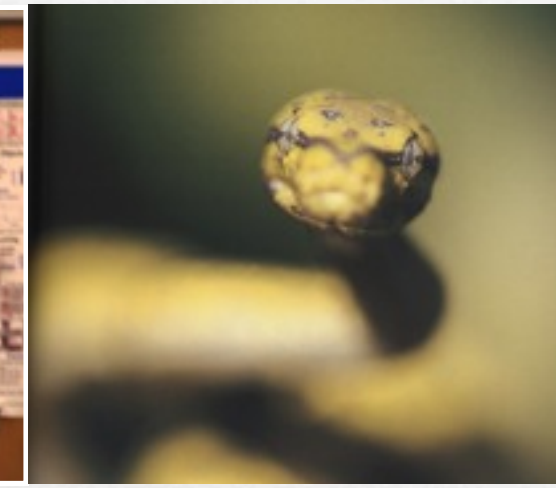
Frequency Analysis in the Space-Time and Light Field Domains

Frédo Durand
MIT CSAIL

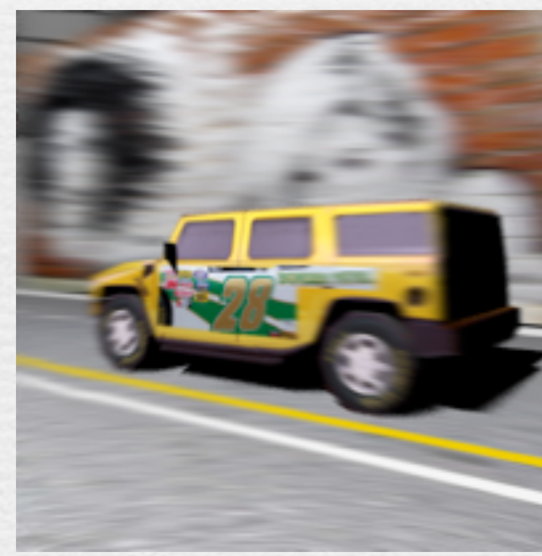
in collaboration with Levin, Freeman, Green, Hasinoff, Egan, Ramamoorthi, Holzschuch, Soler, Subr, Sil lion, Chan, Zwicker, Matusik, Tseng, Cho, Sand

Image capture and synthesis

- Image capture
  - Goal: acquire sharper images
  - Degradations: noise, motion blur, defocus blur, etc.

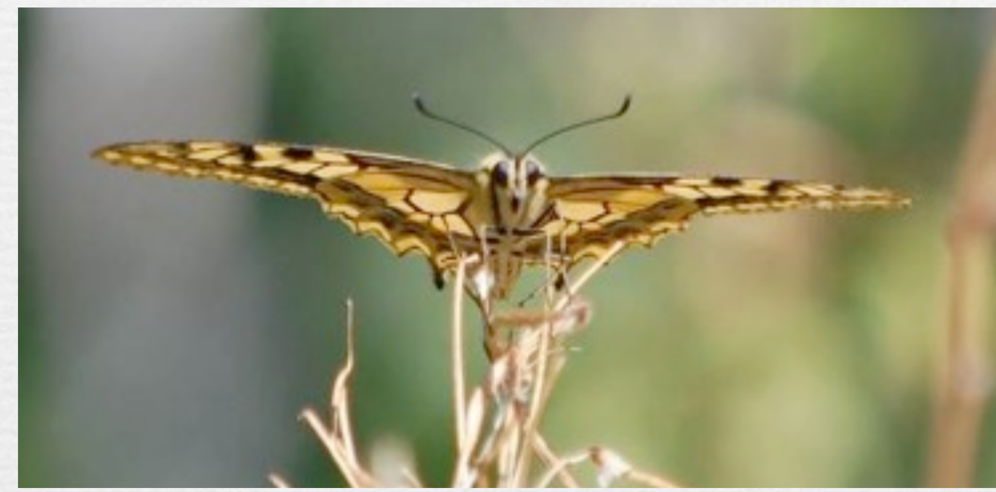
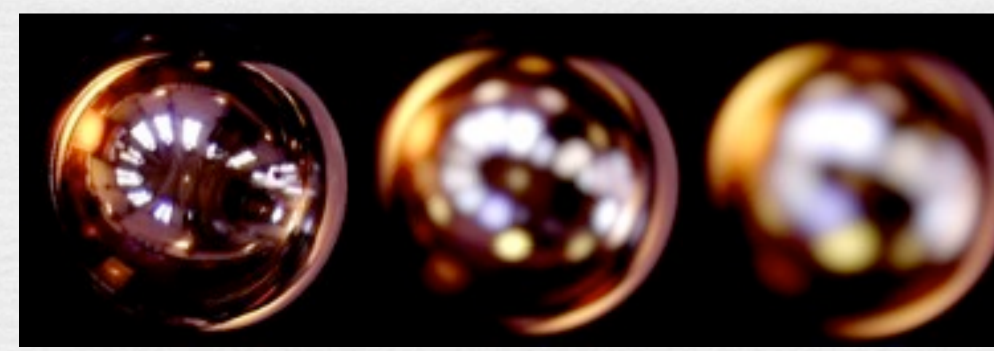
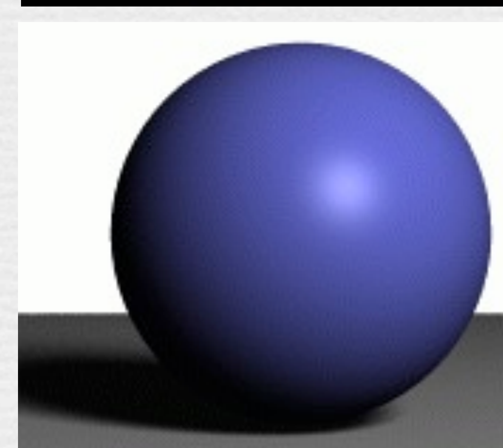
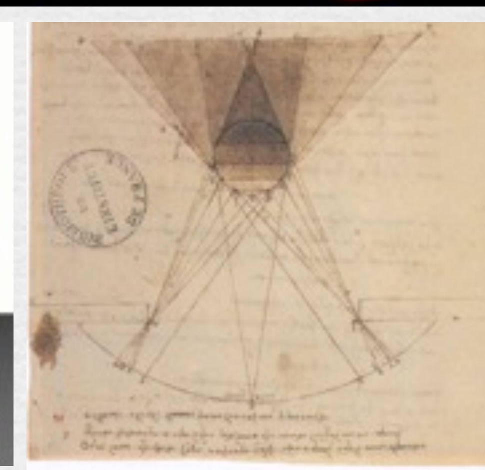
- Image synthesis
  - Goal: reduce simulation cost
  - Complex light transport
  - Complex imaging: simulate defocus and motion blur




- **Better understanding of image formation can lead to efficient synthesis algorithms and capture strategies**
  - **Fourier analysis!**

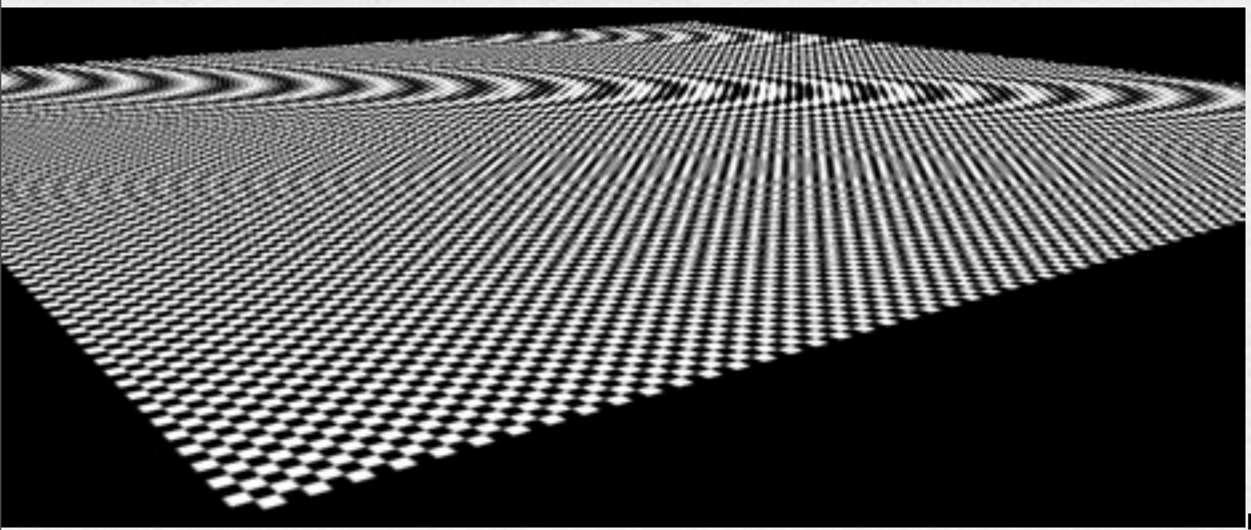
Blurring phenomena

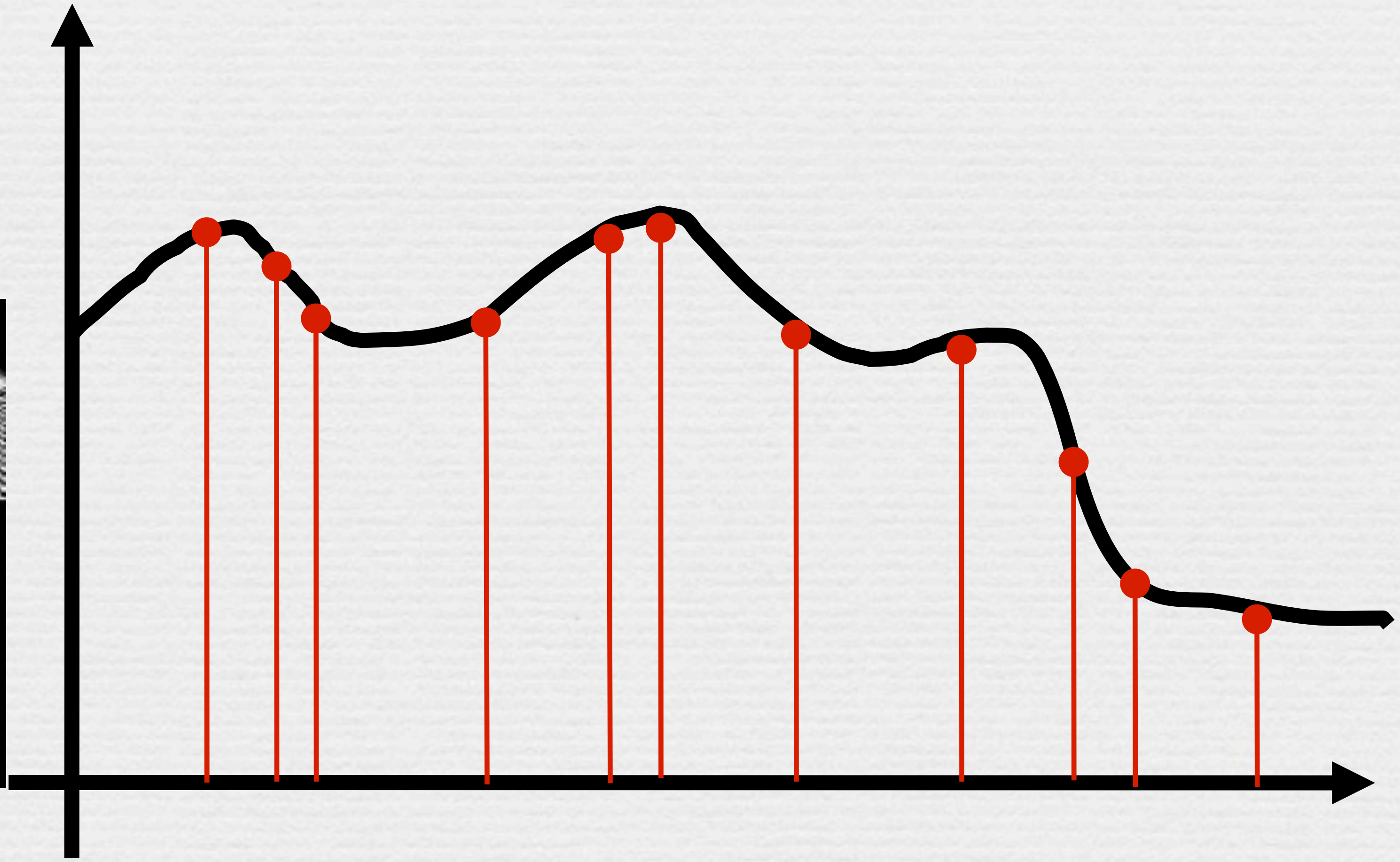
- Motion blur
  - integral over shutter time
- Defocus blur
  - integral over aperture
- Glossy highlights
  - integral over incident light
- Soft shadows
  - integral over light source

Sampling

- Digital image sampling & aliasing
- Monte Carlo integration: random sampling





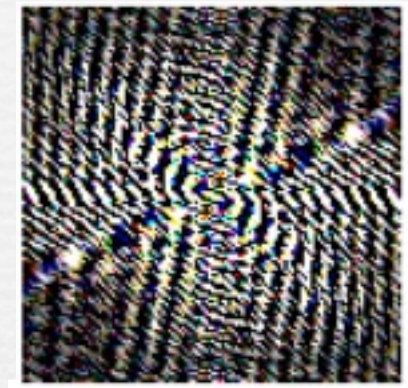
Fourier to the rescue!

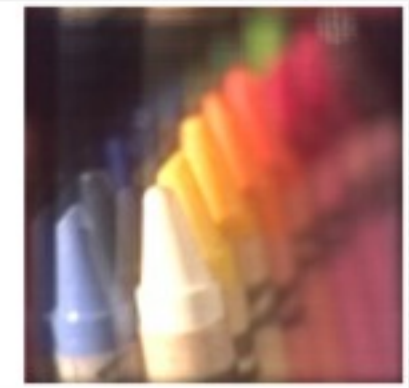
- Both blur and sampling are well expressed in the Fourier domain
- Studied in appropriate space:
  - Space-time for motion blur
  - 4D light field for depth of field

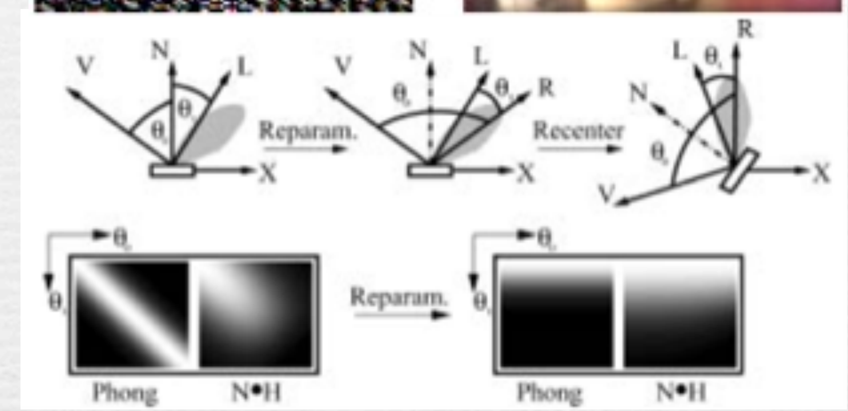
- Disclaimer: different from Fourier optics... and I will sweep diffraction under the rug

Standing on the shoulders of giants

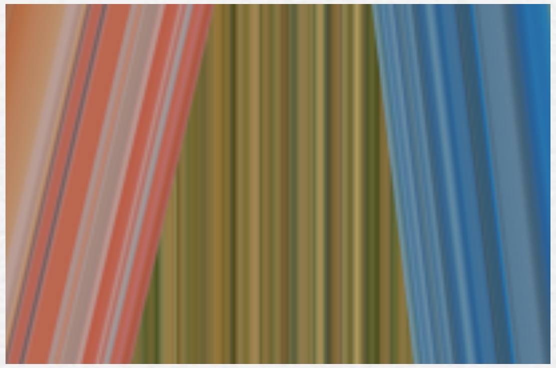
- Fourier analysis in graphics
  - Light field sampling [Chai, Isaaksen, Ng,...]
  - Signal processing for reflection [Ramamoorthi, Basri,...]
- PSF engineering
  - Wavefront coding [Cathey & Dowski, ...]
  - Flutter shutter [Raskar]
  - ...







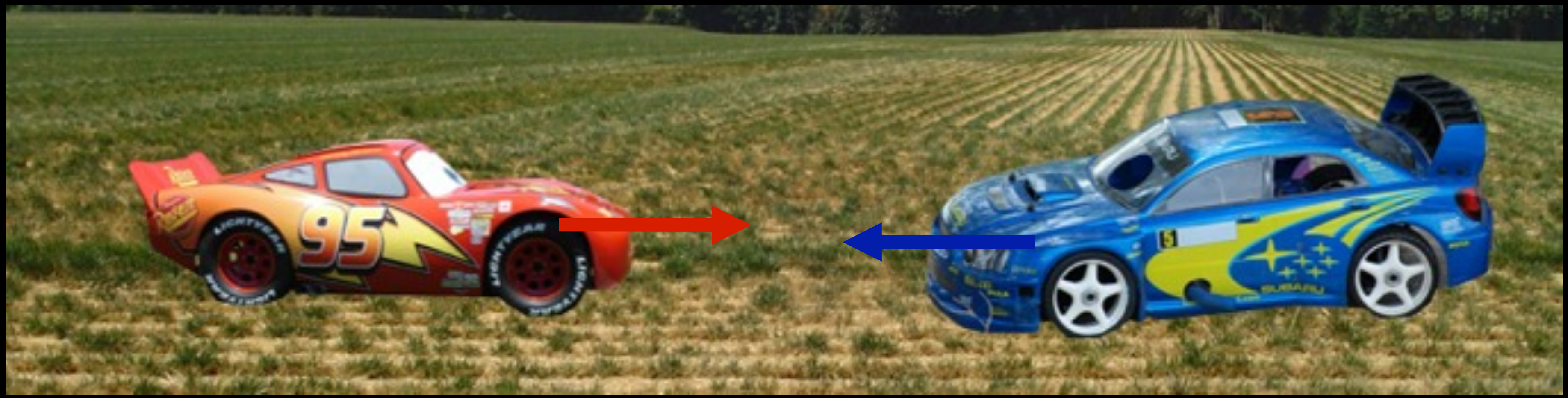
Basic Recipe



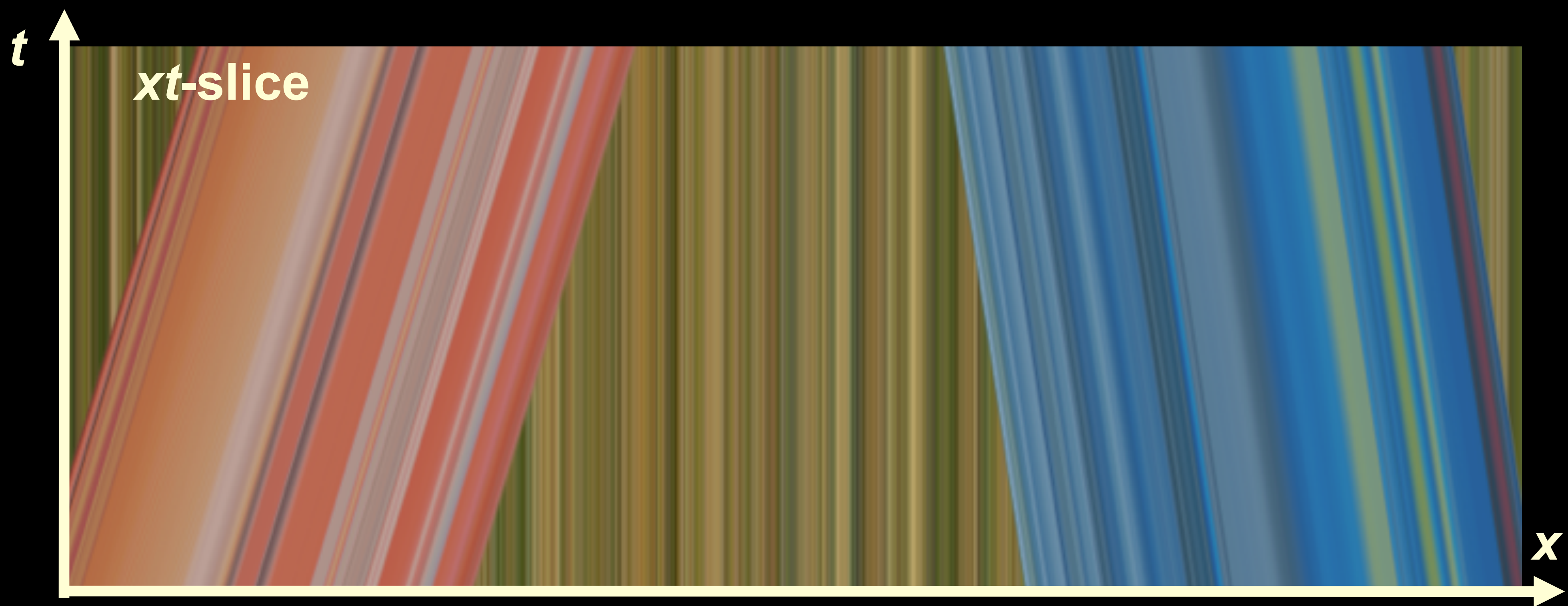
- Use high dim Fourier space
  - Signal is anisotropic
    (slope depends on speed/distance)
  - Imaging = integral = convolution => bandlimits
- Capture
  - Use deconvolution to remove blur
  - Modify optics to bandlimit less
- Synthesis
  - Exploit bandwidth & anisotropy for sparse sampling

