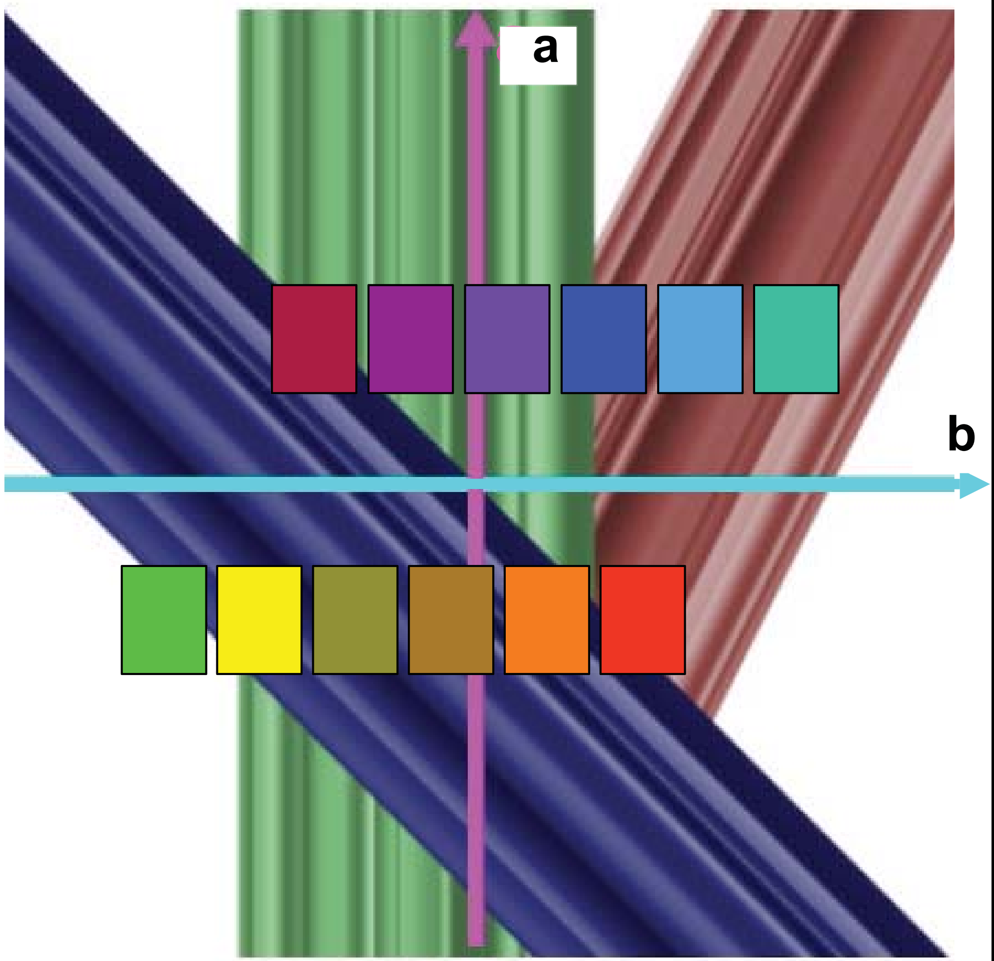


Stereo

flatworld 1D scene

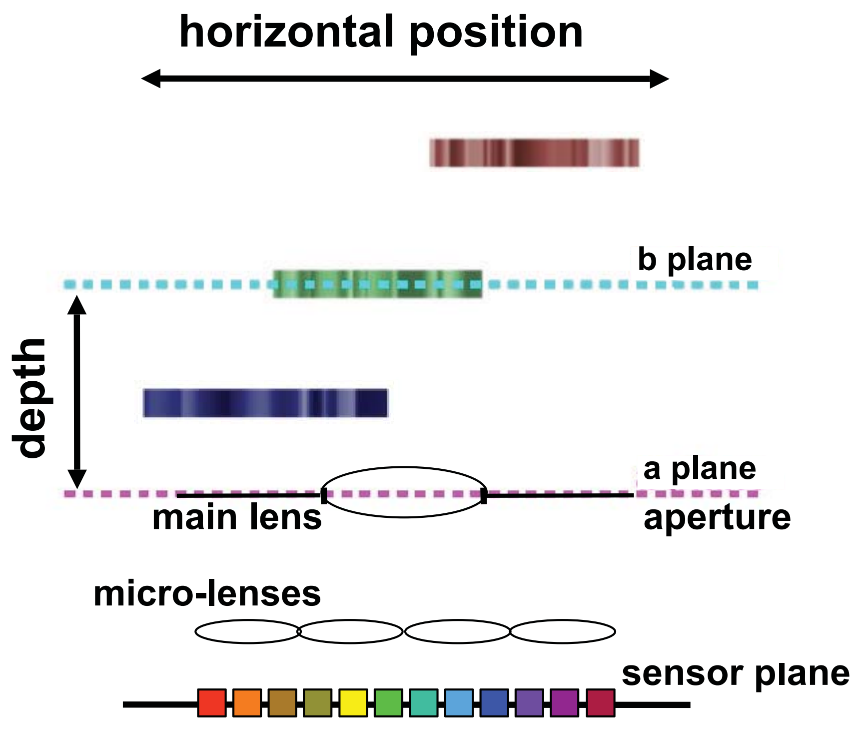


2D lightfield

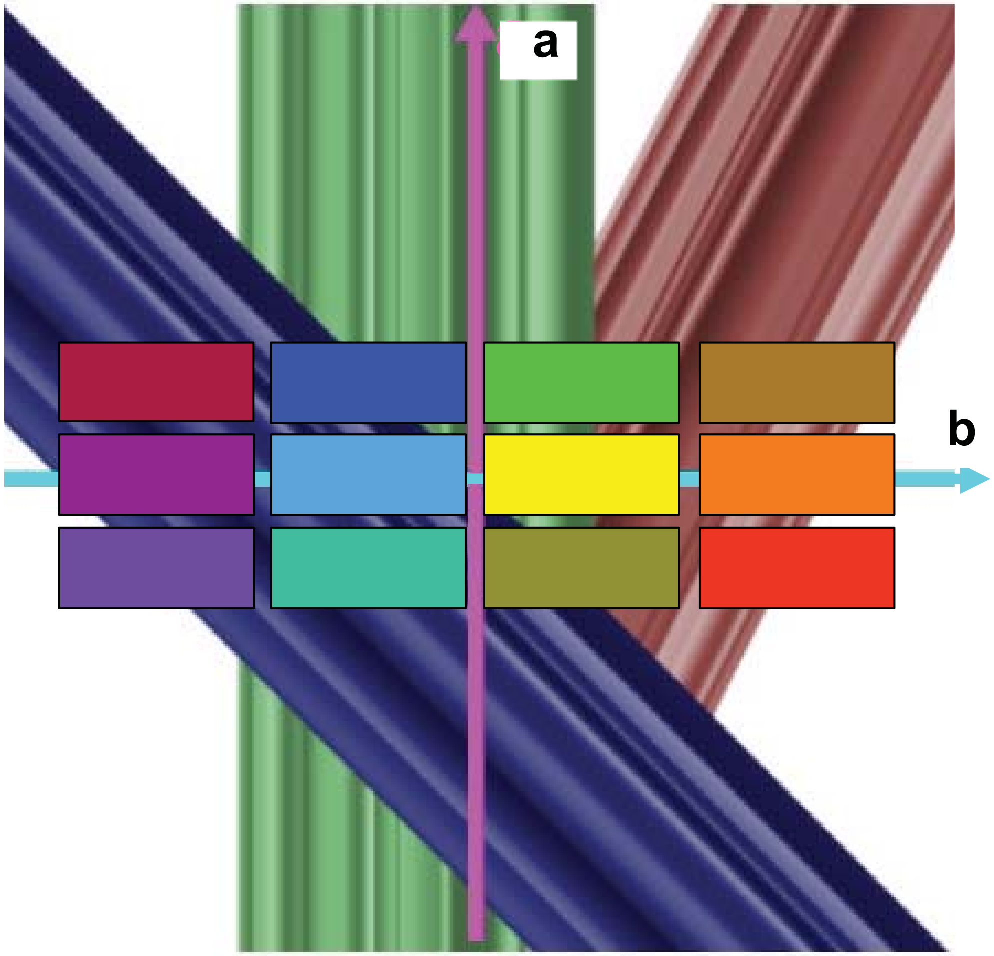


Plenoptic camera

flatworld 1D scene



2D lightfield

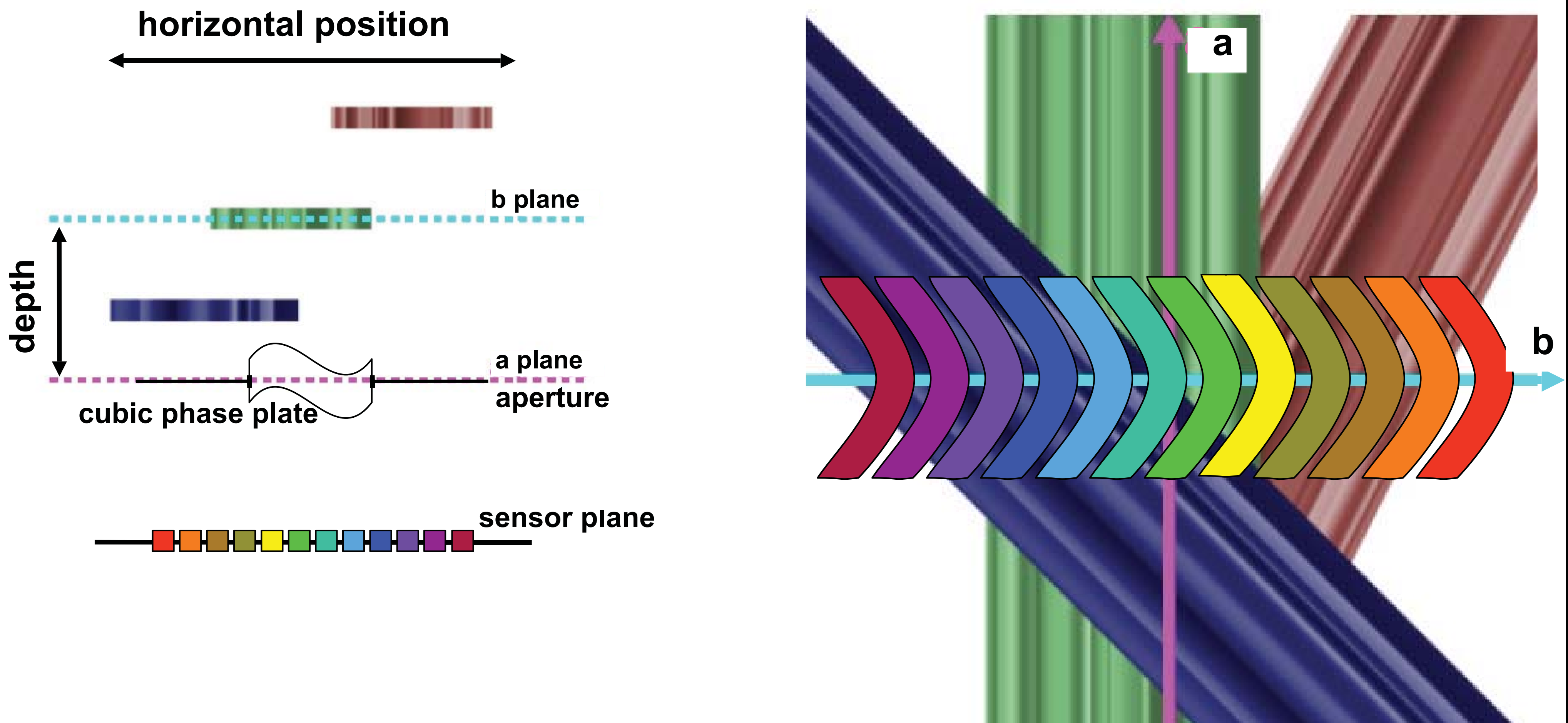


Adelson and Wang 92, Ng et al 05

Wavefront coding

flatworld 1D scene

2D lightfield