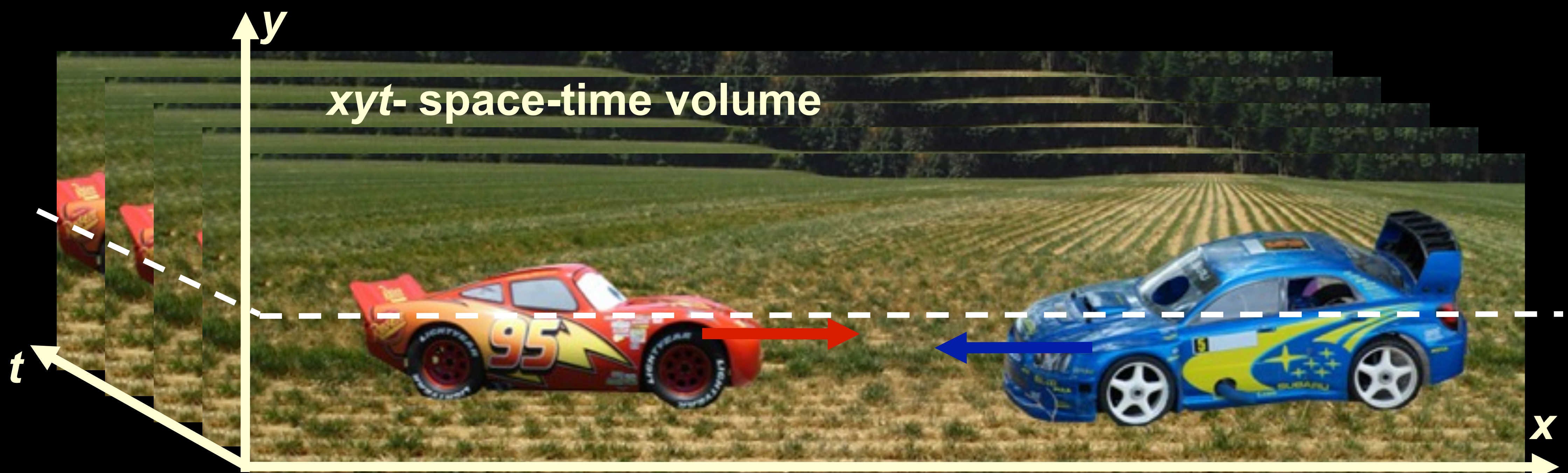
First, let's look at motion blur

Scene with motion



The space time volume

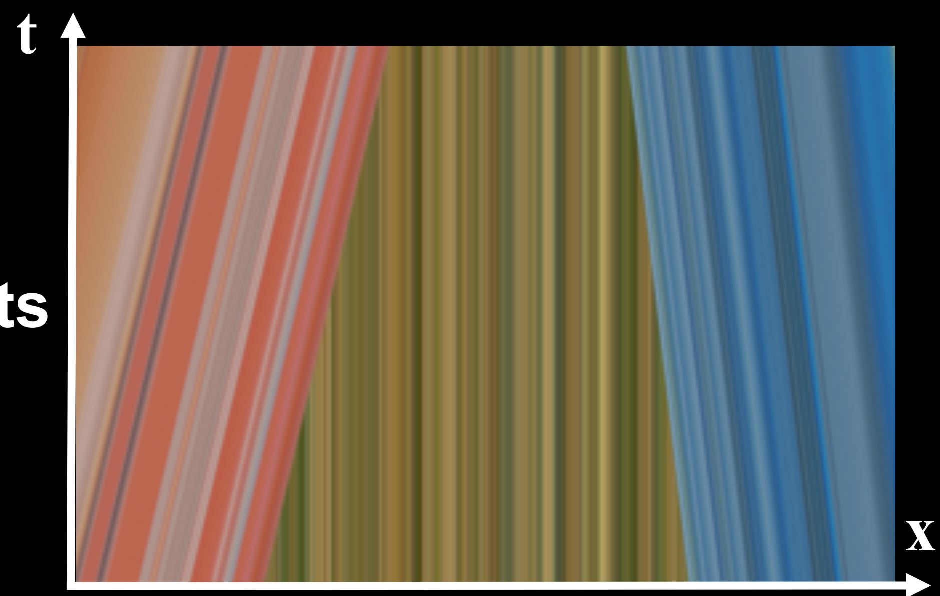




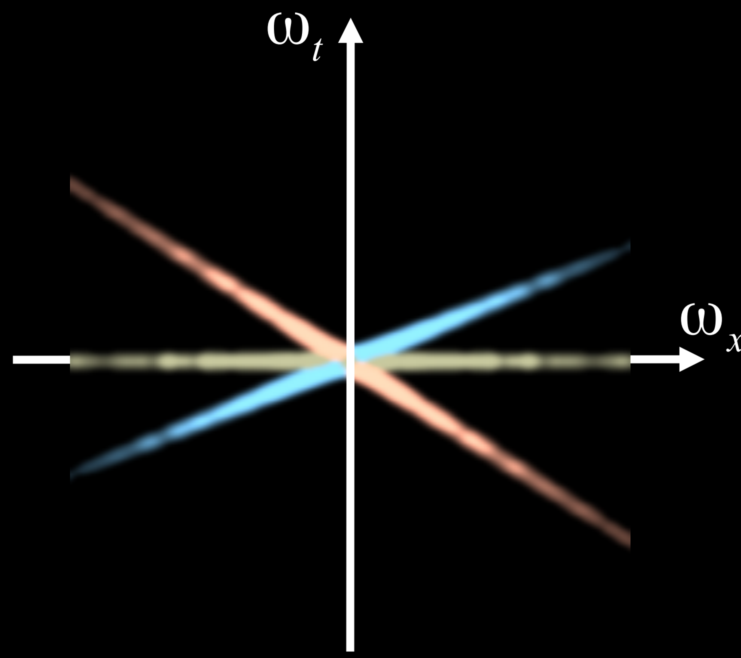
Space-time Fourier domain

Primal Domain

Objects



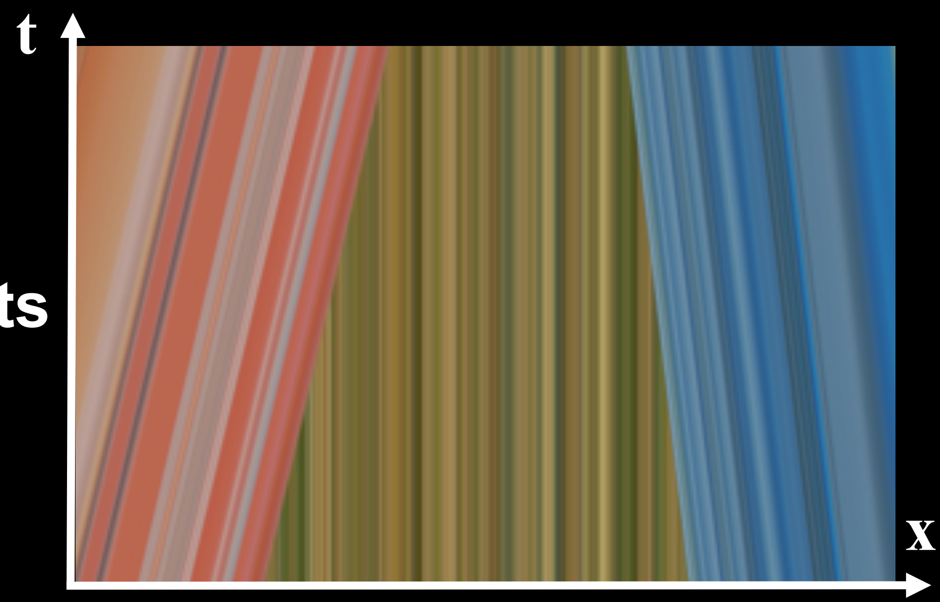
Frequency Domain



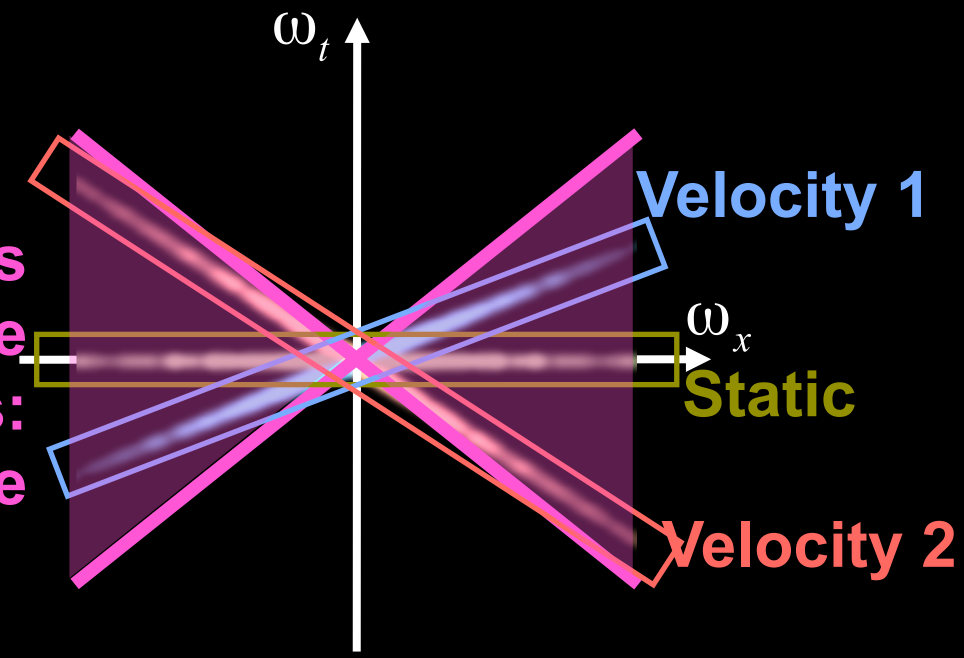
Slope = 1/velocity

Primal Domain

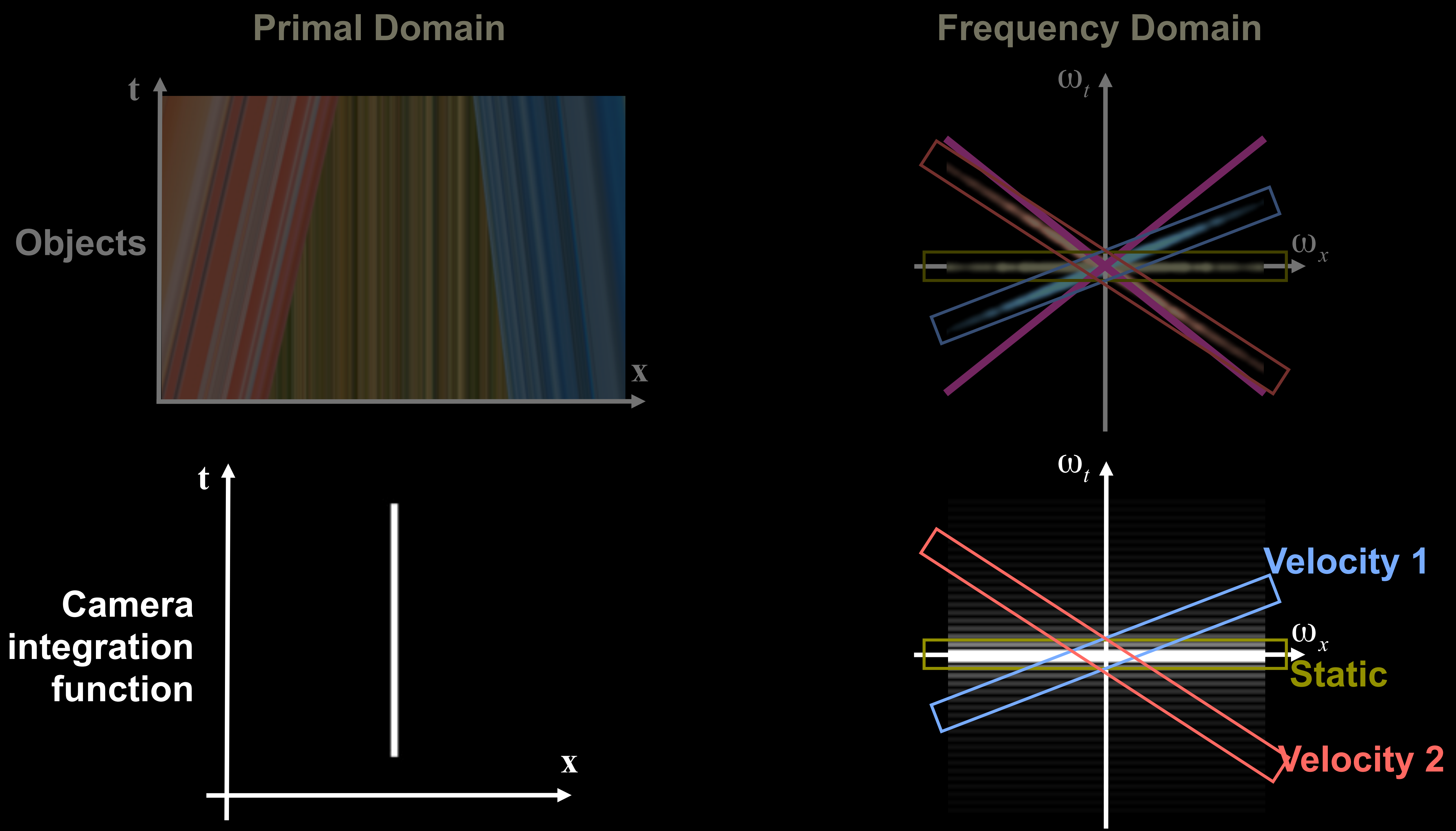
Objects



Frequency Domain



Camera integration => motion blur



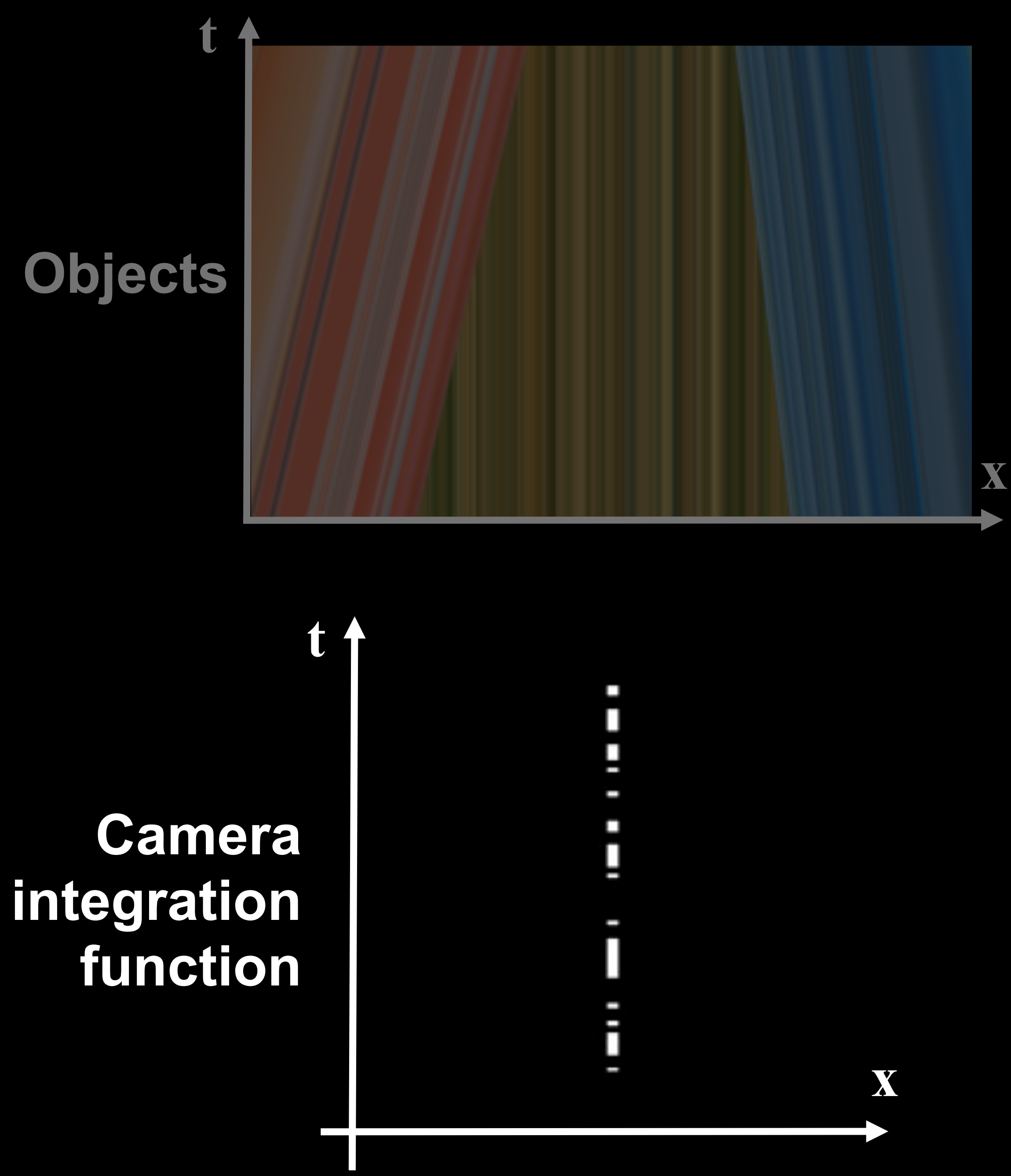
Vertical integration segment

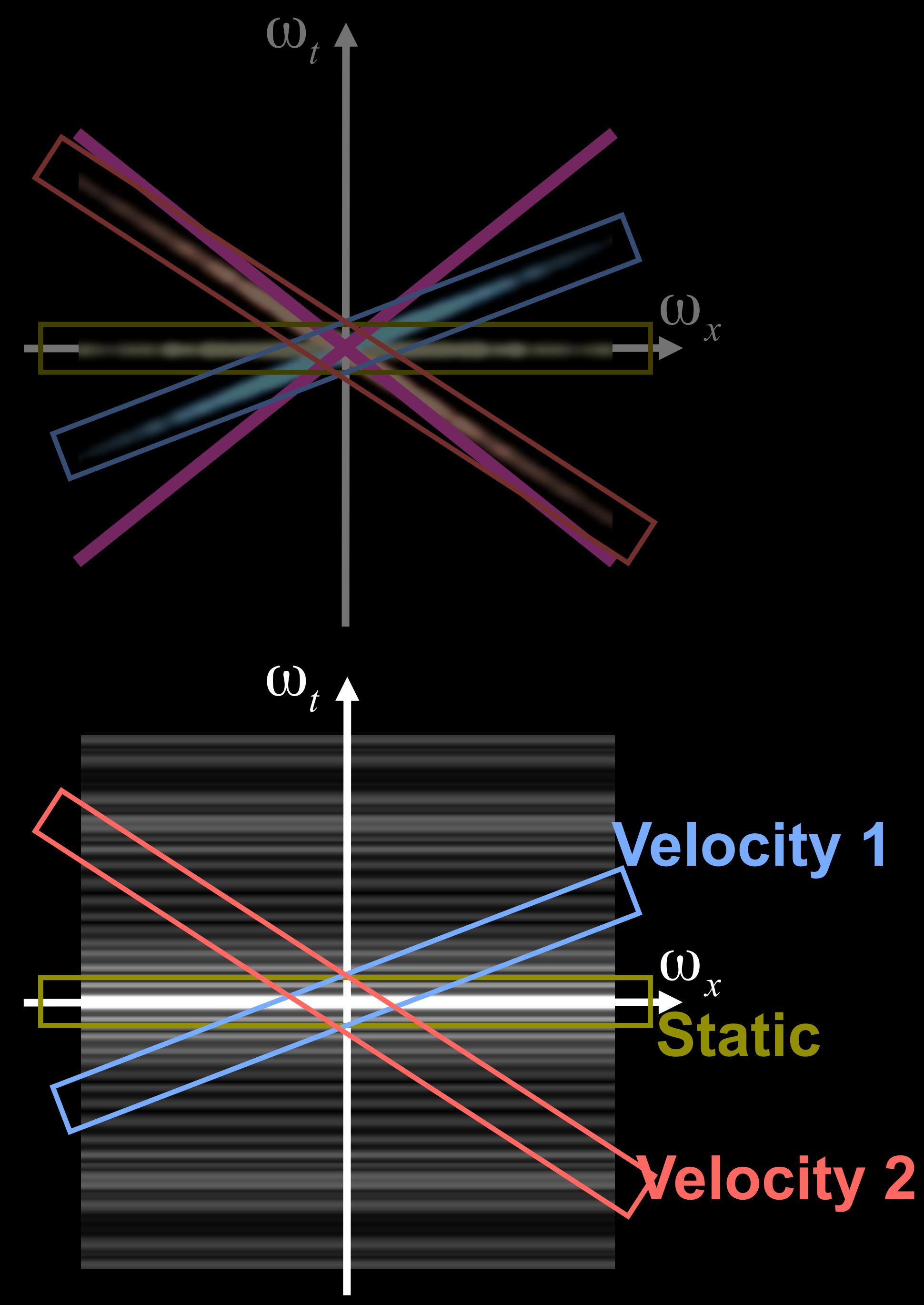
Static object: high response
Higher velocities: low