Dowski and Cathey,94

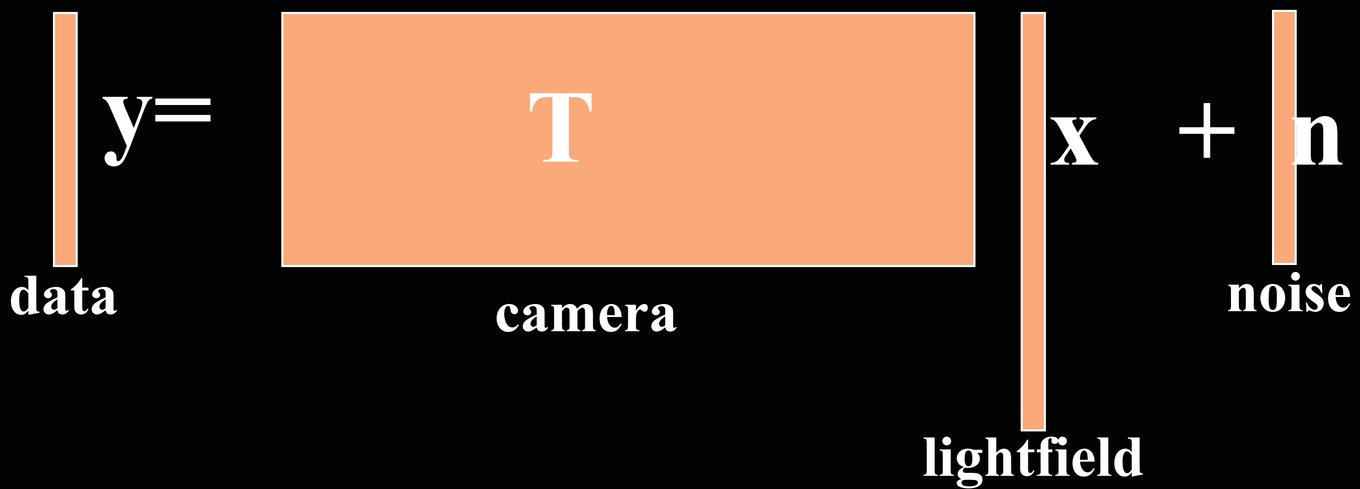
Computational imaging

Camera: Rank deficient projection of a 4D lightfield.

Decoding: ill-posed inversion, need prior on lightfield signals.

Camera evaluation: How well can recover the lightfield from projection?

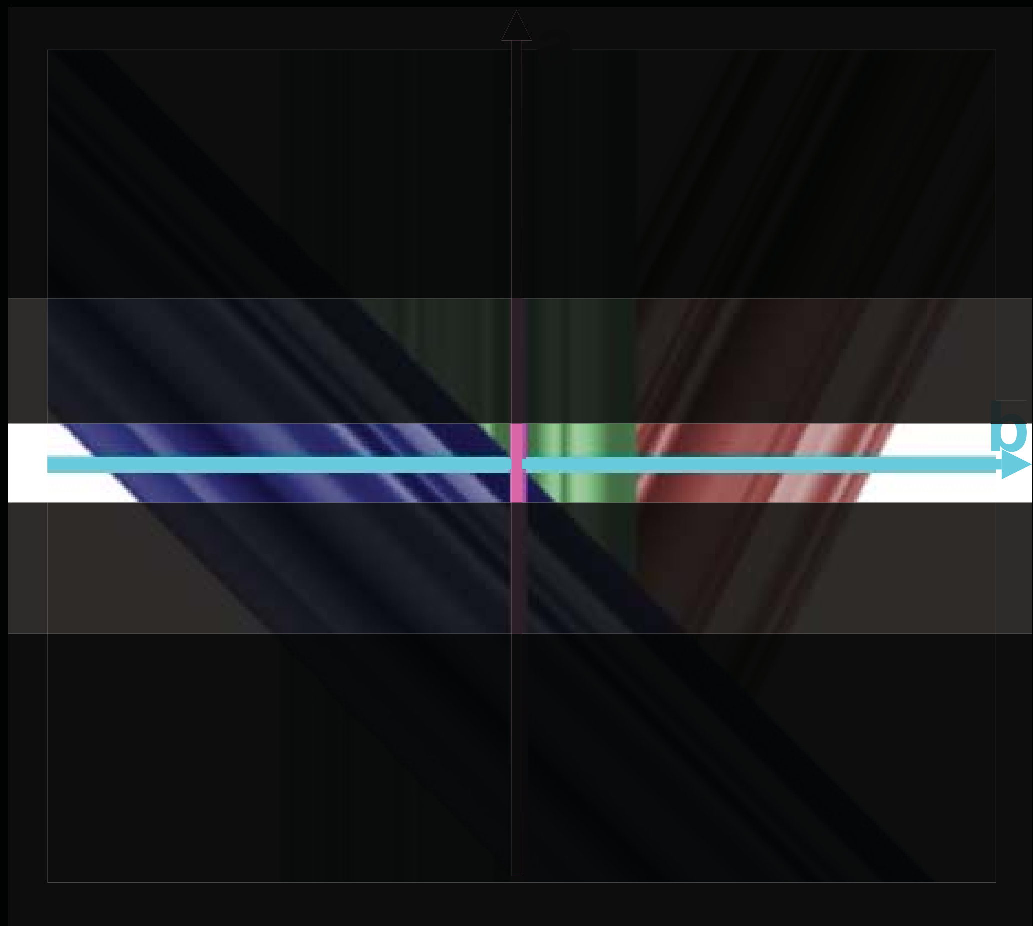
$$\mathbf{y} = \mathbf{T}\mathbf{x} + \mathbf{n}$$



Varying imaging goals by weighted lightfield reconstruction

- Full light field reconstruction (potentially image&depth)
- Reconstruct a bounded view range
- Single row light field reconstruction (pinhole all focused image)

Weigh reconstruction error differently in different light field entries



Bayesian lightfield imaging – Outline

- Specify lightfield reconstruction goals
  - Full lightfield / Single, all-focus view /…
- Specify lightfield prior
- Imaging with one computational camera
  - Specify camera projection matrix
  - Camera decoding - Bayesian inference
- Comparing computational cameras
  - Specify camera projection matrices
  - Evaluate expected error in lightfield reconstruction

Bayesian lightfield imaging – Outline

- Specify lightfield reconstruction goals
  - Full lightfield / Single, all-focus view /…
- **Specify lightfield prior**
- Imaging with one computational camera
  - Specify camera projection matrix
  - Camera decoding - Bayesian inference
- Comparing computational cameras
  - Specify camera projection matrices
  - Evaluate expected error in lightfield reconstruction

Our light field prior:
a mixture of signals at different slopes

Hidden variable S modeling local slope

Conditioning on slope:

- small variance along slope direction
- high variance along spatial direction

Light field prior is a mixture of oriented Gaussians (MOG):

$$P(x) = \int_S P(S)P(x \mid S)$$

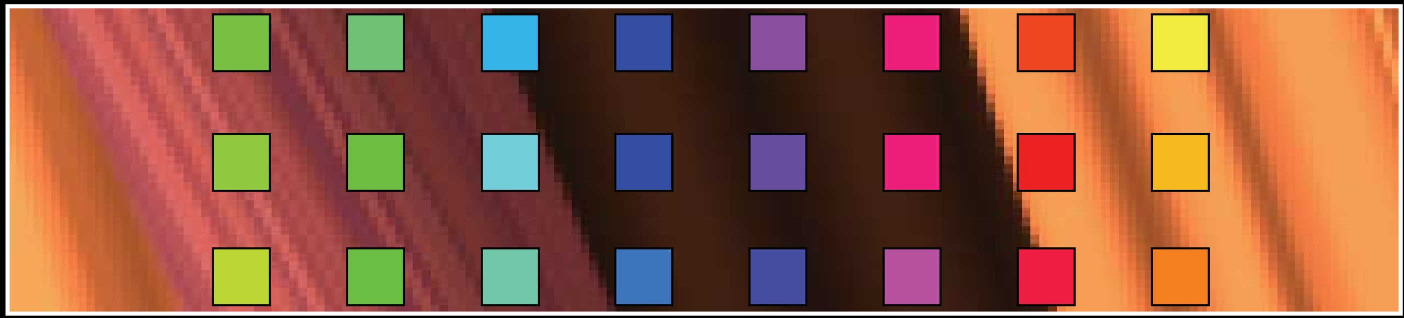
Piecewise smooth prior on slopes

Given slope, lightfield prior is Gaussian and simple

Bayesian lightfield imaging – Outline

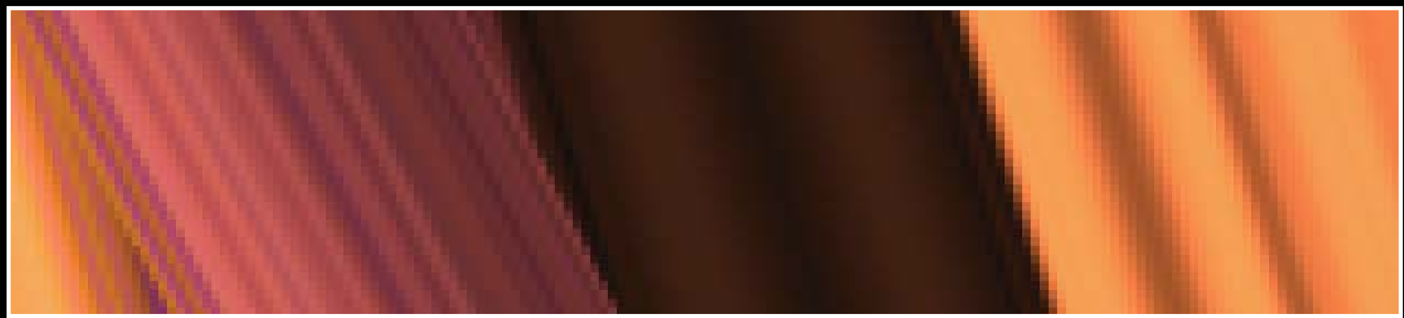
- Specify lightfield reconstruction goals
  - Full lightfield / Single, all-focus view /…
- Specify lightfield prior
- Imaging with one computational camera
  - Specify camera projection matrix
  - Camera decoding - Bayesian inference
- Comparing computational cameras
  - Specify camera projection matrices
  - Evaluate expected error in lightfield reconstruction

Prior effect on reconstruction



Band-limited reconstruction to account for unknown depth

See paper for inference details



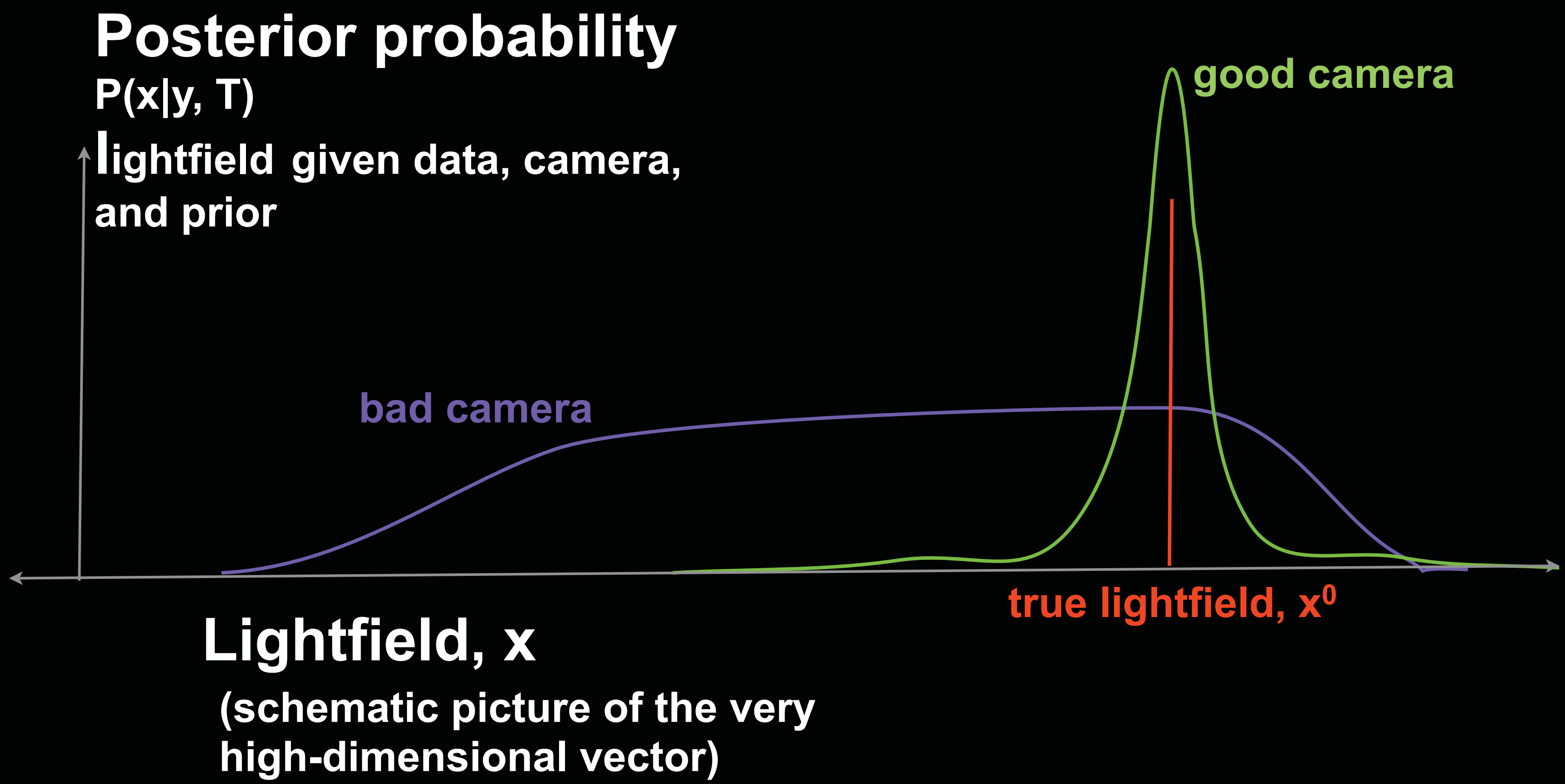
Reconstruction using light field prior

Bayesian lightfield imaging – Outline

- Specify lightfield reconstruction goals
  - Full lightfield / Single, all-focus view /…
- Specify lightfield prior
- Imaging with one computational camera
  - Specify camera projection matrix
  - Camera decoding - Bayesian inference
- Comparing computational cameras
  - Specify camera projection matrices
  - Evaluate expected error in lightfield reconstruction