Flutter shutter (Raskar et al 2006)

Primal Domain

Frequency Domain

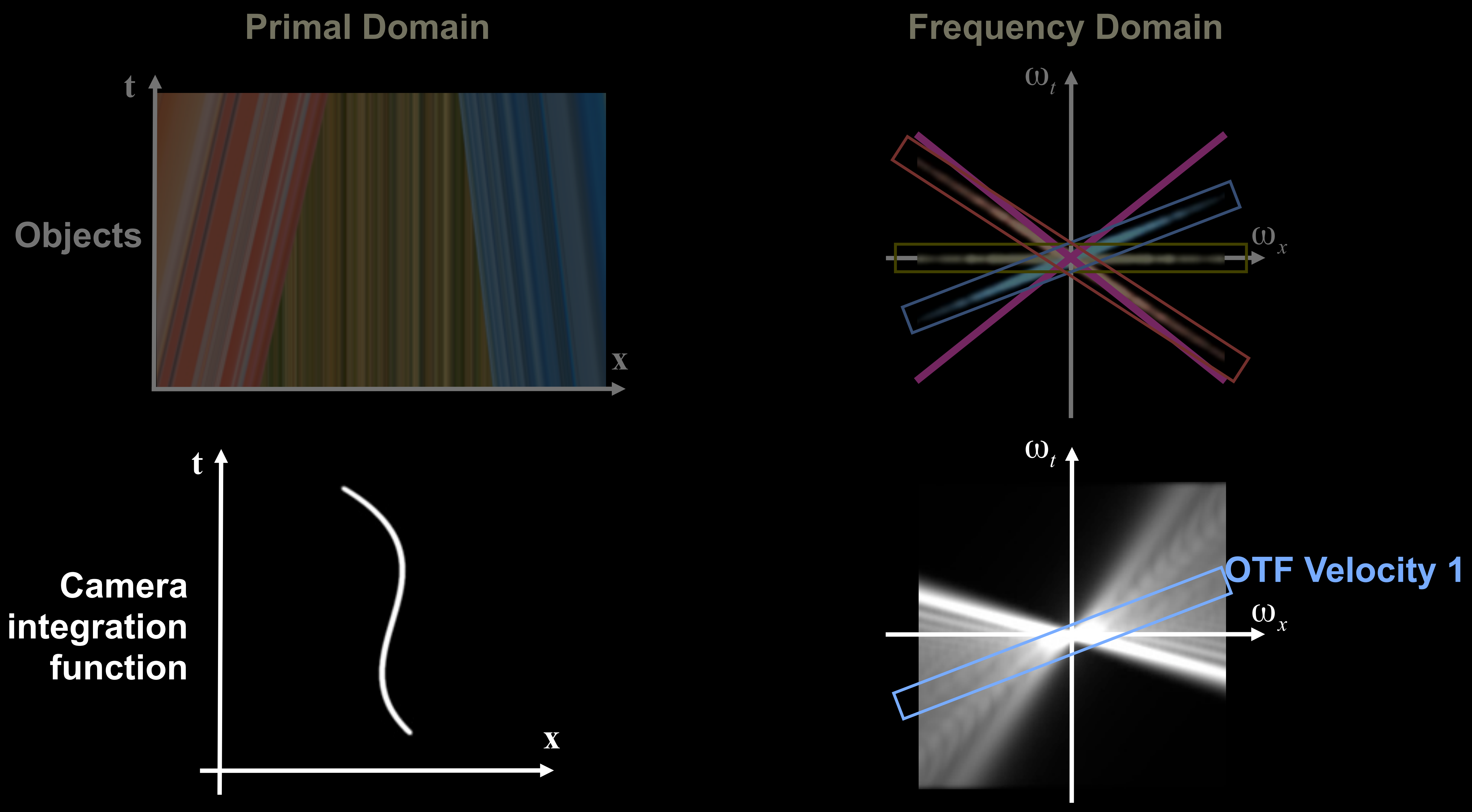




Vertical but discontinuous integration segment

Higher velocities: better than static camera

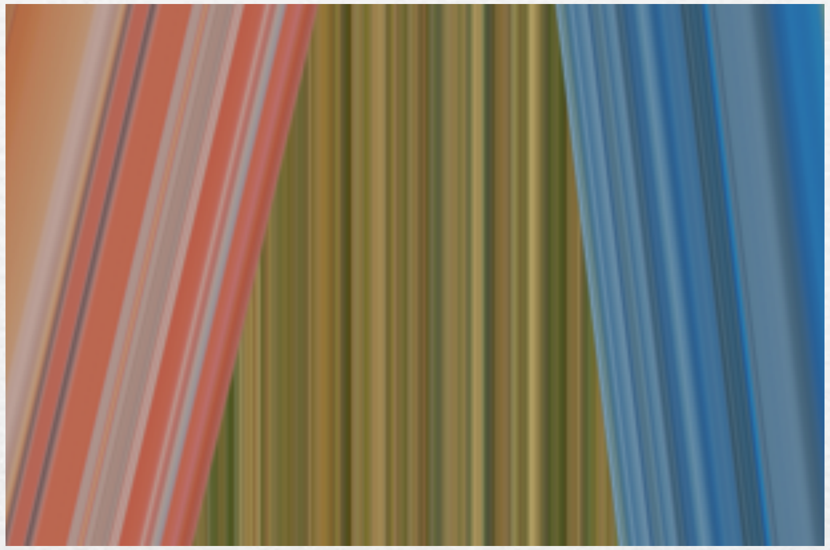
General camera with moving sensor

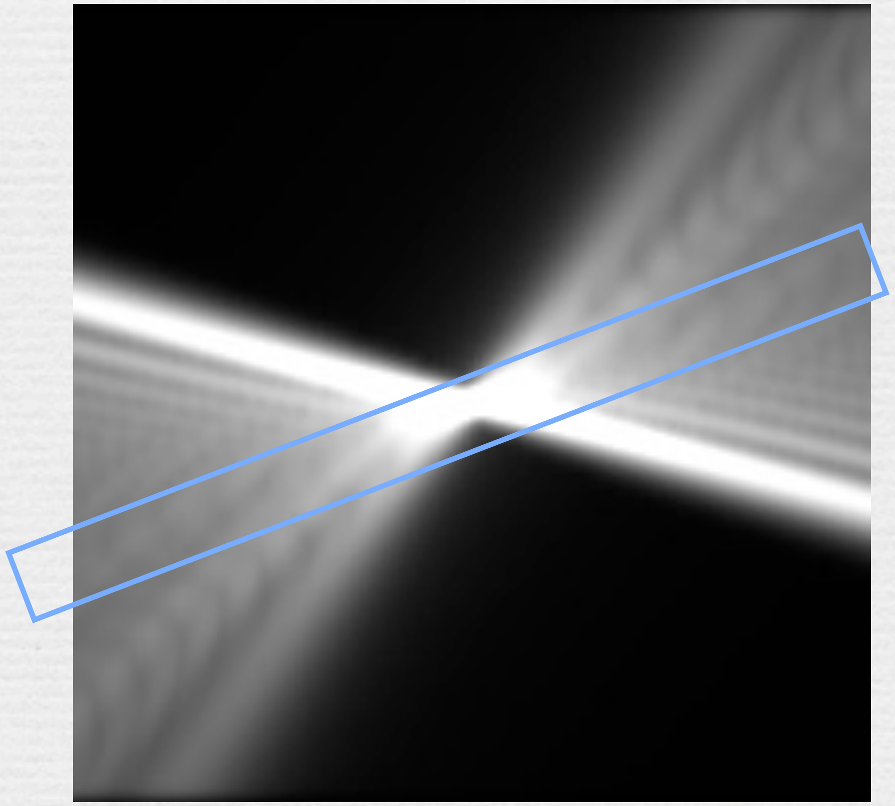


The OTF for given velocity is a slice of the Fourier transform of the camera integration function

Recap

- 3D space time
- Motion = shear,
  slope = 1/velocity
- OTF for given velocity is a slice of Fourier transform of camera integration function





Motion Invariant Photography

with Levin, Cho, Sand & Freeman, Siggraph 08

- Goals:
  - Make blur invariant to velocity
  - Maximize MTF for a range of velocities
- Inspired by wavefront coding [Cathey & Dowski]
- Caveat: works only for 1D motion (known direction, arbitrary velocity)

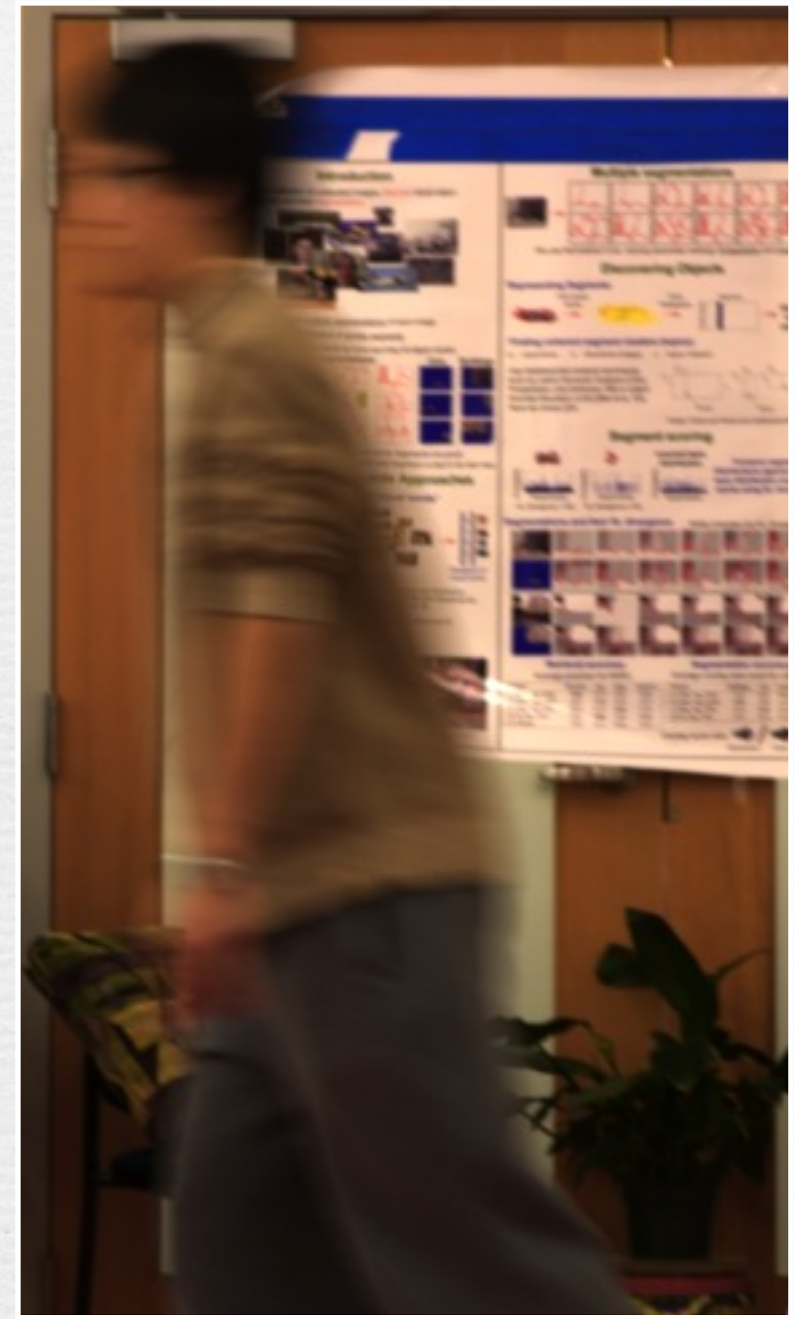
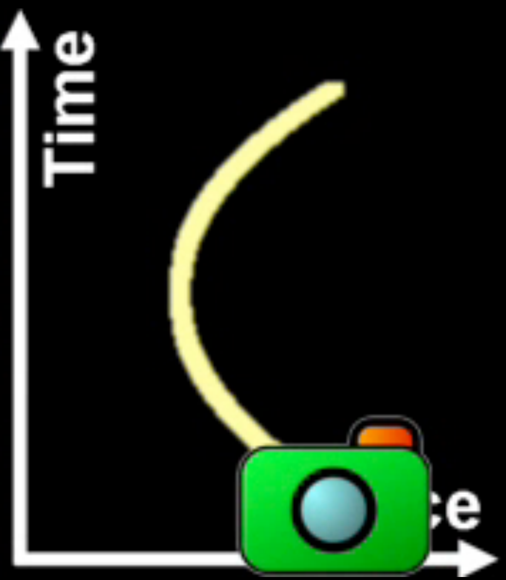
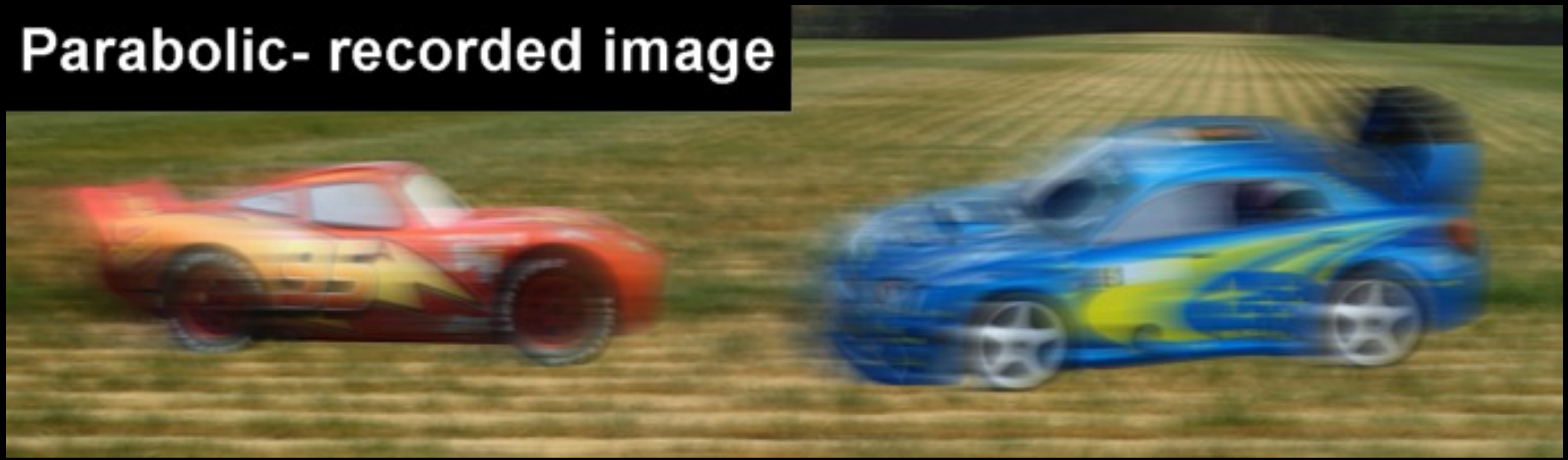


Image with motion blur. Instead, we want all objects to be sharp.

Parabolic sensor motion



Motion invariant blur

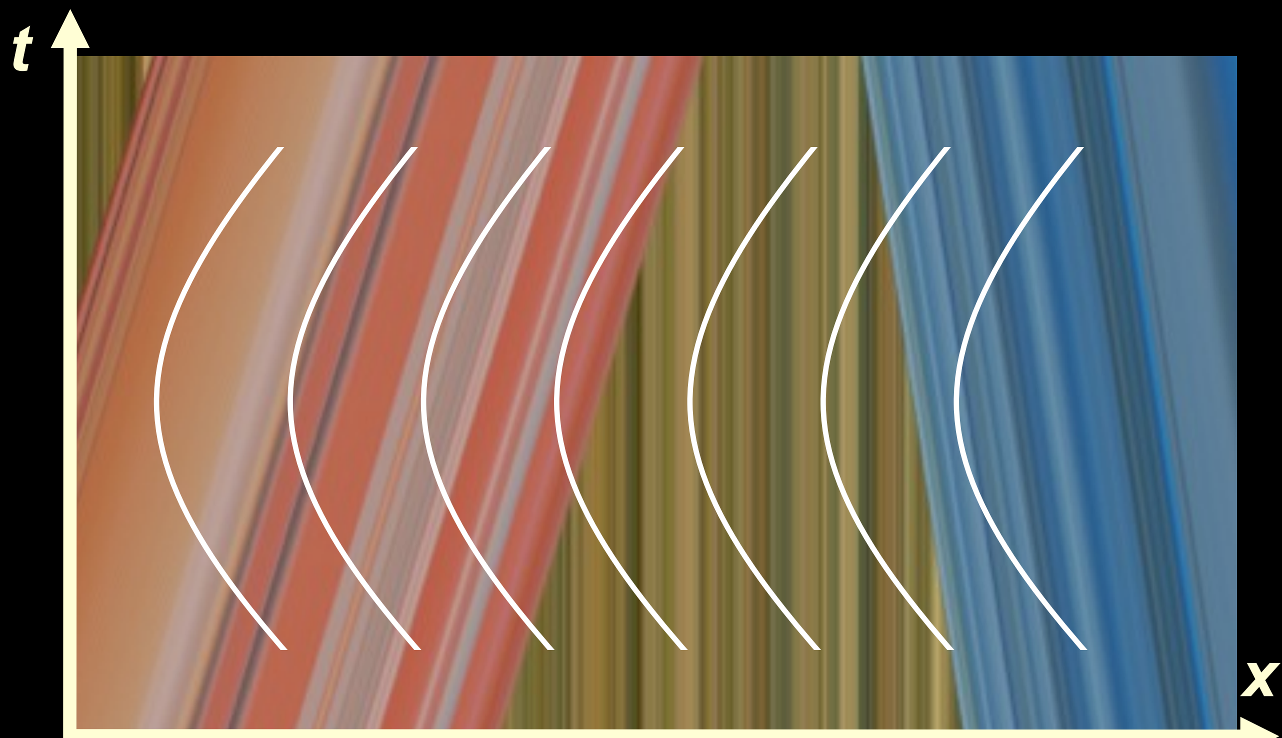


Motion invariant blur

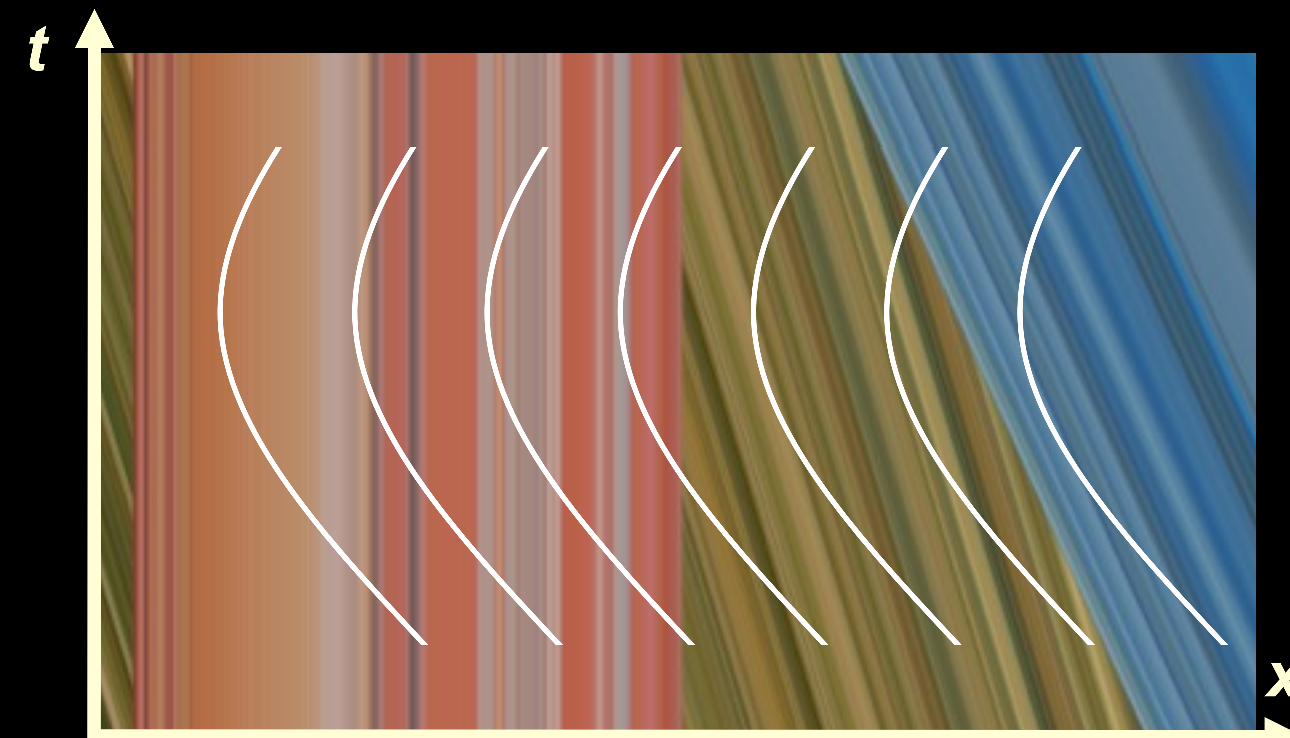
Property 1: parabolic curve is shear invariant

This means it's invariant to velocity changes

Static object coordinates



Moving object coordinates



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

Sheared parabola ➡ Shifted parabola

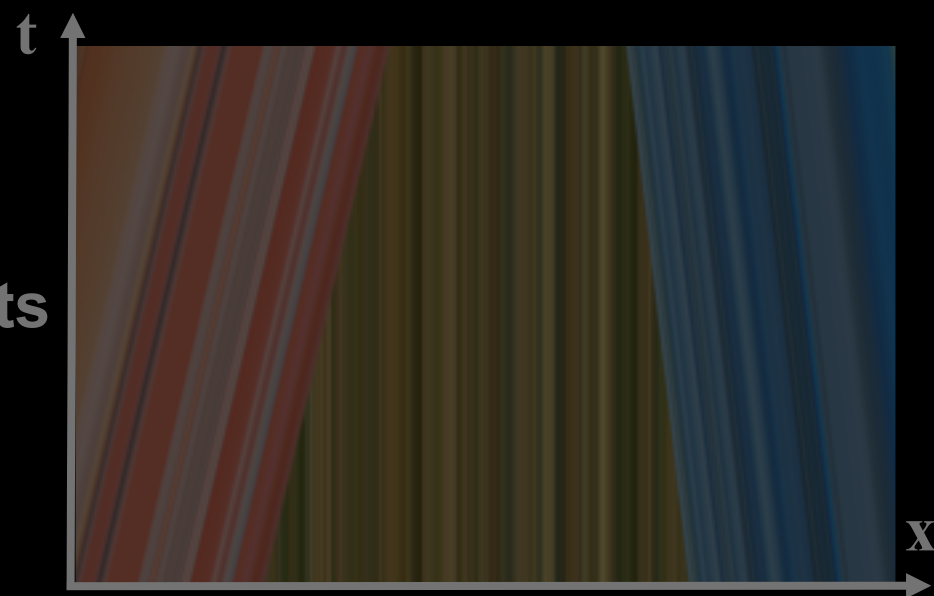
Only curve with this property

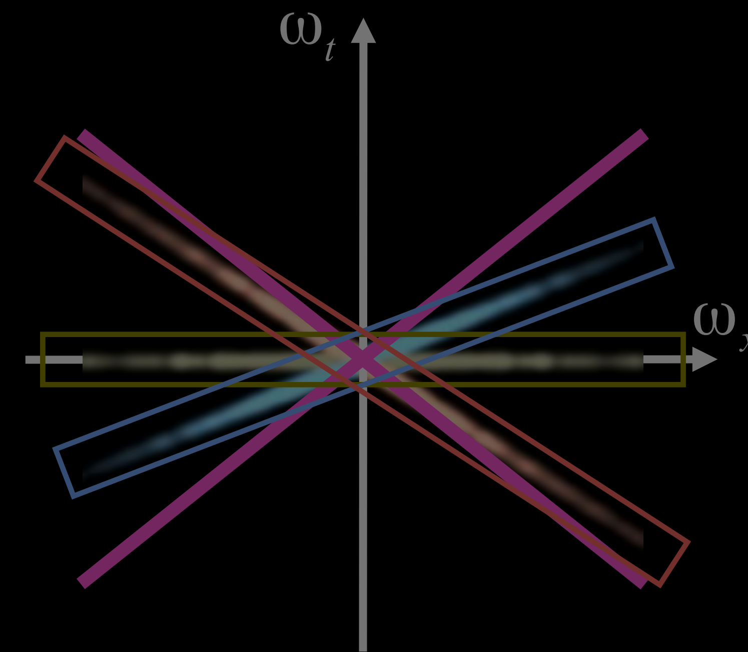
Property 2: parabola yields uniform MTF

Primal Domain

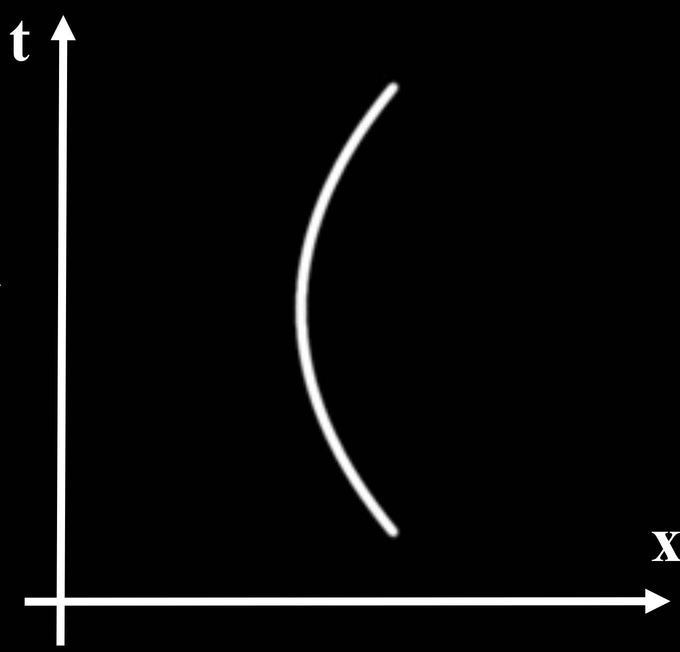
Frequency Domain

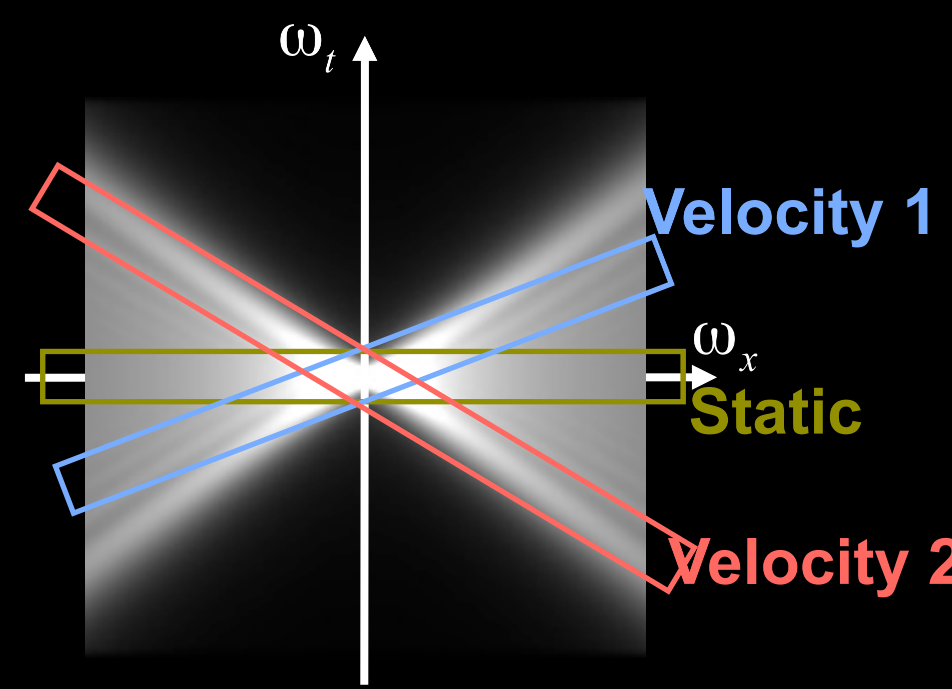
Objects





Camera integration function





Parabola

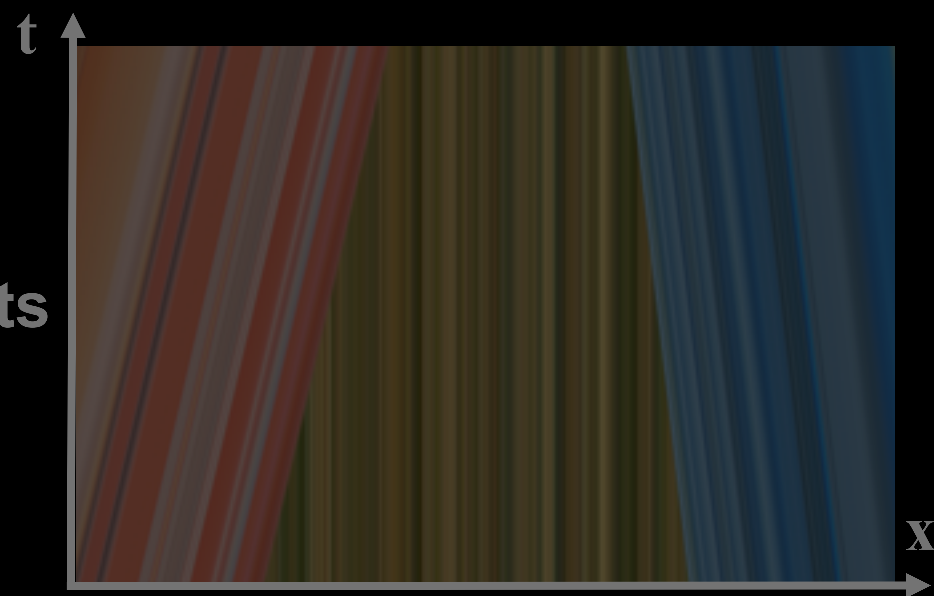
Equal high response in all range

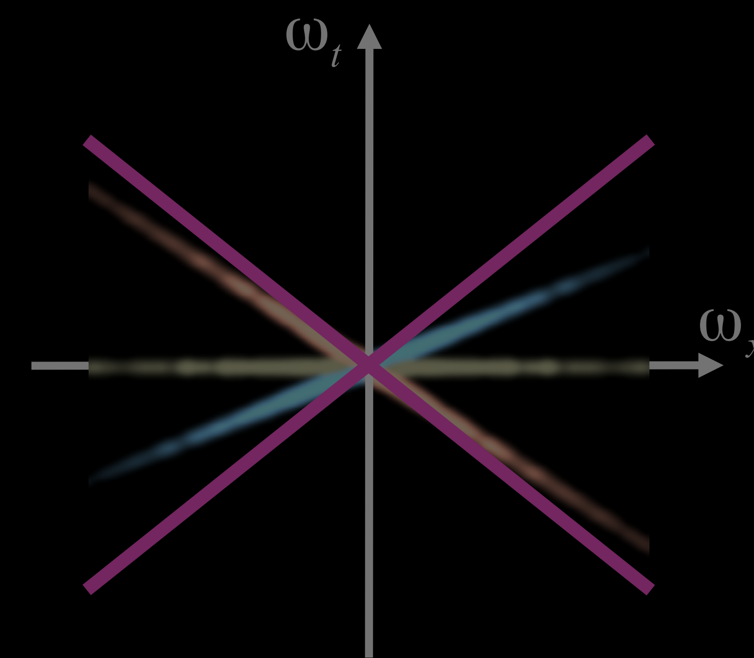
Information budget

Primal Domain

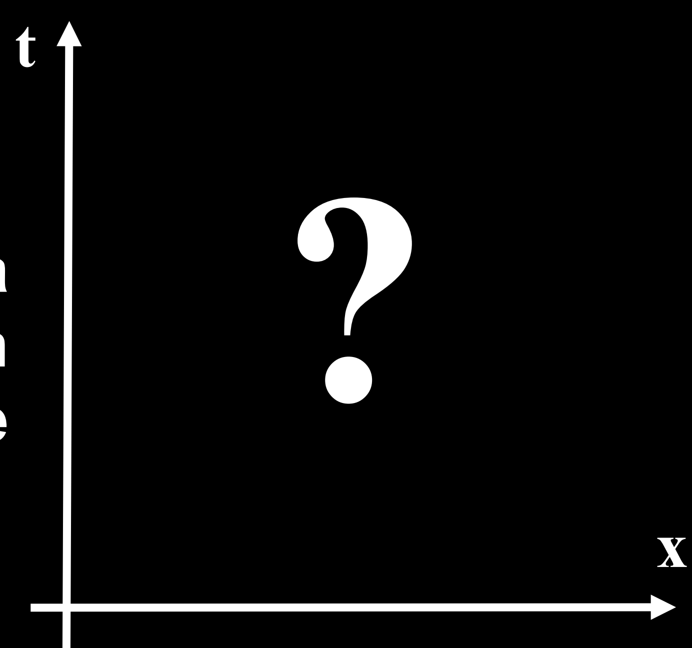
Frequency Domain

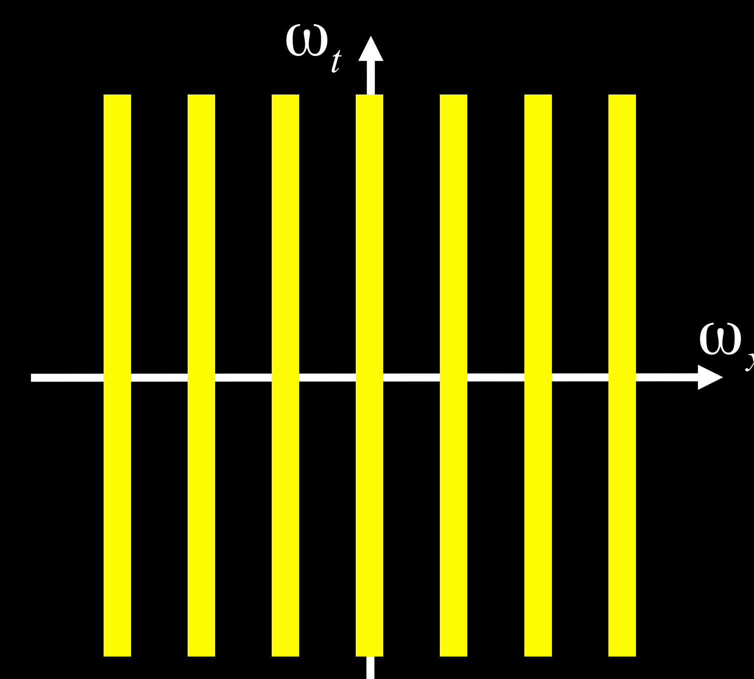
Objects





Camera integration curve





Bounded number of photons

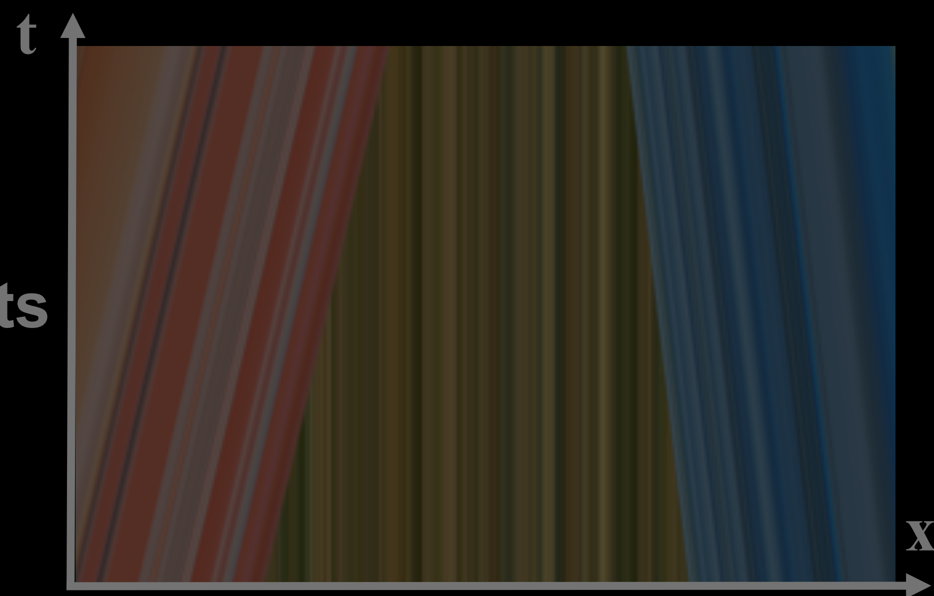
Bounded budget per column (norm of power spectrum)