Camera evaluation

Goal: evaluate inherent ambiguity of a camera projection, independent of inference algorithm



Camera evaluation function: expected squared error

$$E_{P(x|y;T)}\left[\| x - x^0 \|^2\right] = \int_x P(x \mid y;T) \| x - x^0 \|^2$$

With our mixture model prior, conditioned on the lightfield slopes S, everything is Gaussian and analytic. So let's write the posterior as:

$$P(x \mid y;T) = \int_S P(S \mid y;T) P(x \mid y,S;T)$$

Then our expected squared error becomes an integral over all slope fields:

$$E_{P(x|y;T)}\left[\| x - x^0 \|^2\right] = \int_S P(S \mid y;T) E_{P(x|y,S;T)}\left[\| x - x^0 \|^2\right]$$

Approximate by Monte Carlo sampling near the true slope field:

$$E_{P(x|y;T)}\left[\| x - x^0 \|^2\right] \approx \sum_{S_i} P(S_i \mid y;T) E_{P(x|y,S_i;T)}\left[\| x - x^0 \|^2\right]$$

Bayesian camera evaluation tool

Input parameters:

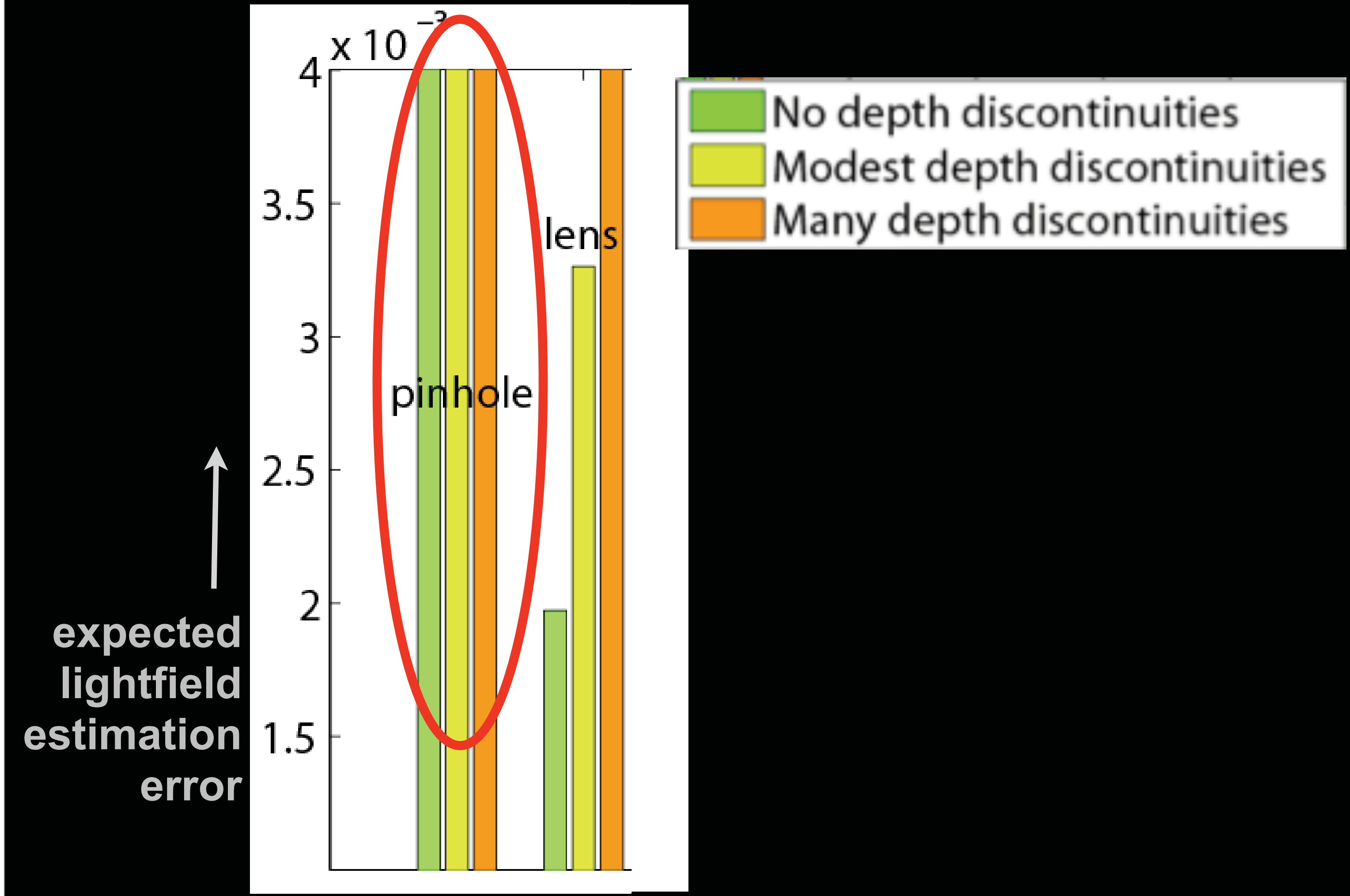
- **Reconstruction goals (weight on light field entries)**
- **Camera matrix**
- **Noise level**
- **Spatial and depth resolution**

Output: expected reconstruction error

Matlab software online:

people.csail.mit.edu/alevin/papers/lightfields-Code-Levin-Freeman-Durand-08.zip

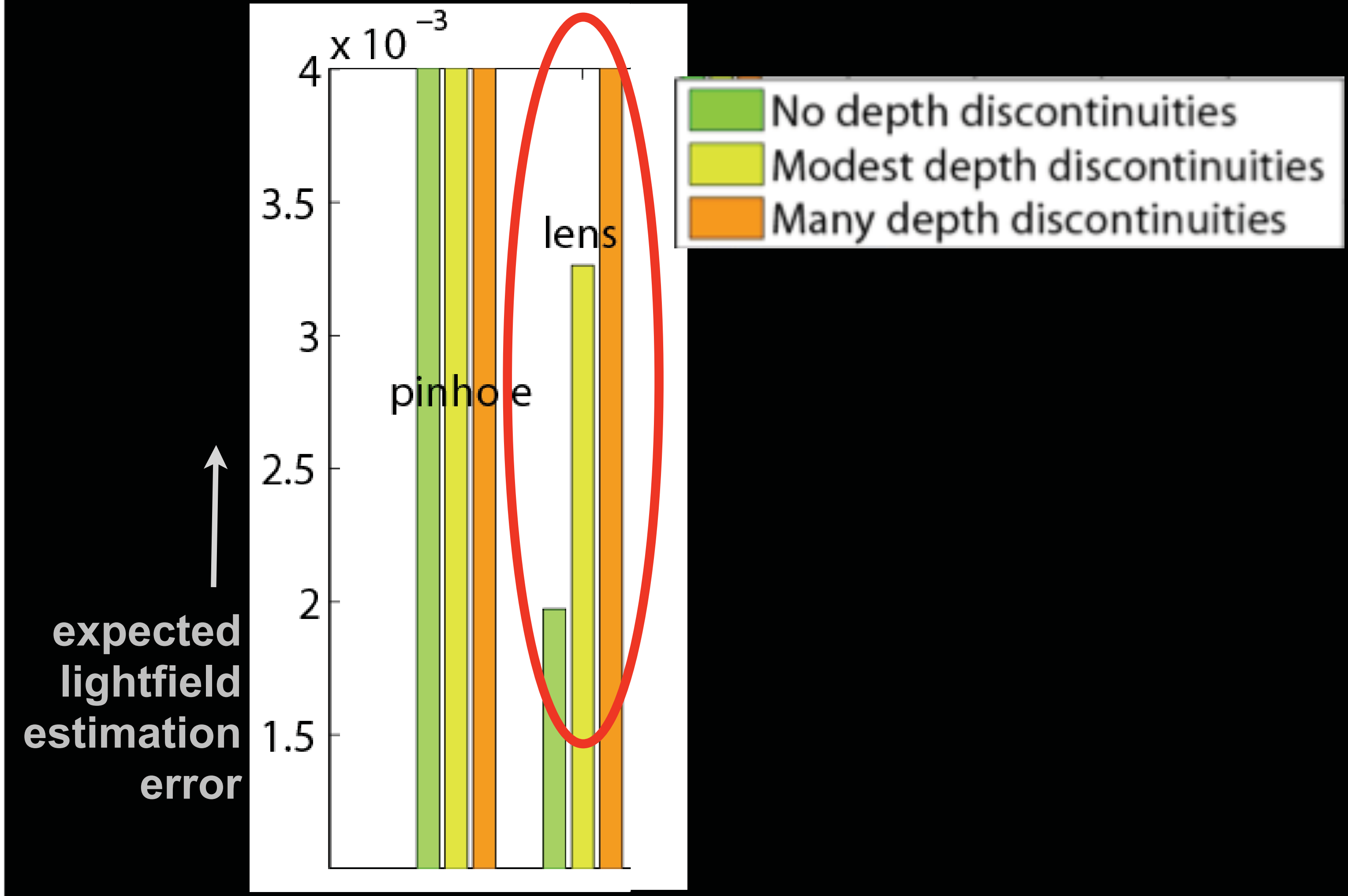
1D camera evaluation– full light field reconstruction