Upper bound given velocity range

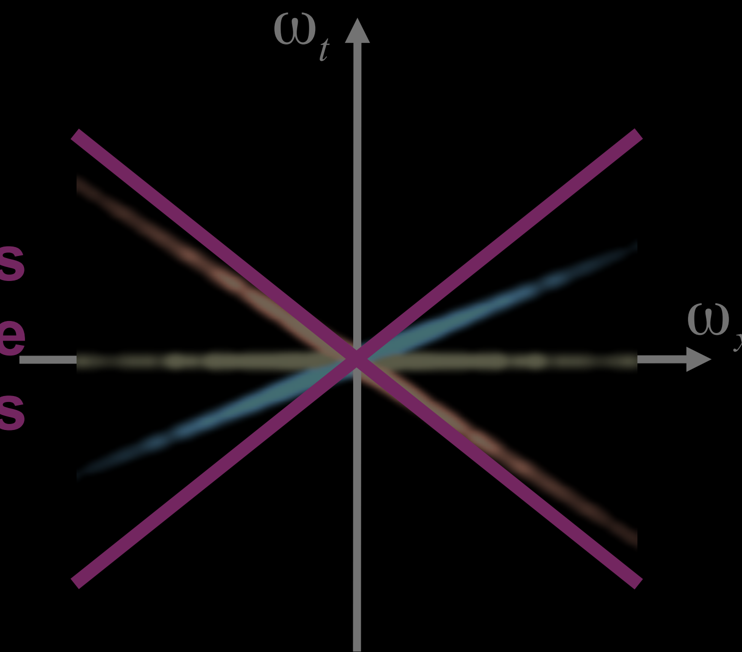
Primal Domain

Frequency Domain

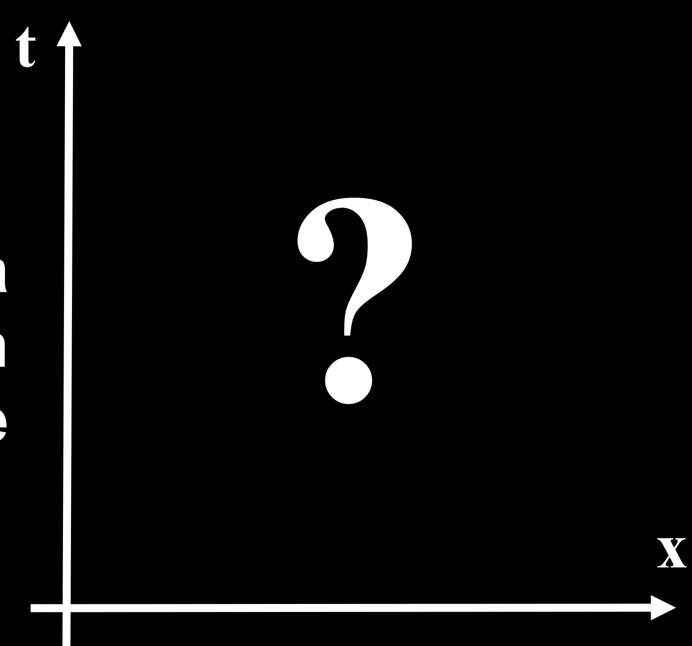
Objects

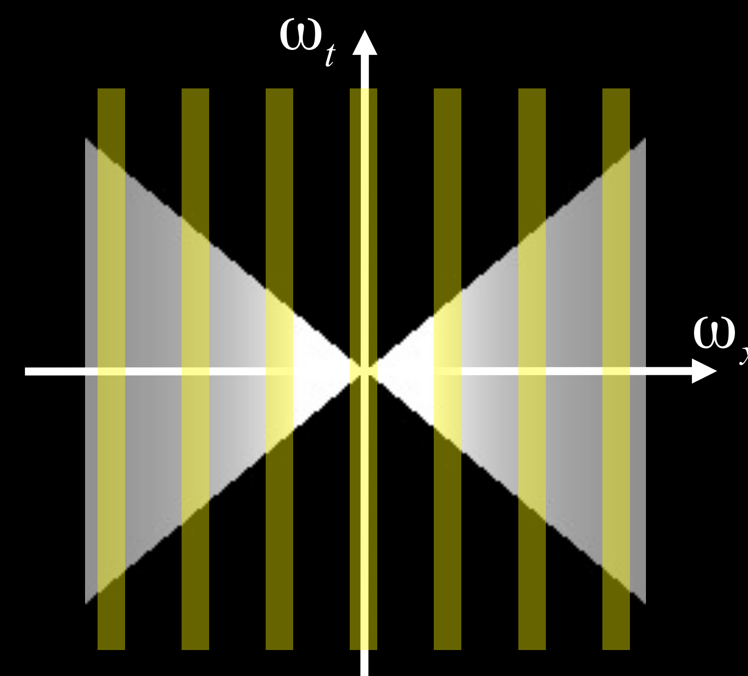


Frequencies from possible motions



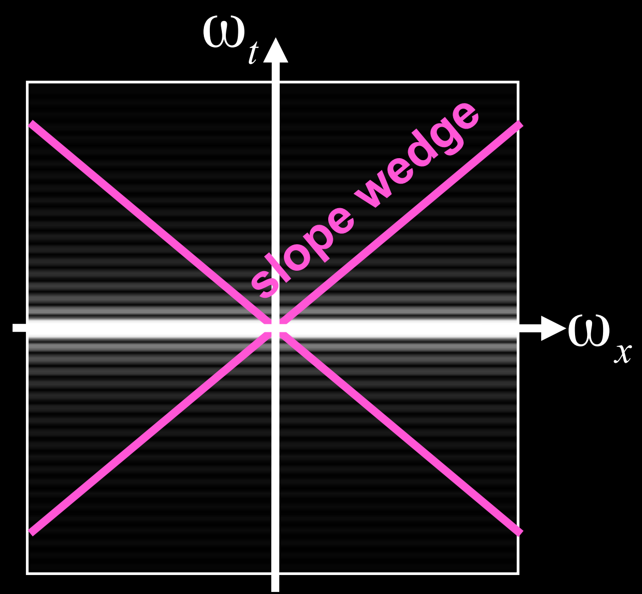
Camera integration curve



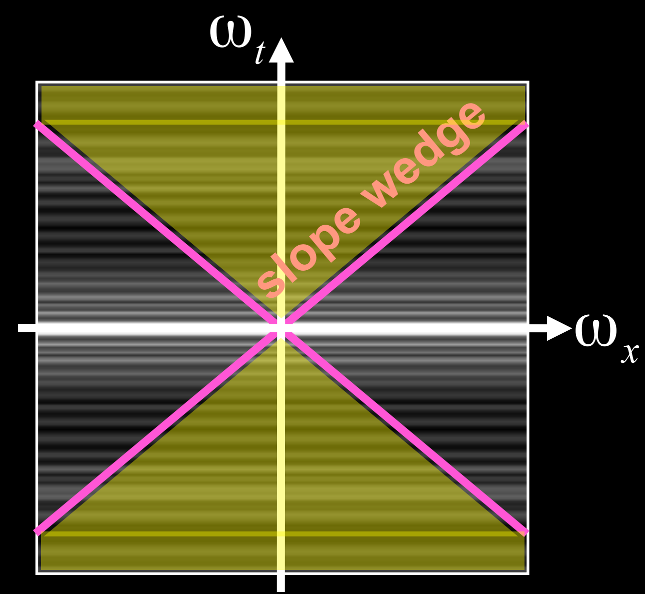


For each column, distribute budget uniformly within wedge $|K(\omega_x,\omega_t)|^2 \le \frac{1}{|\omega_x|}$

Cameras and information preservation



Static

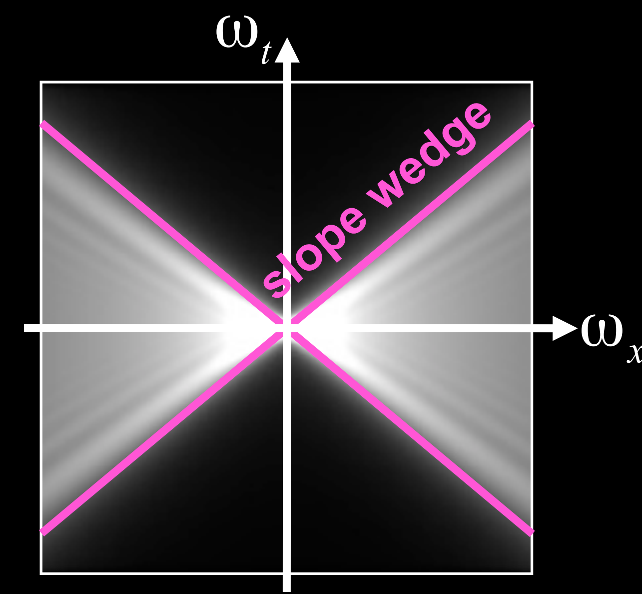


Flutter shutter

Constant horizontally

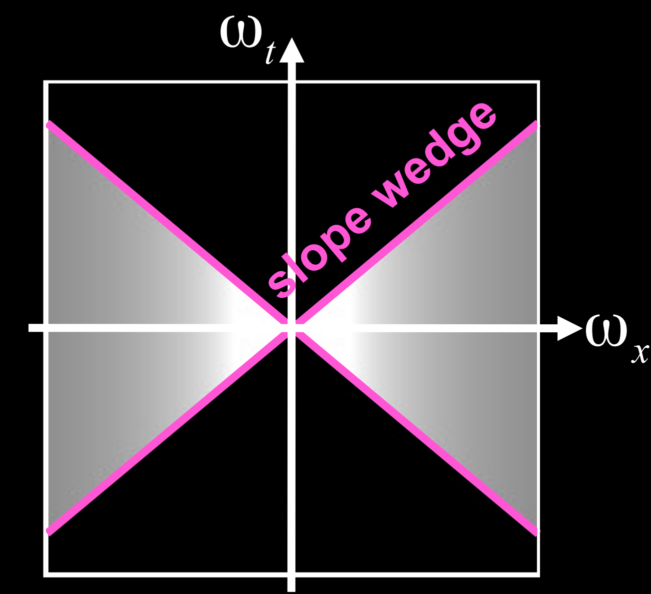
Spends frequency “budget” outside wedge

Handles 2D motion



Parabolic

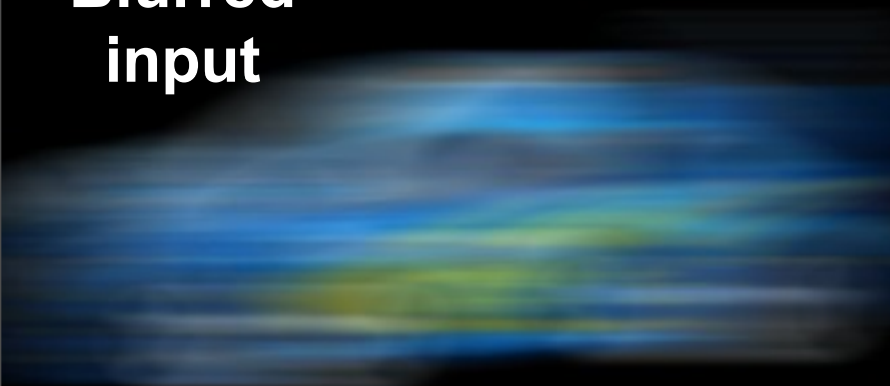
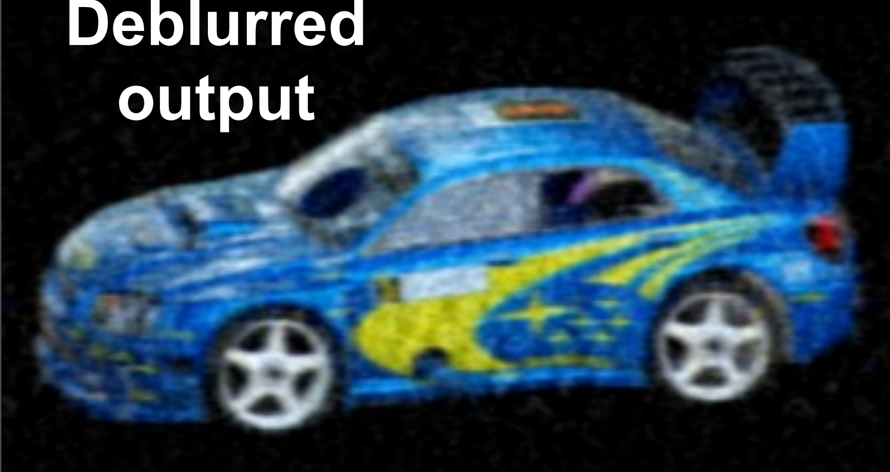
Near optimal “budget” usage at all frequencies



Upper bound

Bounded “budget” per column ω_x

Comparing camera reconstruction

	Static	Flutter Shutter	Parabolic
Blurred input			
Deblurred output			

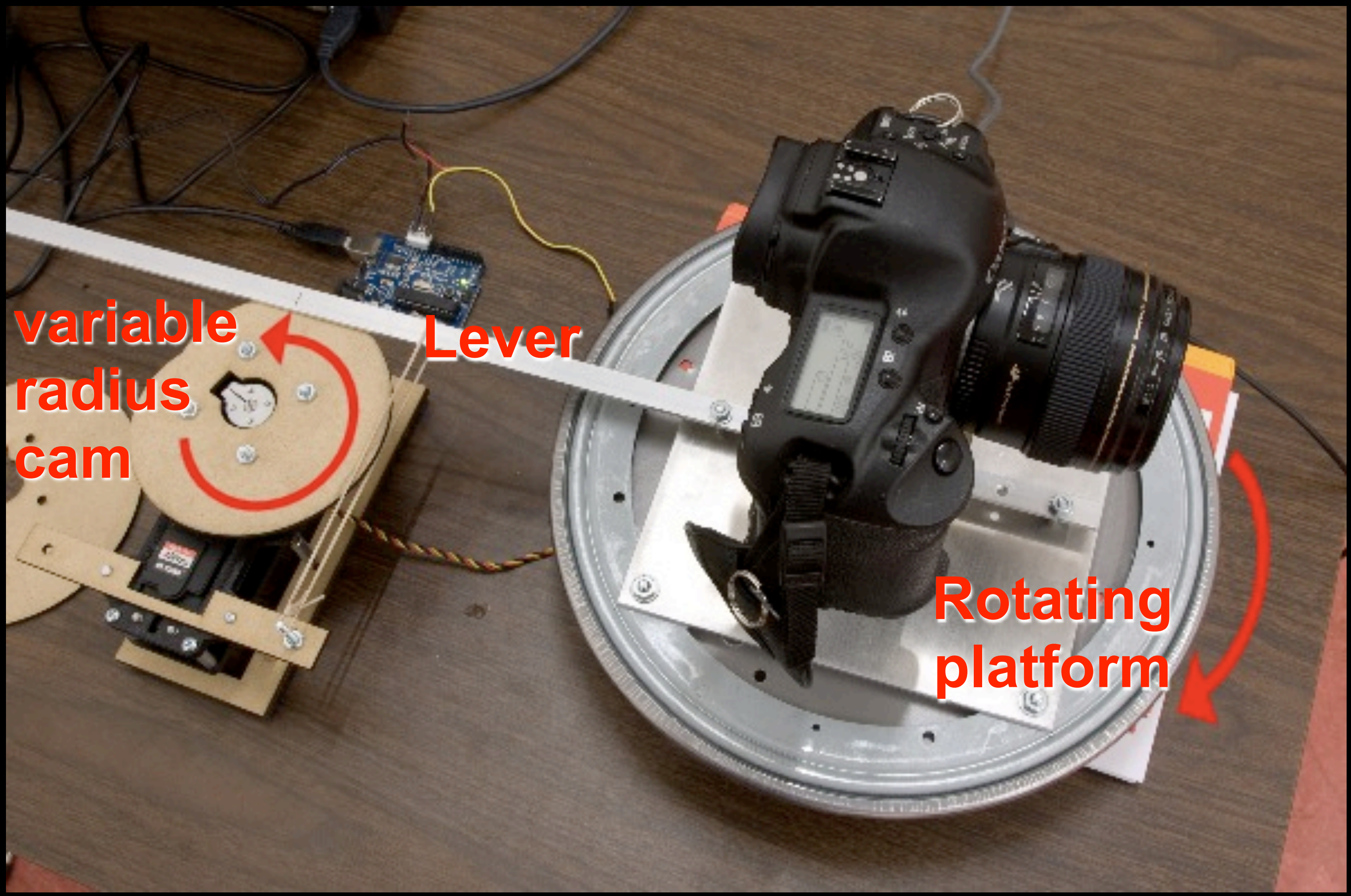
Note: synthetic rendering, exact PSF is known

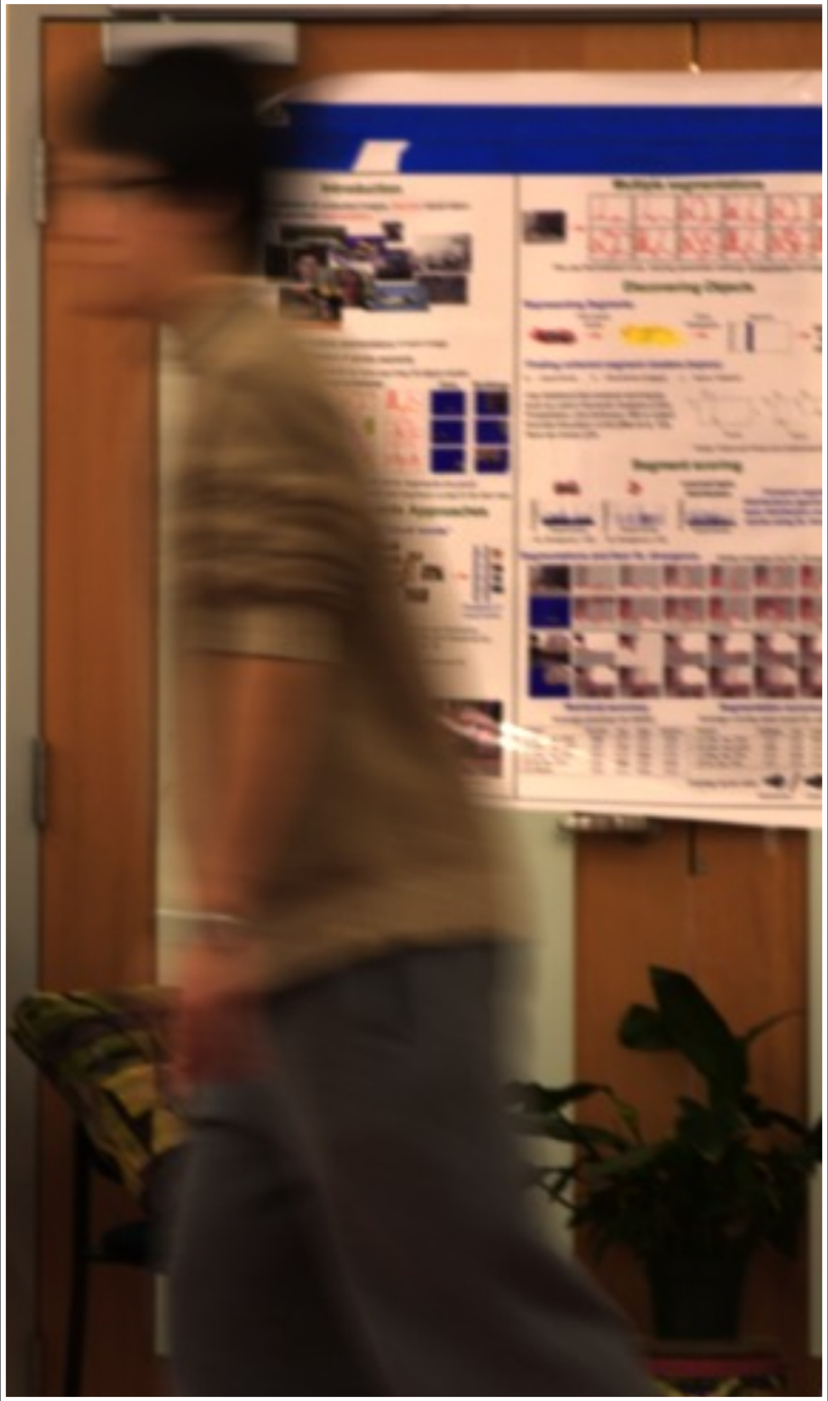
Hardware construction

- **Ideally move sensor**

  (requires same hardware as existing stabilization systems)

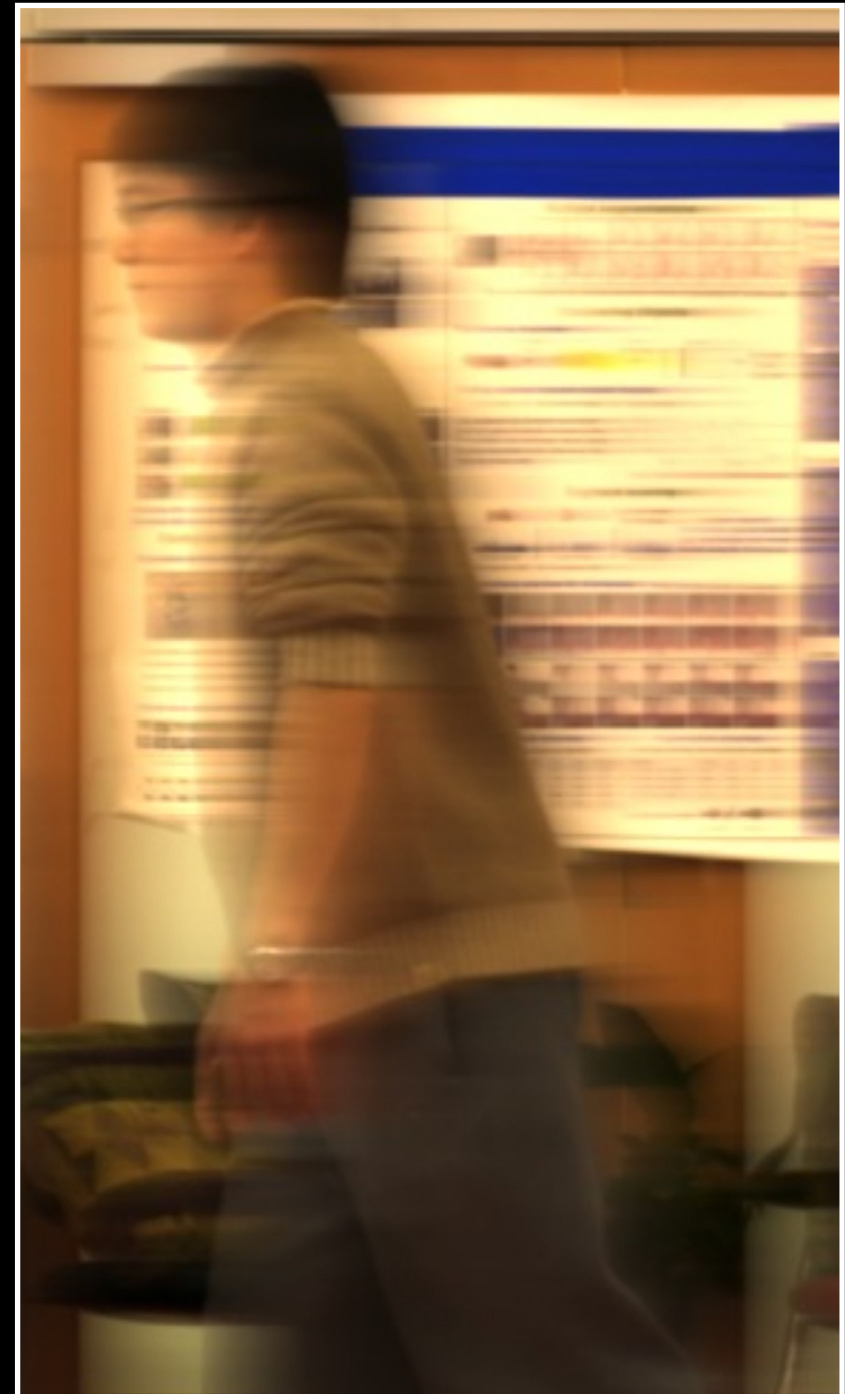
- **In prototype implementation: rotate camera**





Static camera

Unknown and variable blur kernels



Our parabolic input

Blur kernel is invariant to velocity

Our output after deblurring

NON-BLIND deconvolution

Human motion– no perfect linearity

Input from a static camera

Deblurred output from our camera

Application #2

with Egan, Tseng, Holzschuch & Ramamoorthi

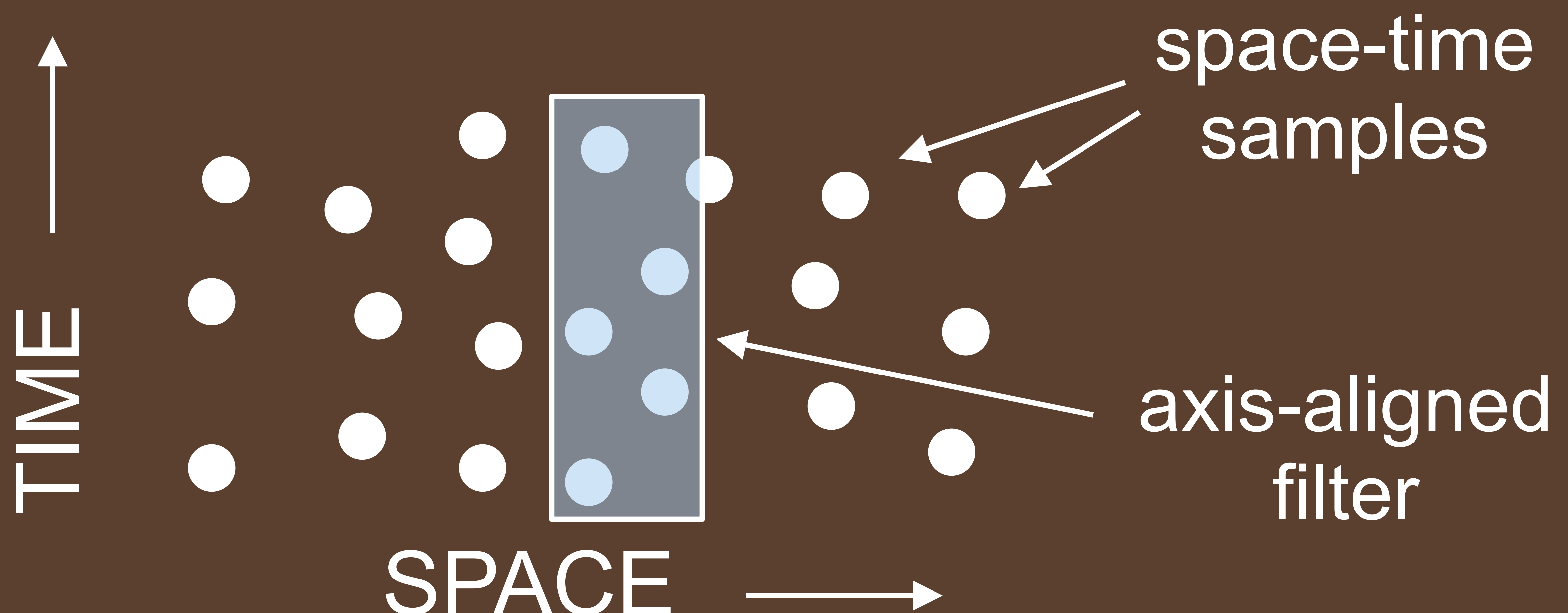
- Fast image synthesis
- Goal: reduce noise in Monte-Carlo simulation of motion blur

Observation

- Motion blur is expensive
- Motion blur *removes* spatial complexity

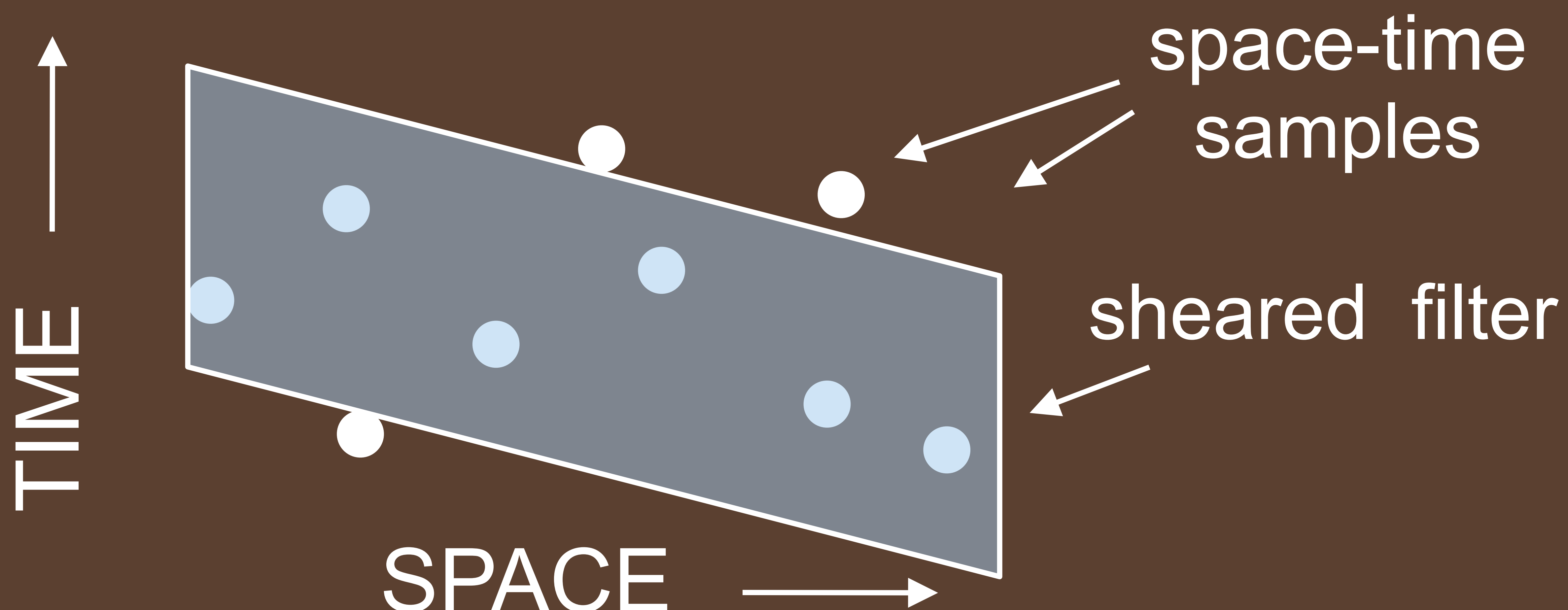
Standard Method

- Use axis-aligned pixel filter at each pixel
  - space => antialiasing
  - time => motion blur
- Requires many samples



Our Method

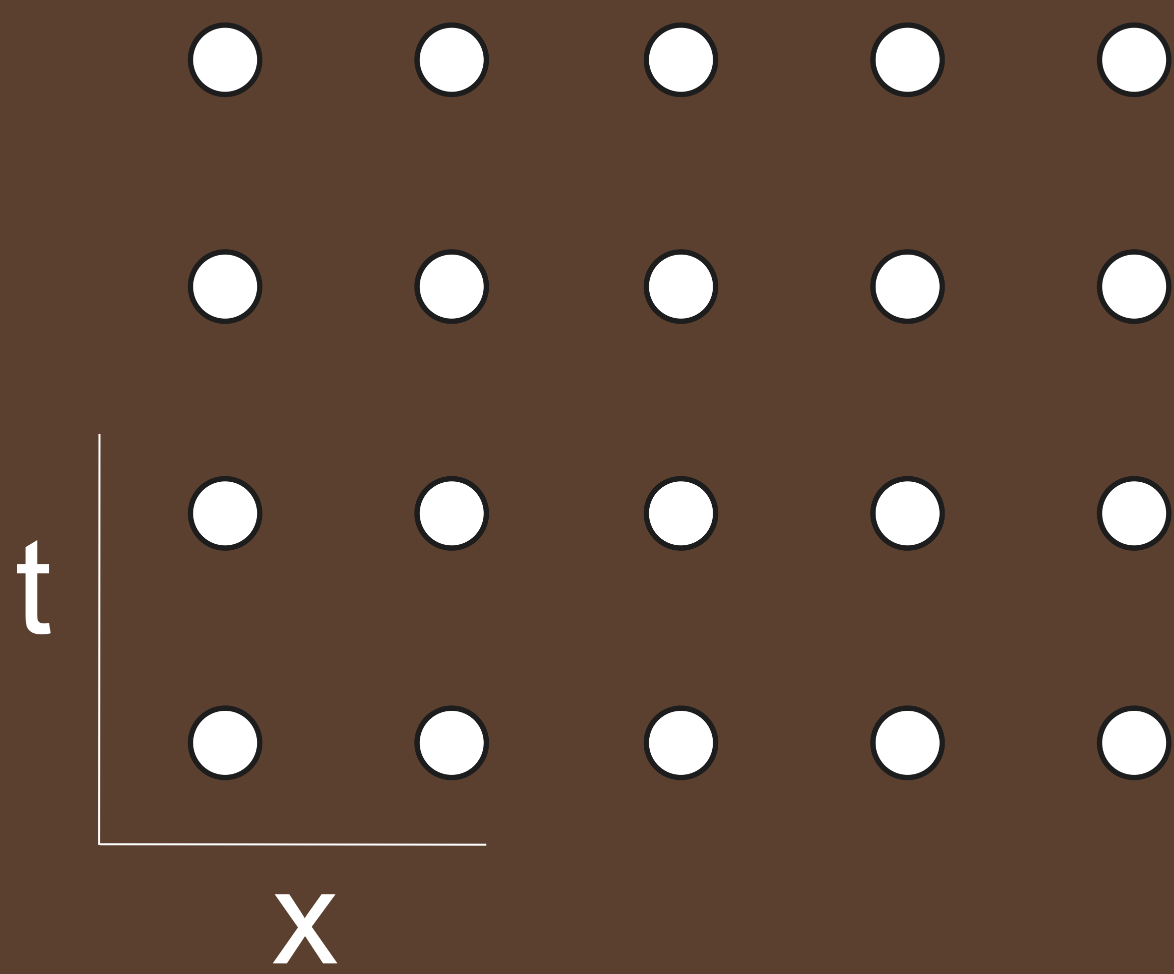
- Use a different filter shape at each pixel
- Filter sheared in space-time
- Fewer samples and faster renders



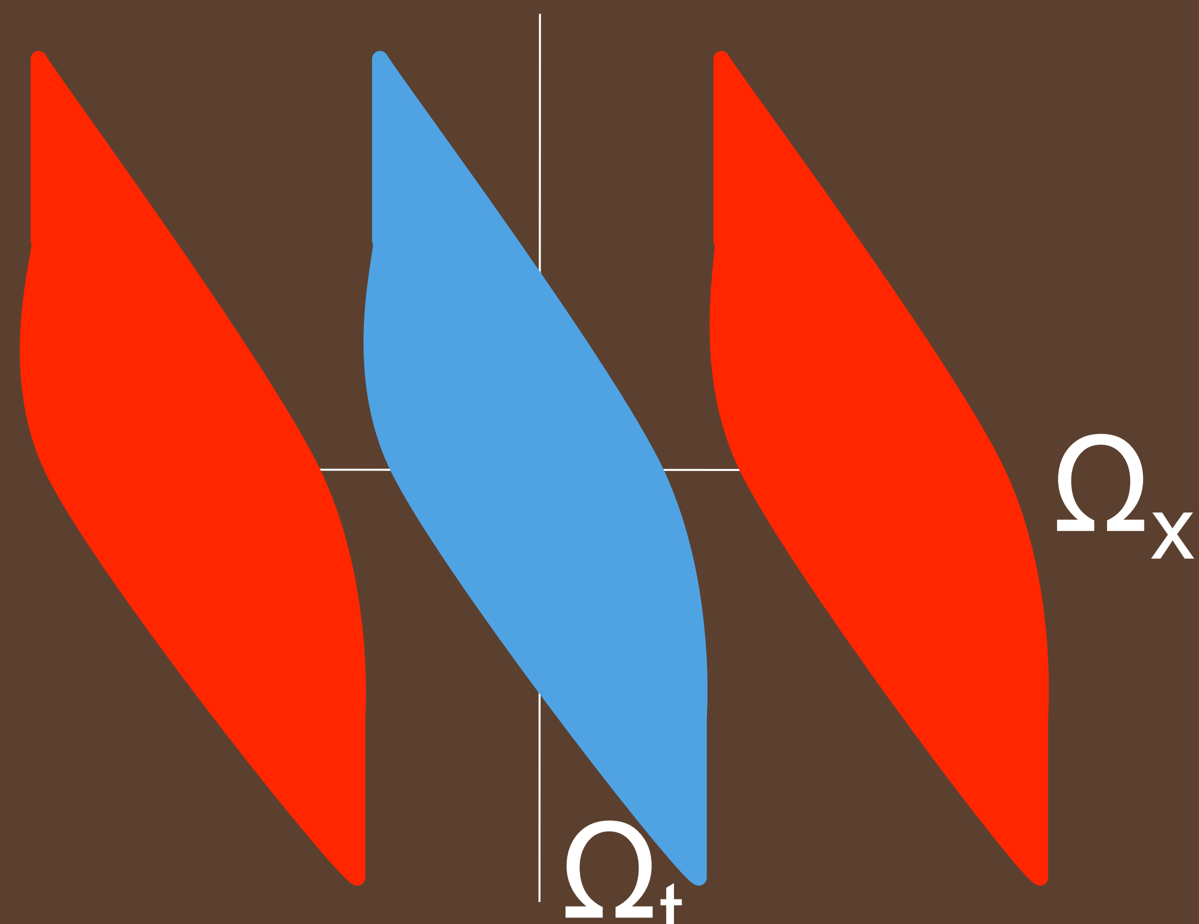
Sampling in Fourier Domain

- Sampling produces replicas in Fourier domain
- Sparse sampling produces dense replicas