Observation:

As expected, a pinhole camera doesn't estimate the lightfield well

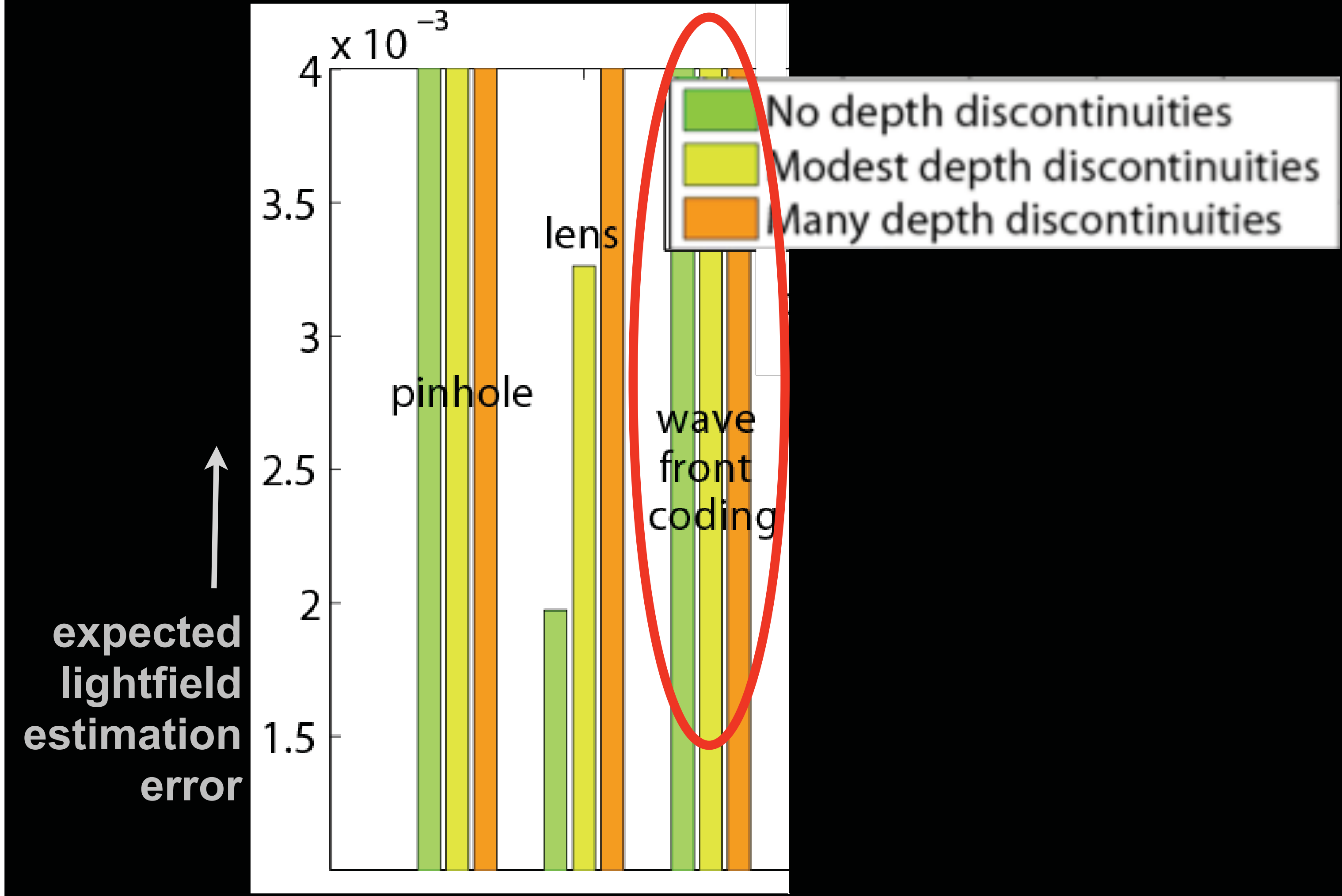
1D camera evaluation– full light field reconstruction



Observation:

When depth variation is limited, some depth from defocus exist in a single monocular view from a standard lens

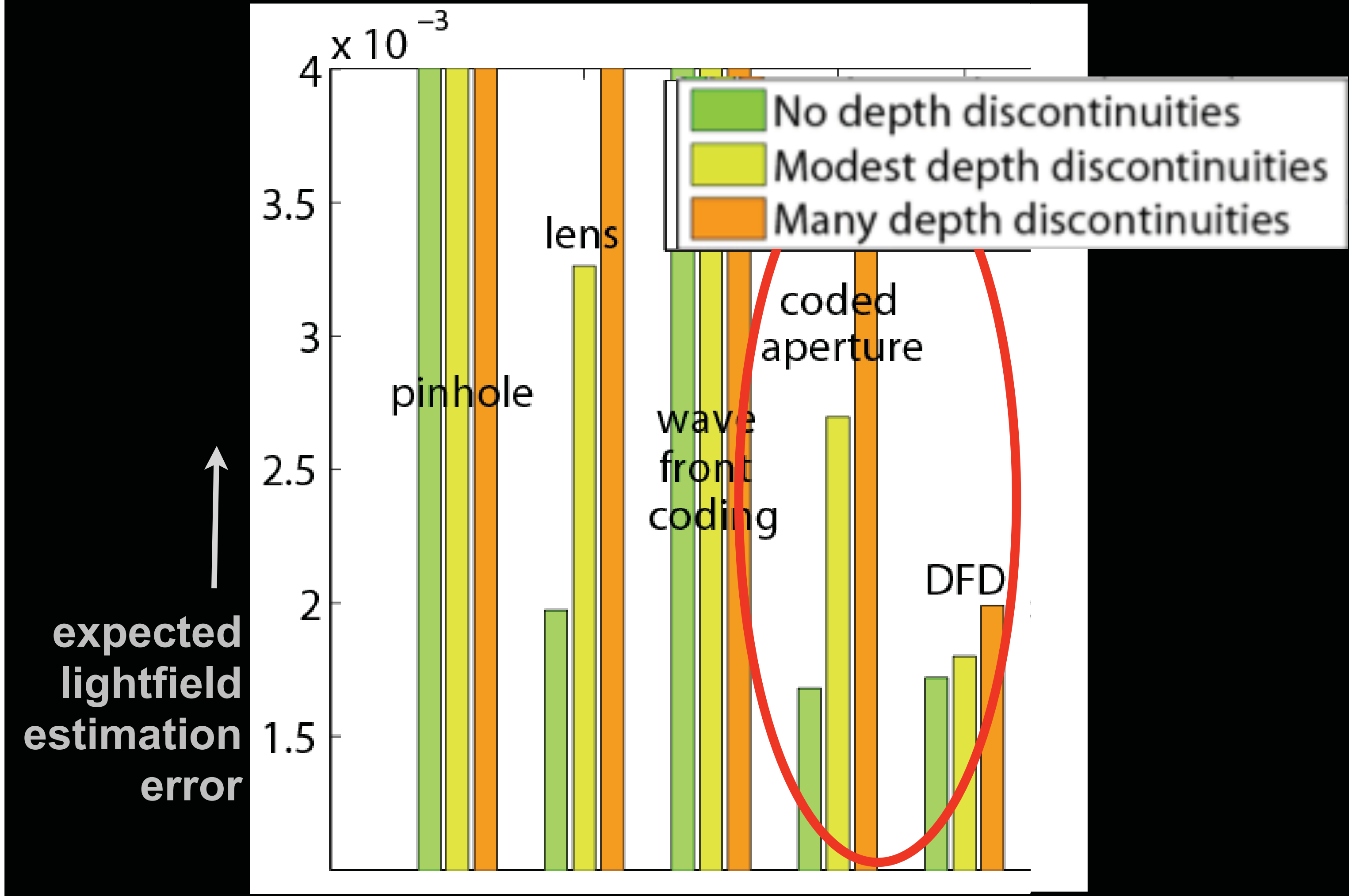
1D camera evaluation– full light field reconstruction



Observation:

Wavefront coding, not designed to estimate the lightfield, doesn't.