Space-Time Domain



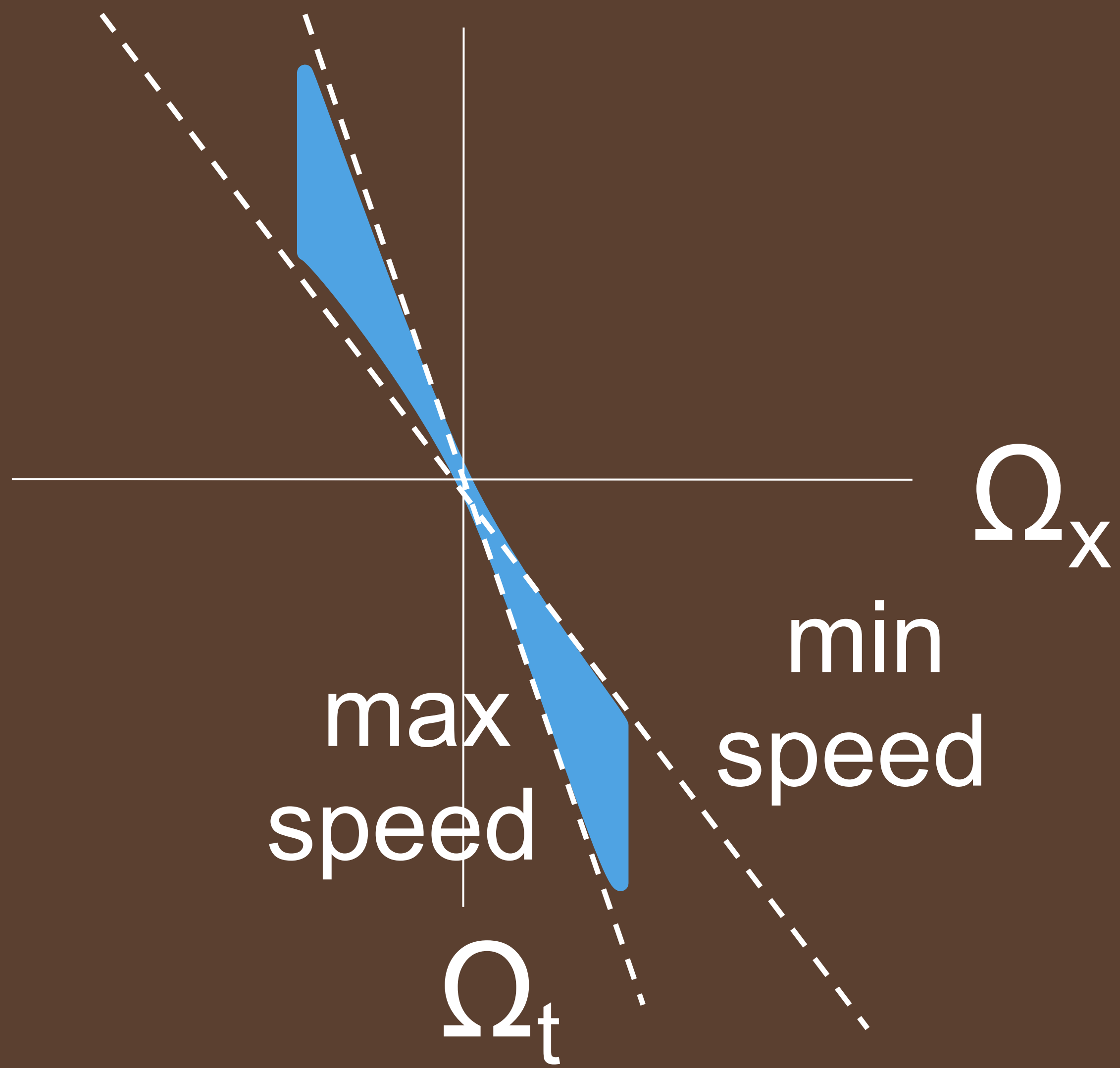
Fourier Domain



But spectra have wedge shape

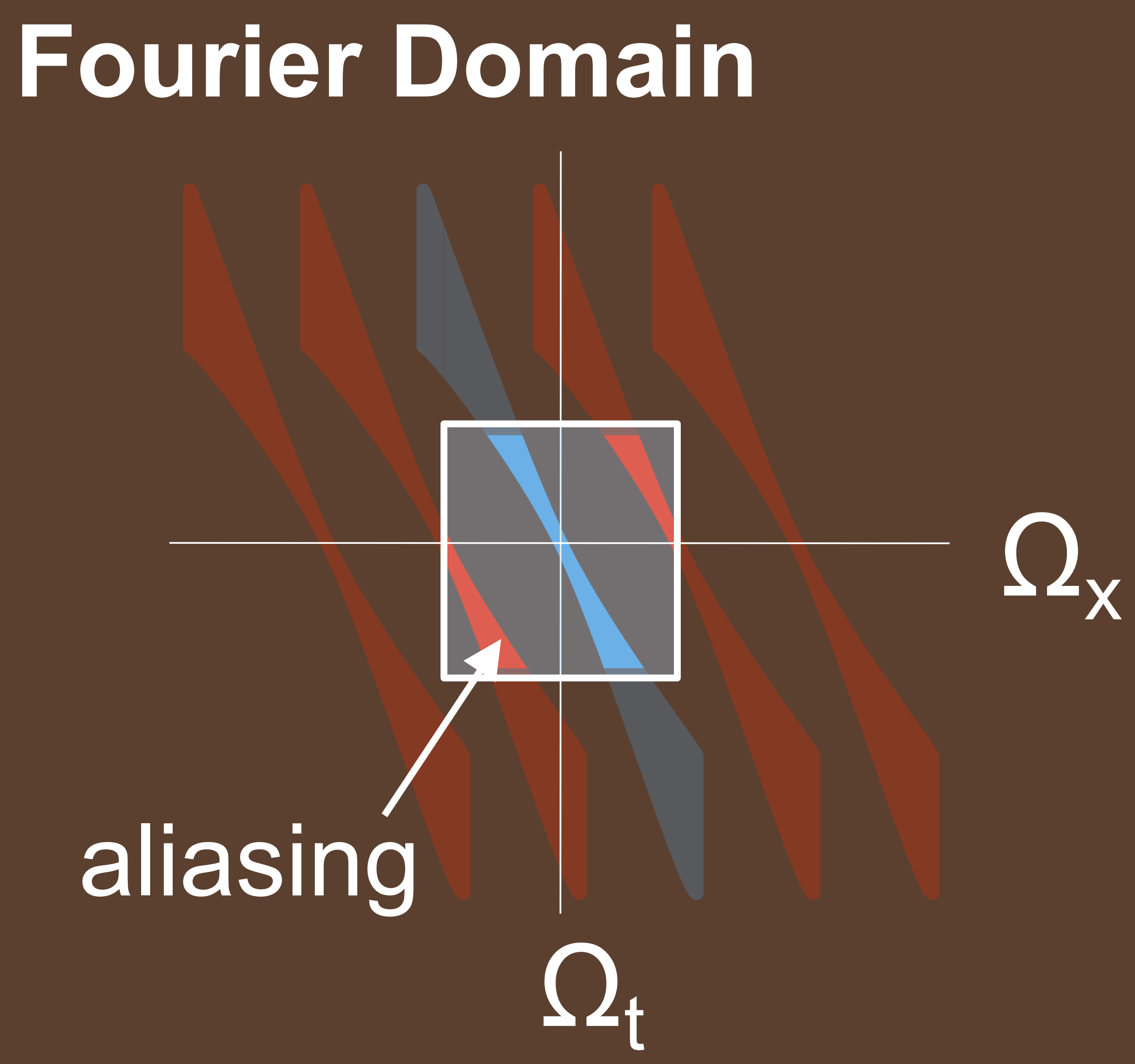
- And we can compute (local) bandwidth based on velocity

Fourier Domain



Standard Reconstruction Filter

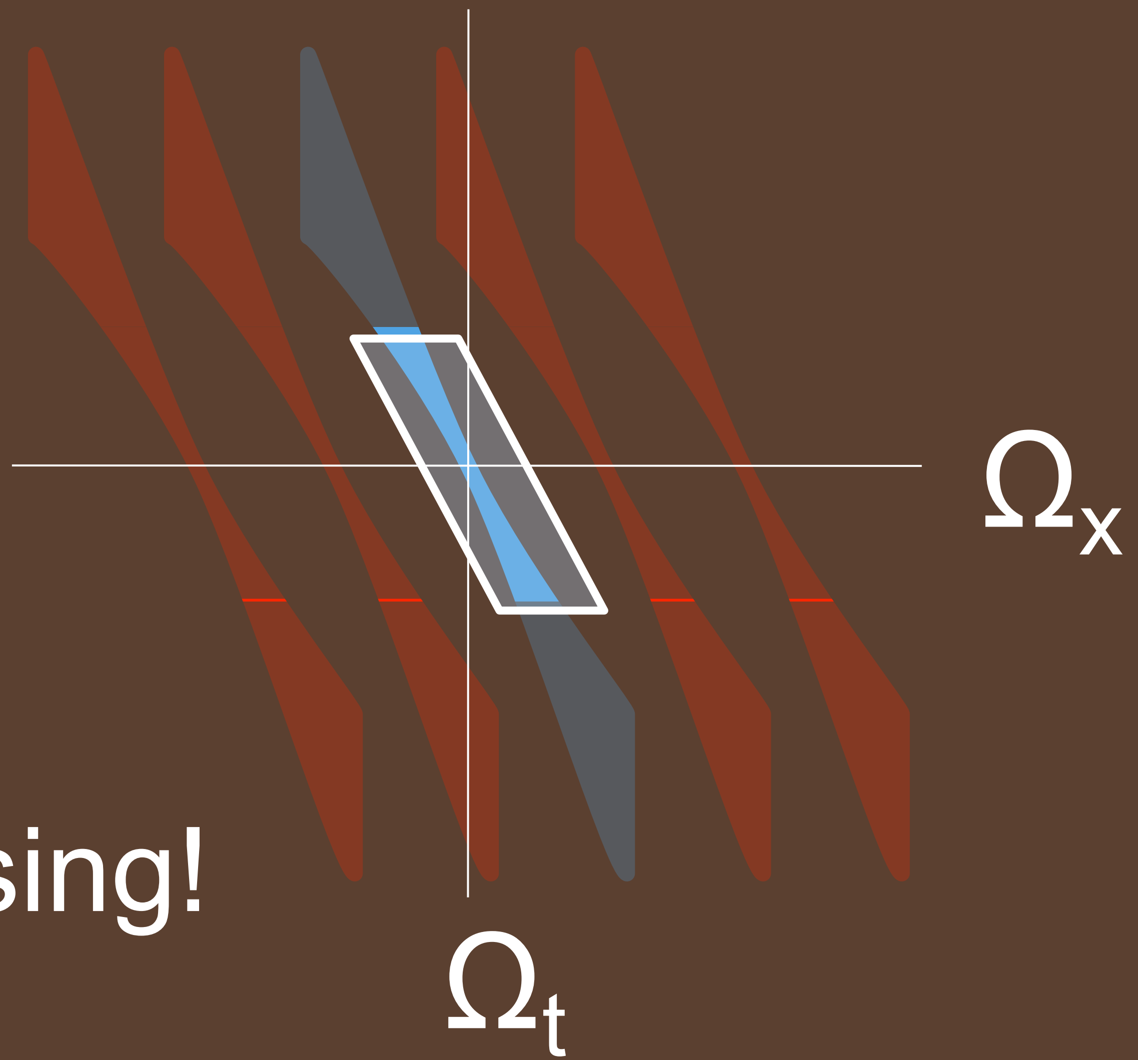
- Aliasing with sparse sampling



Sheared Reconstruction Filter

- No aliasing with sparse sampling
  => no noise in Monte-Carlo integration

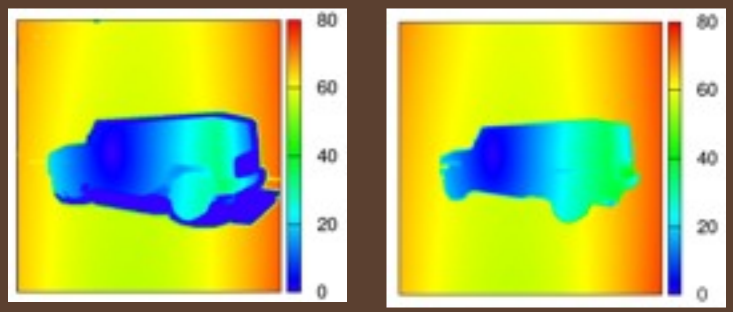
Fourier Domain

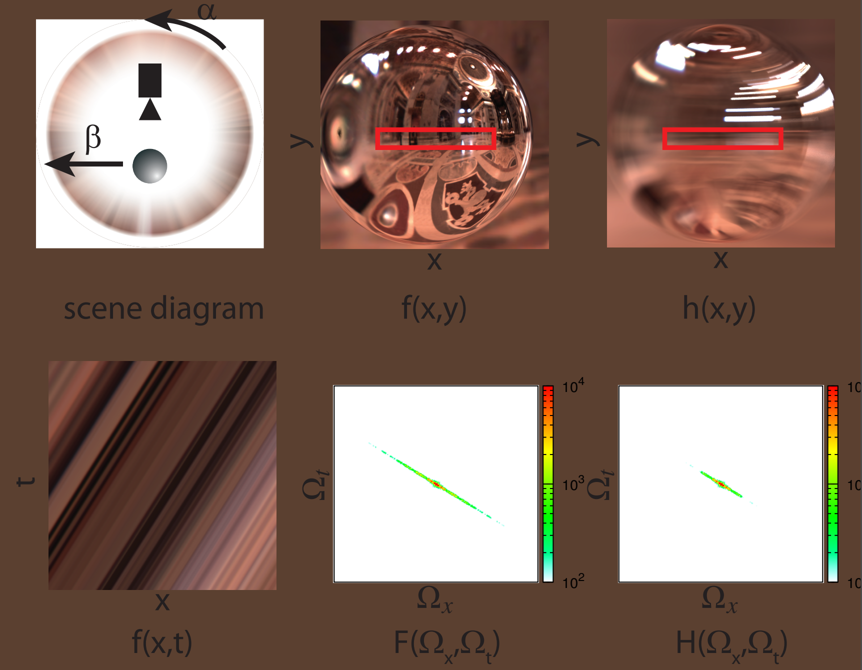


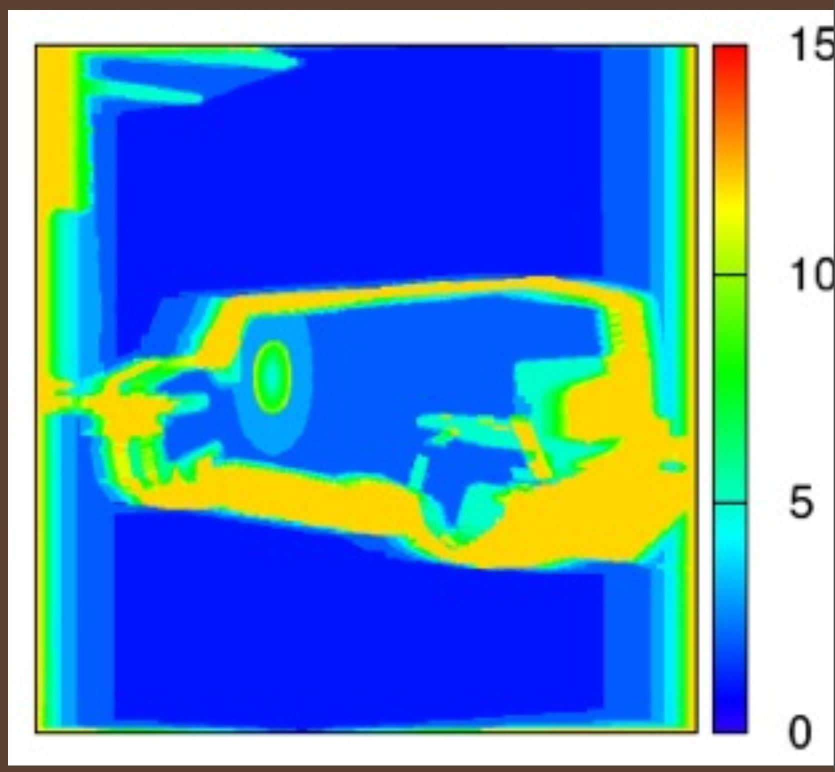
No aliasing!

Making it happen

- Compute per-pixel bandwidth
  - local velocity
  - moving shadows, moving highlights
- Sample based on bandwidth
- Reconstruct final image with sheared filter





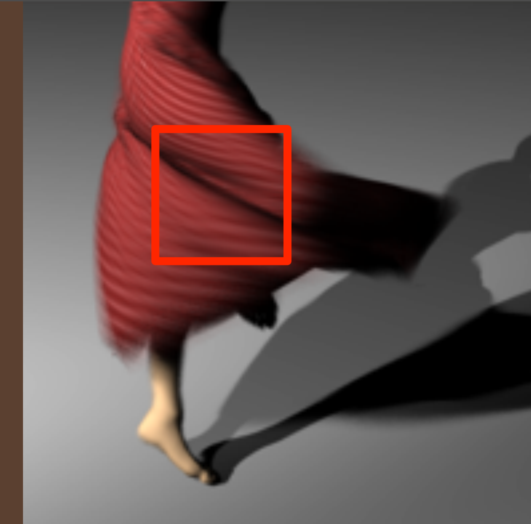


Car Scene

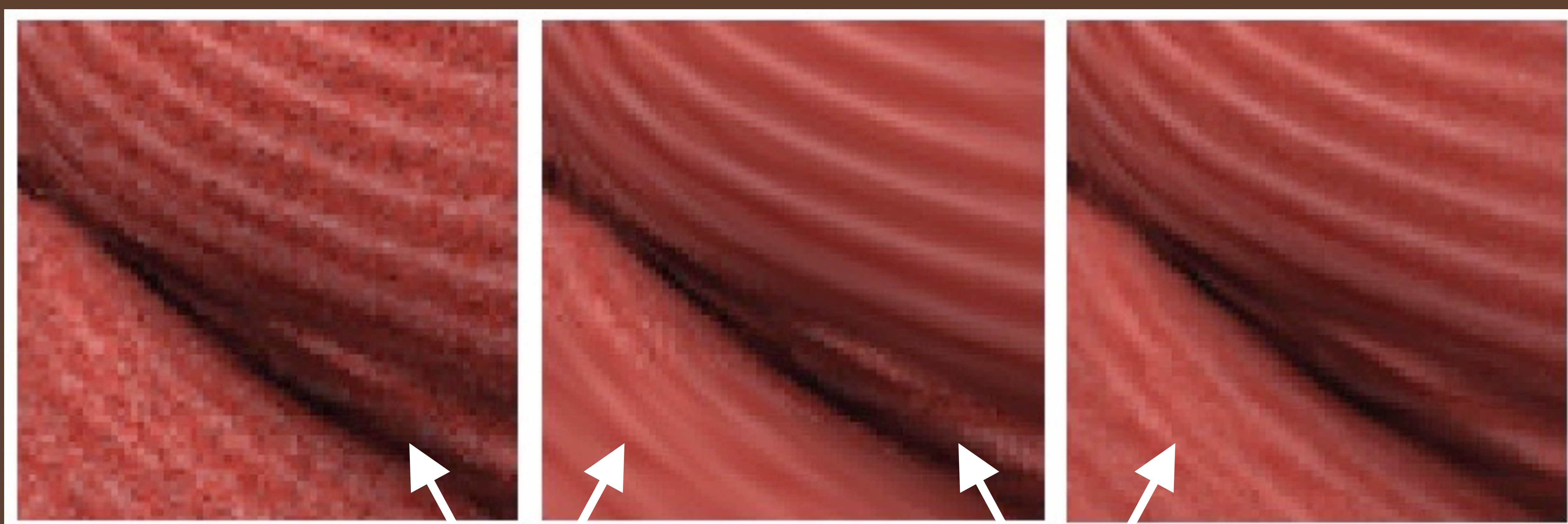
Stratified Sampling, 4 samples per pixel

Our Method, 4 samples per pixel

Ballerina



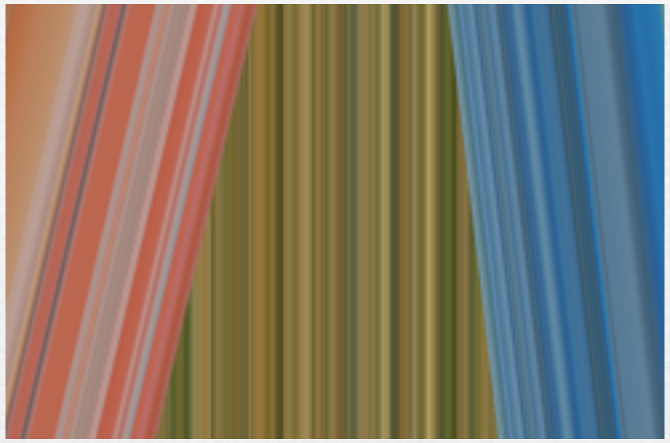
Stratified	Our Method	Stratified
16 samples / pix	8 samples / pix	64 samples / pix
4 min 2 sec	3 min 57 sec	14 min 25 sec

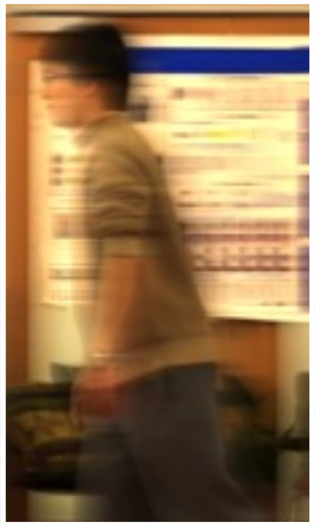


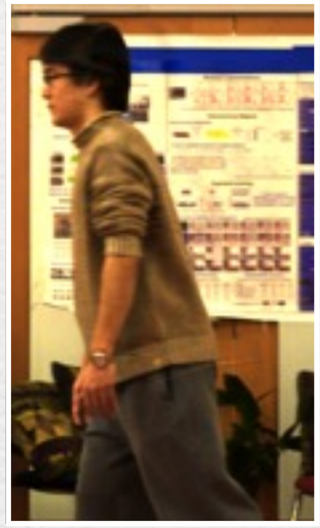
Equal Time **Equal Quality**

Motion blur recap

- Space-time; slope = speed
- OTF = slice of Fourier transform of camera integration function
- Capture
  - shear invariance
  - high & uniform MTF
  - upper bound
- Synthesis
  - sparse sampling
  - sheared reconstruction => sparser