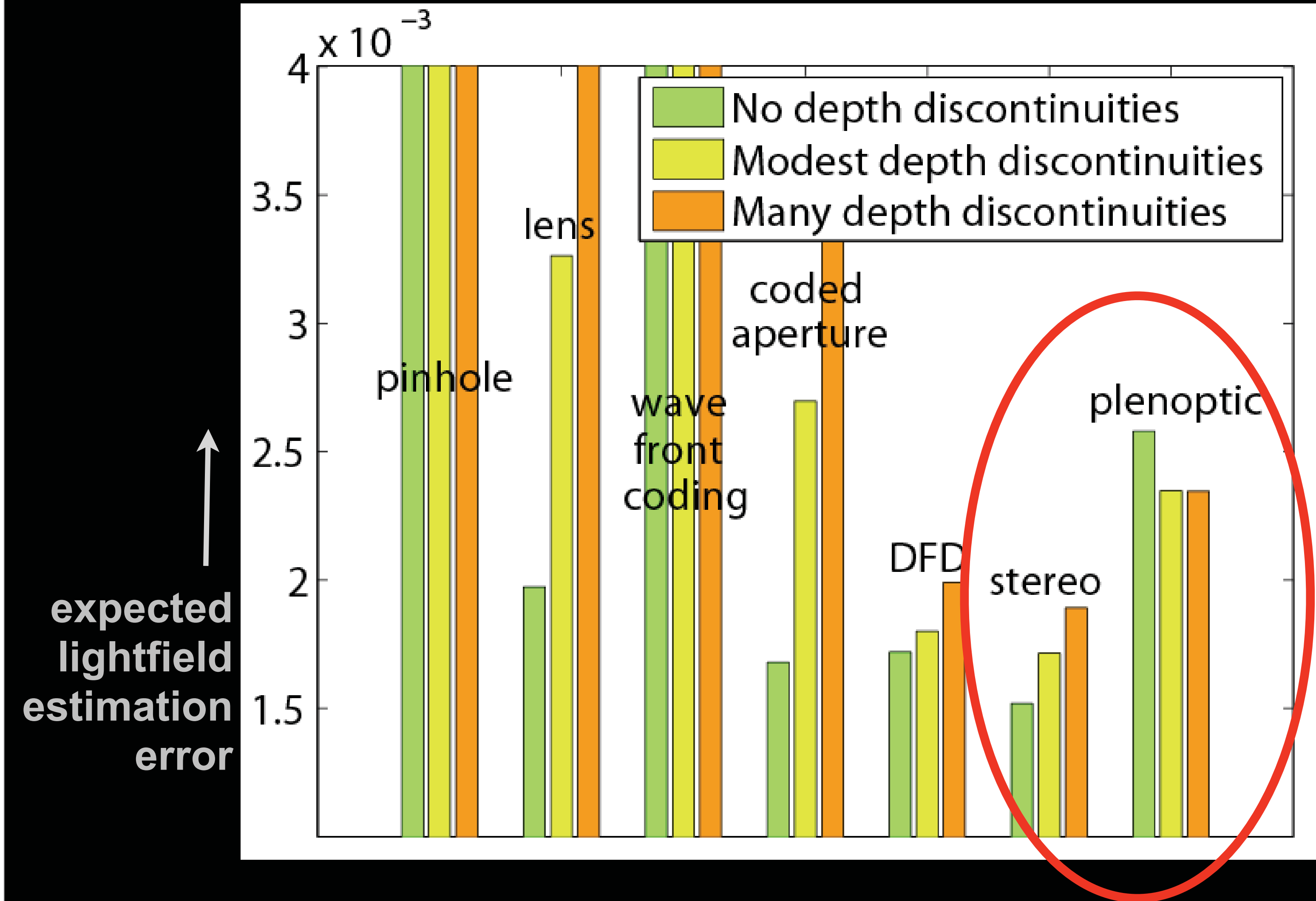
1D camera evaluation– full light field reconstruction



Observation:

Depth-from-defocus (DFD) outperforms the coded aperture at these settings

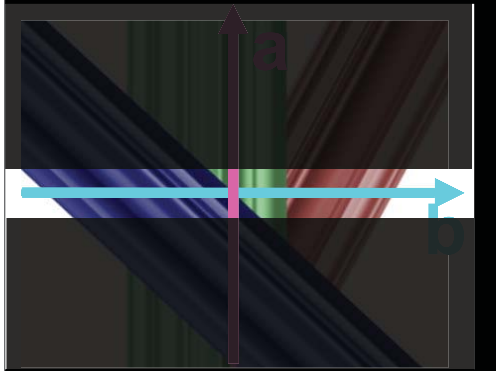
1D camera evaluation– full light field reconstruction

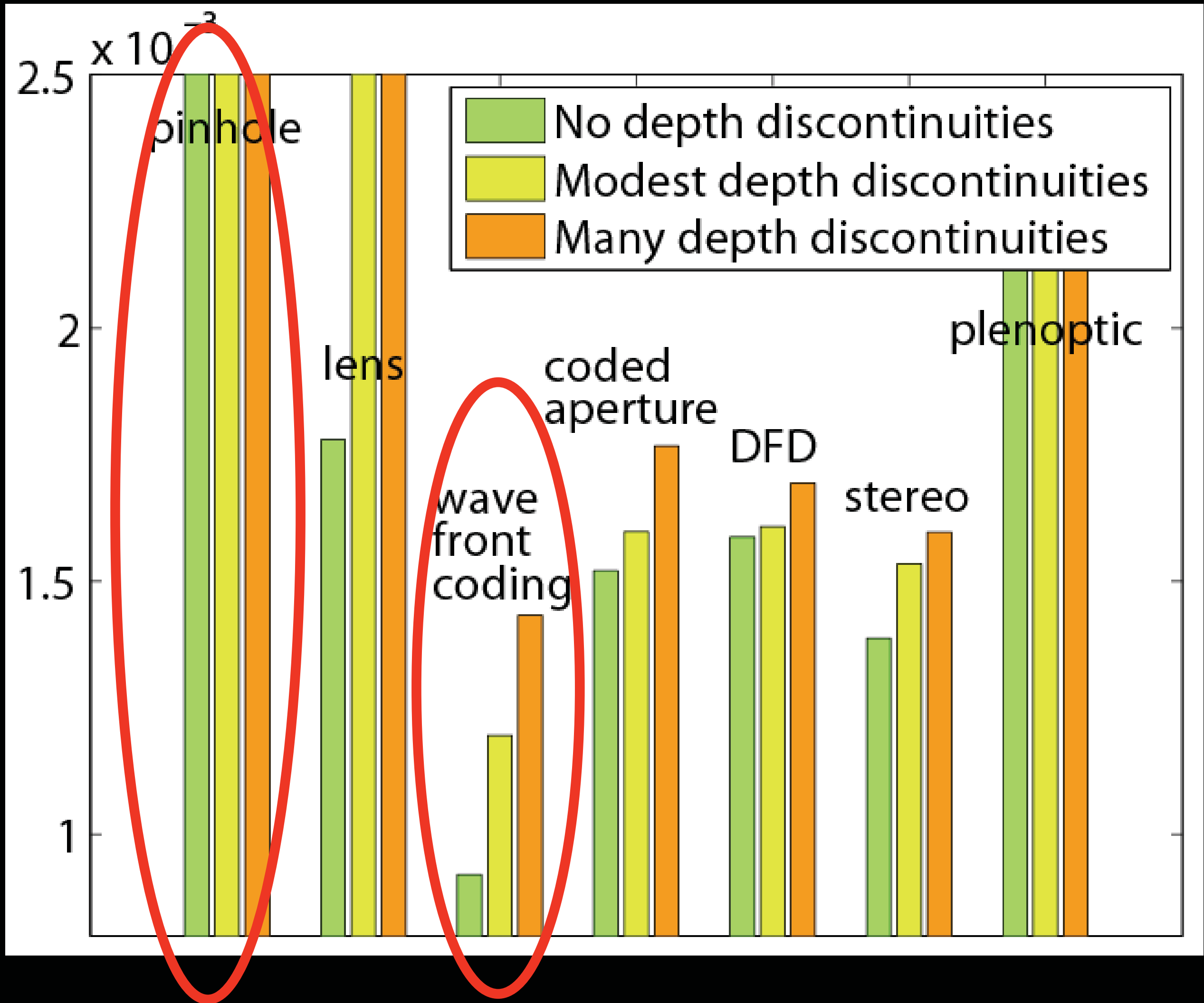


Observation: Stereo error is less than Plenoptic

Since depth variation is smaller than texture variation, no need to sacrifice so much spatial resolution to capture directional information

1D camera evaluation– single row reconstruction





expected lightfield estimation error

Observations:

Pinhole camera- poor estimation due to noise

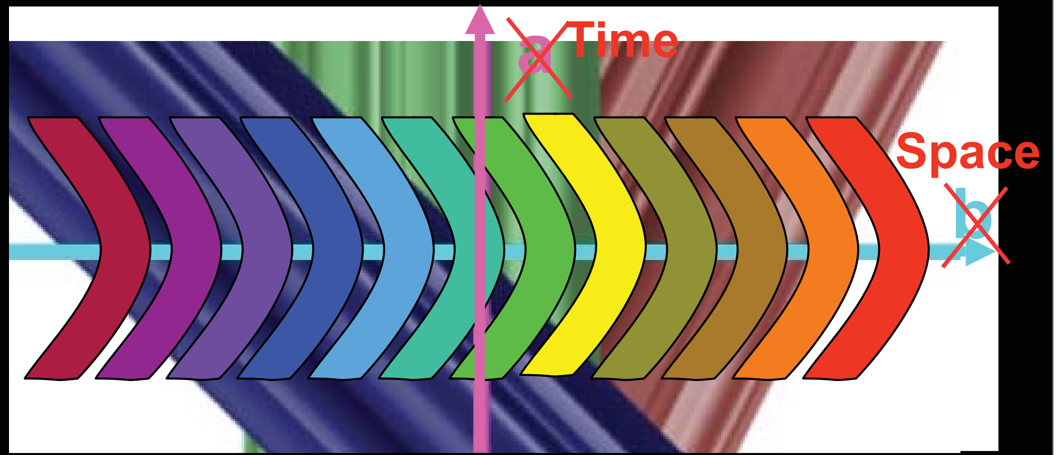
Wavefront coding- no depth information, but accurate reconst for a single view

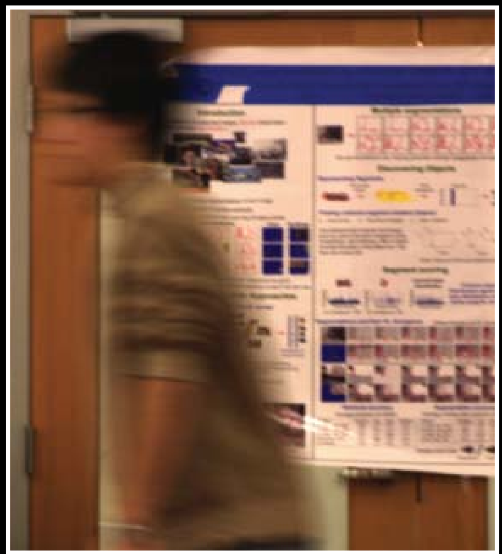
Application: motion invariant photography

Depth invariant integration

Motion invariant integration

SIGGRAPH 2008, Levin et al.





Static camera

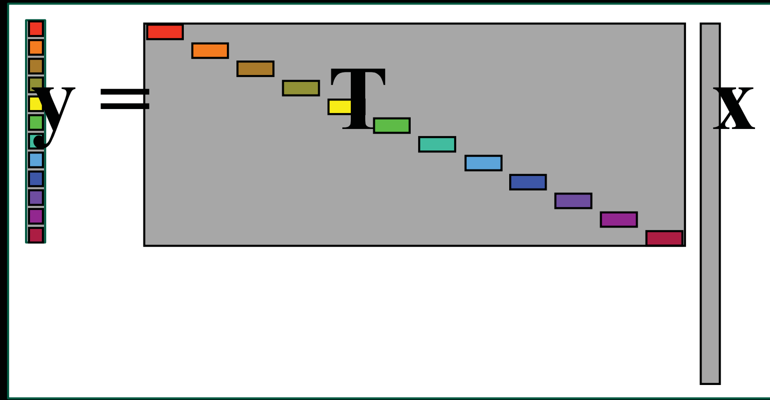
motion invariant

output after

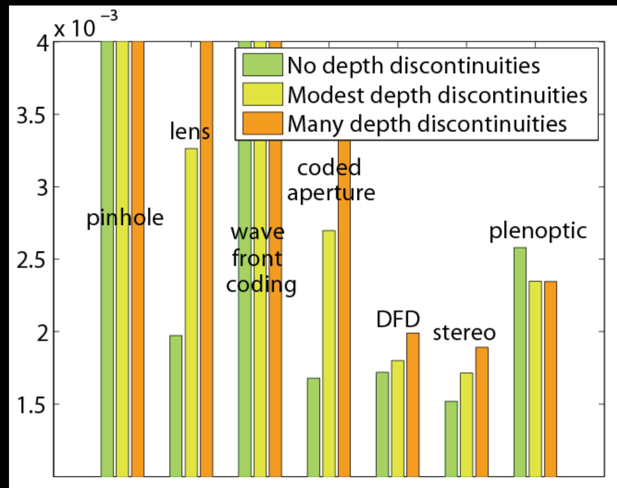
speed-invariant blur allows non-blind deconvolution

Summary: Bayesian lightfield imaging

- Model imaging as linear light field projection

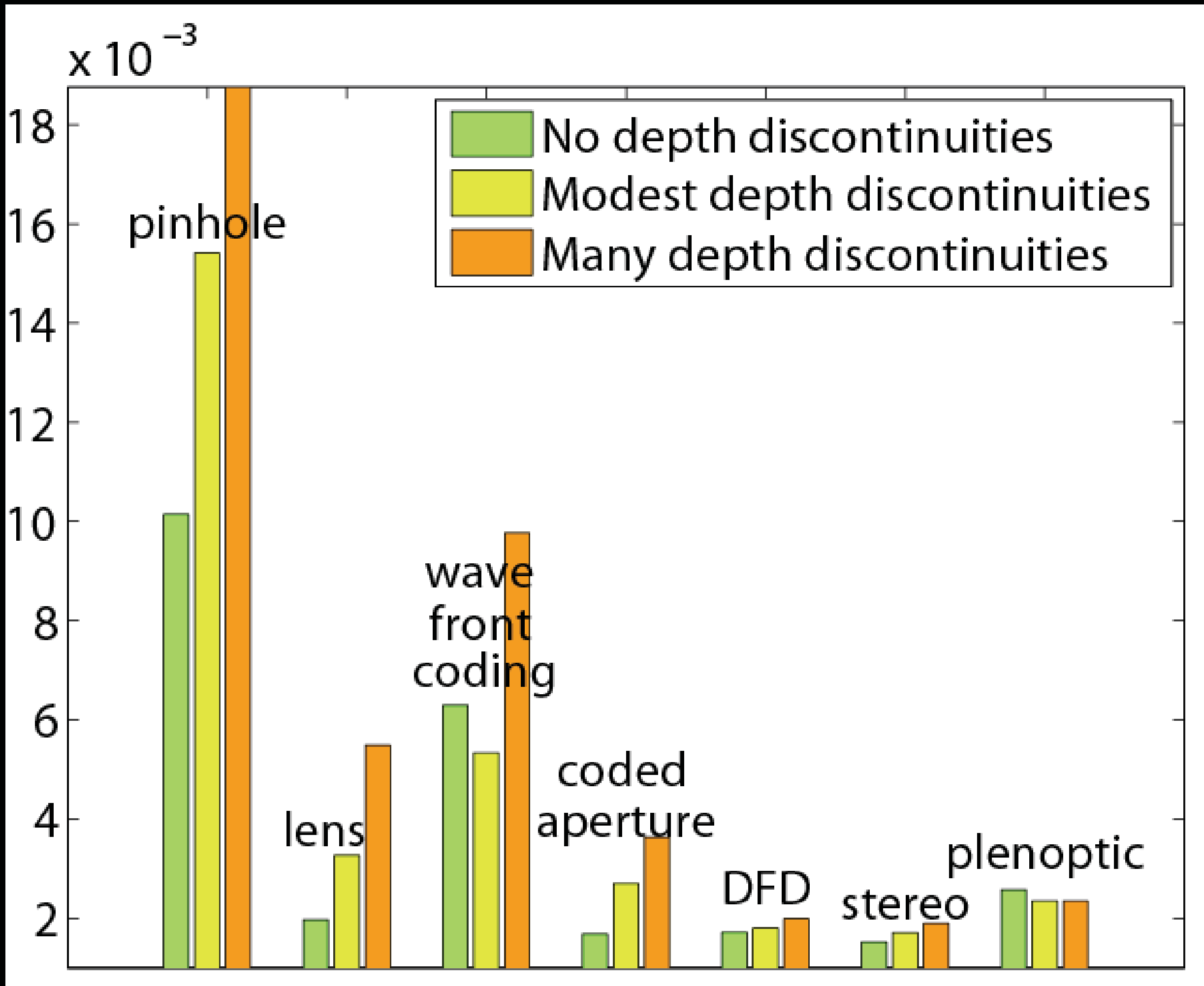


- New prior on light field signals
- Camera decoding expressed as a Bayesian inference problem



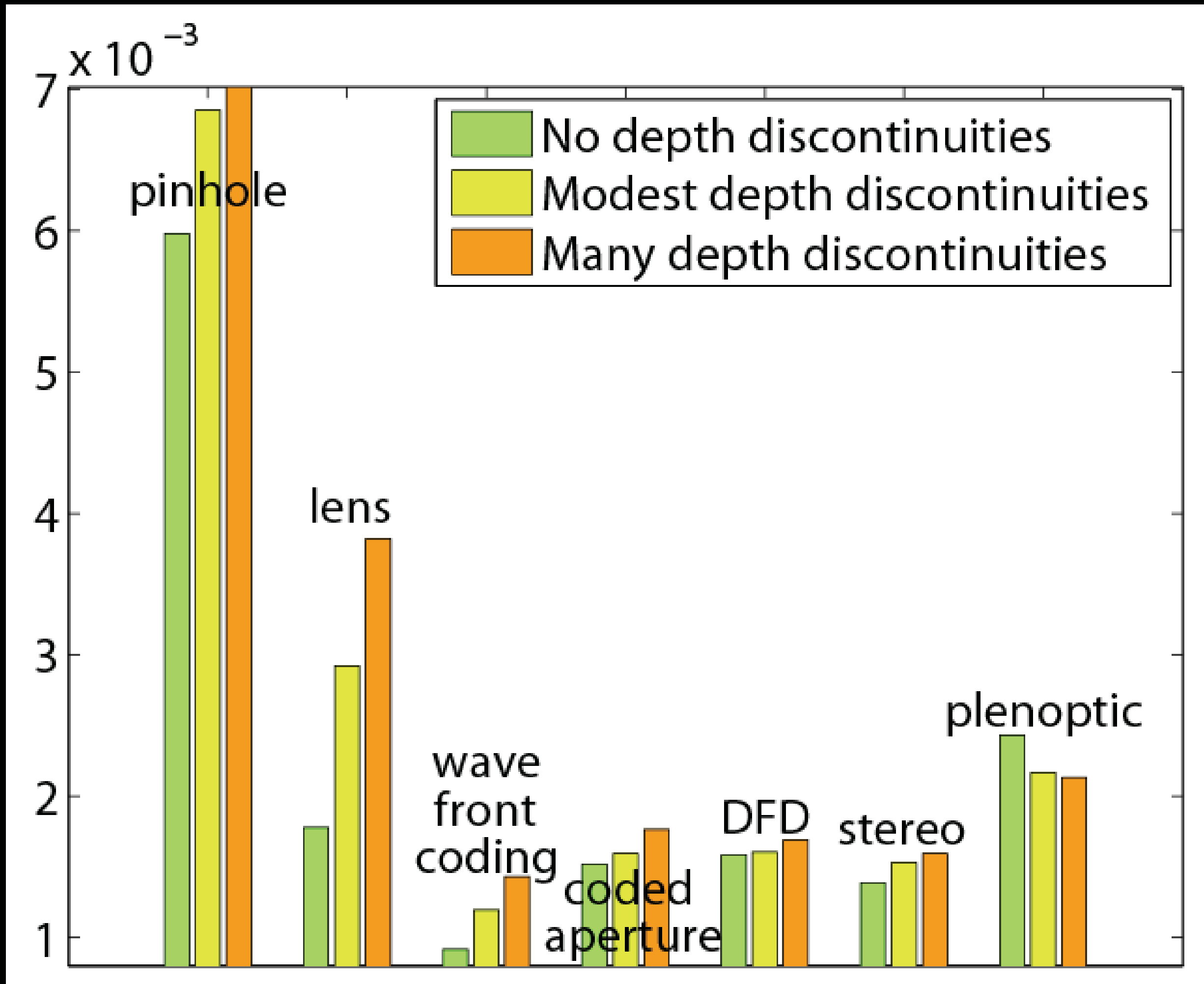
- Framework and software for comparison across camera configurations, by evaluating uncertainty in light field reconstruction
- Principled novel camera design

full lightfield reconstruction, unclipped plot



Full lightfield

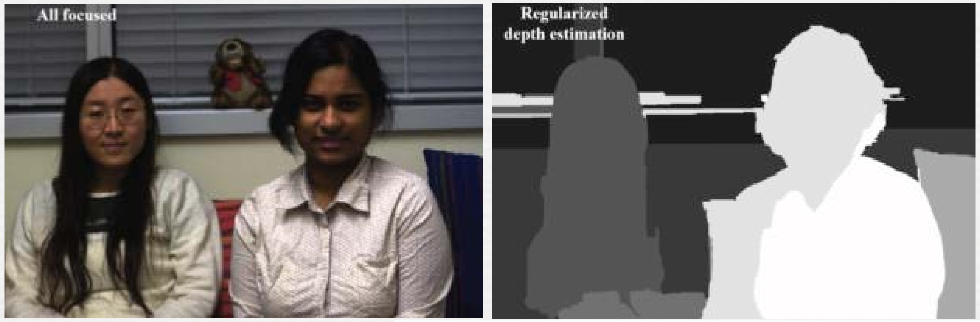
single-row reconstruction, unclipped plot



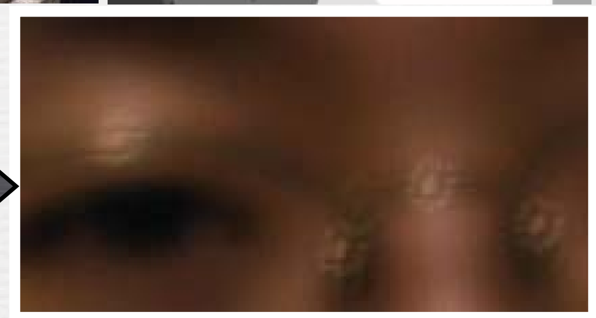
single row

Computational photography

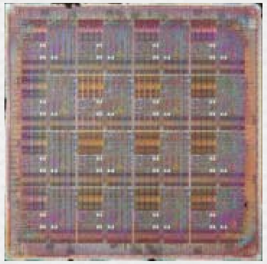
- Reconstruct
  more information
  (light-field / depth)

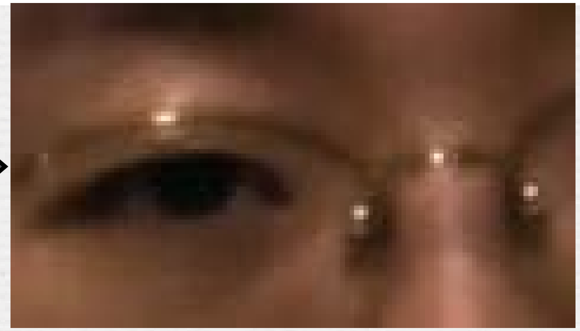


- Coding optics,



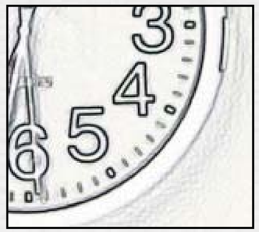
  decoding algorithm





- Prior on "natural" images
  (sparse derivative)

$$\sum ||\nabla x||^{0.8}$$



Code search

- Design constraints:
  - Binary pattern
  - Minimum hole size
    => 1mm^2 (due to diffraction)
  - No floating parts

➡ Sample patterns and optimize depth inference (KL divergence)

➡ Formal derivation of score function in paper