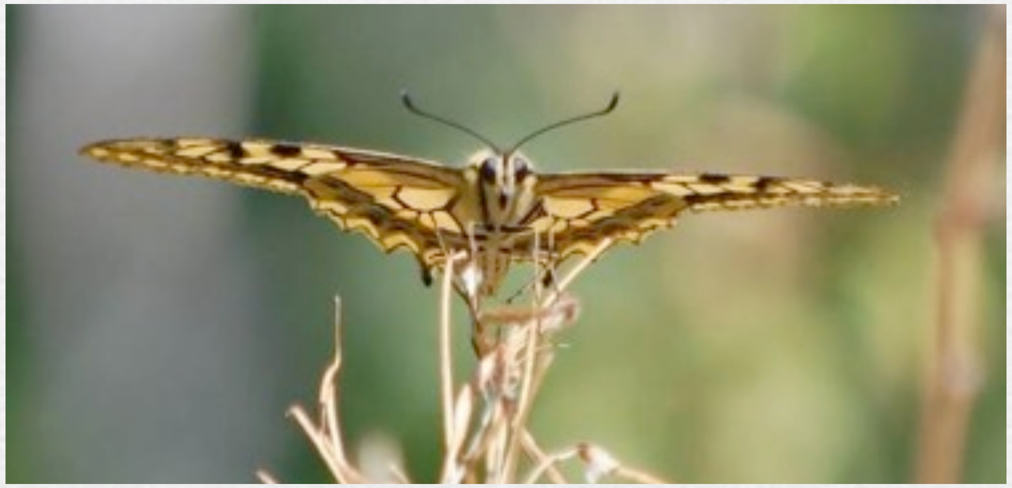






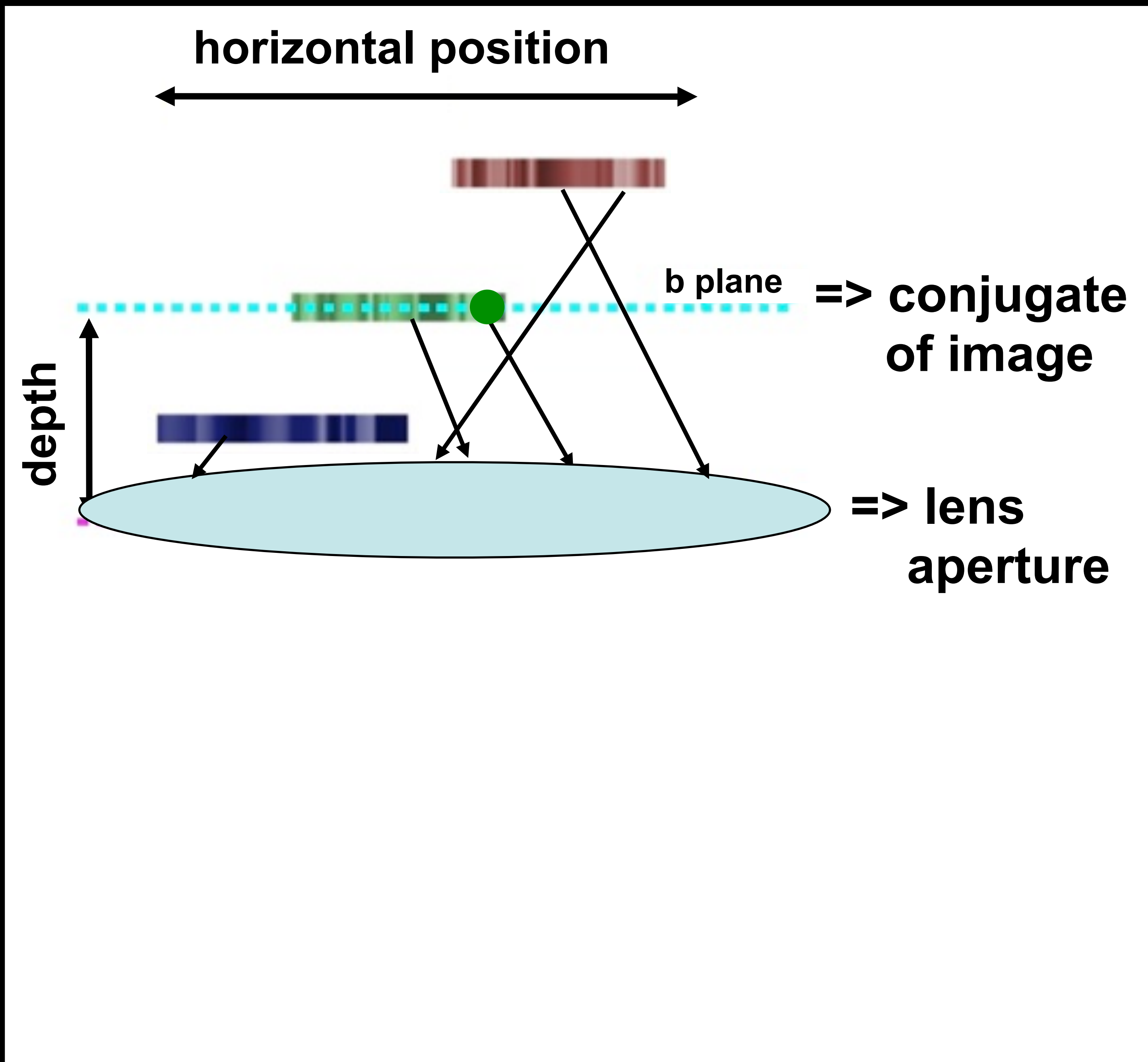
Depth of field

- The same, just different
- Replace 3D space time by 4D light field



Lightfield : 4D radiance reaching lens

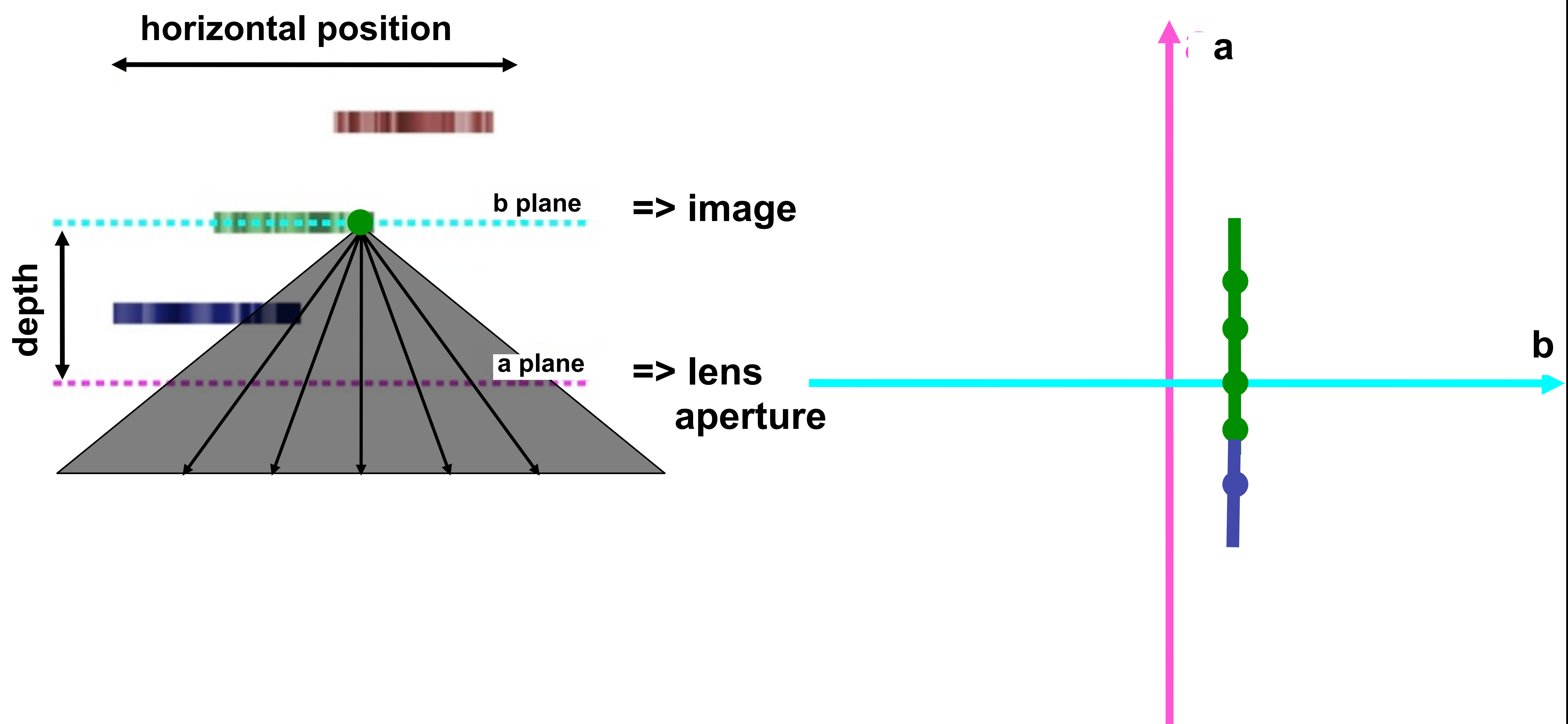
flatland



Lightfield tutorial

flatworld 1D scene

2D lightfield

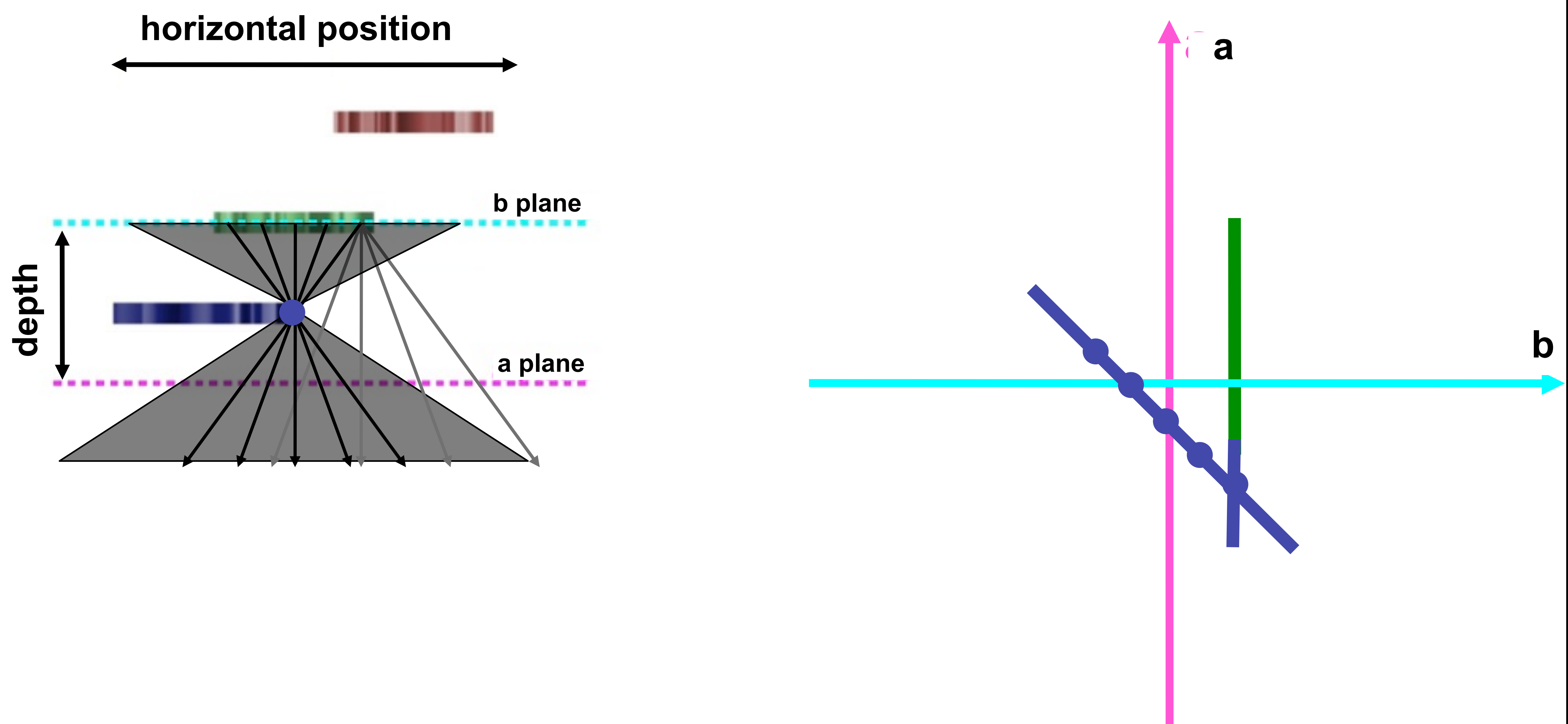


2 plane parameterization [Levoy and Hanrahan 96]

Lightfield tutorial

flatworld 1D scene

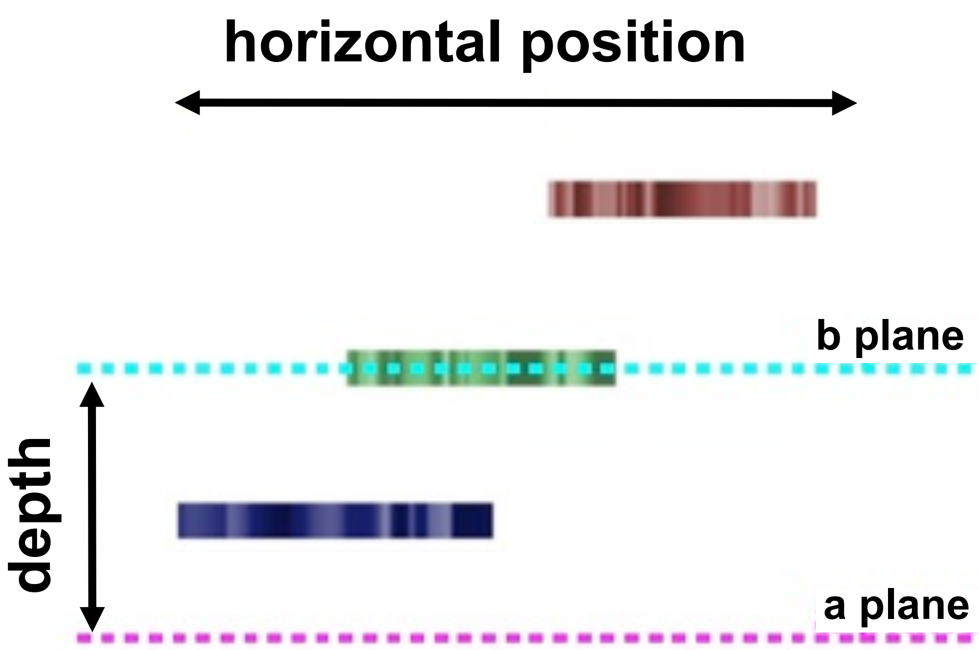
2D lightfield



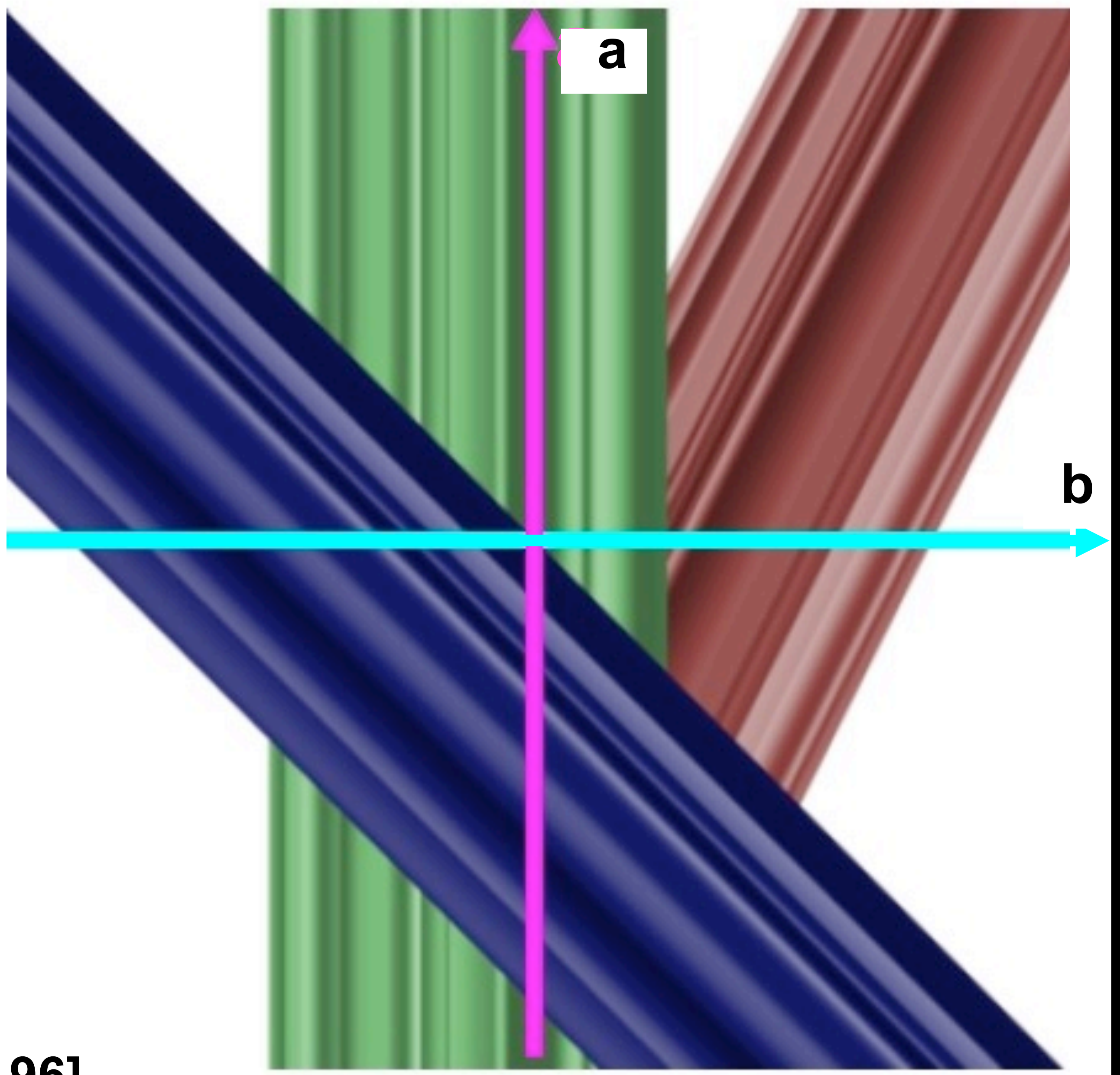
2 plane parameterization [Levoy and Hanrahan 96]

Lightfield tutorial

flatworld 1D scene



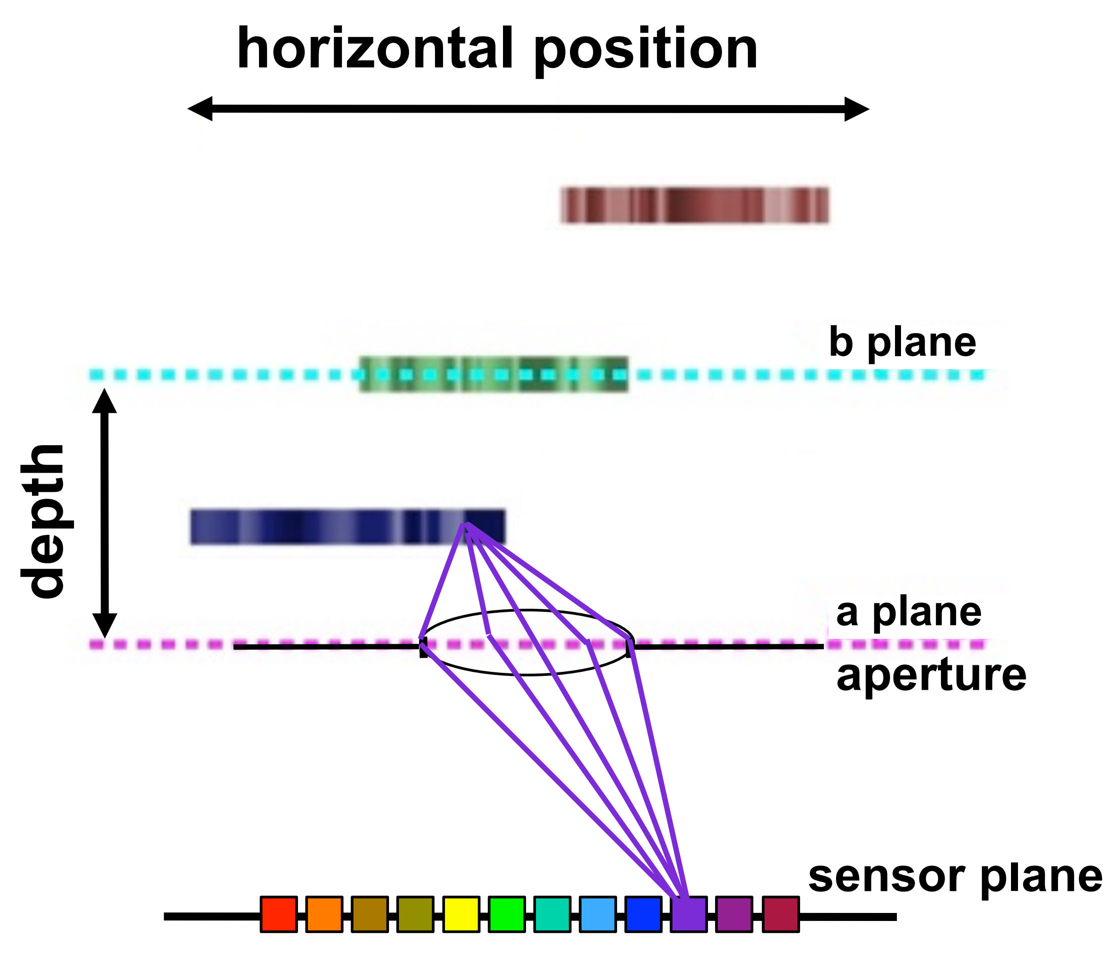
2D lightfield



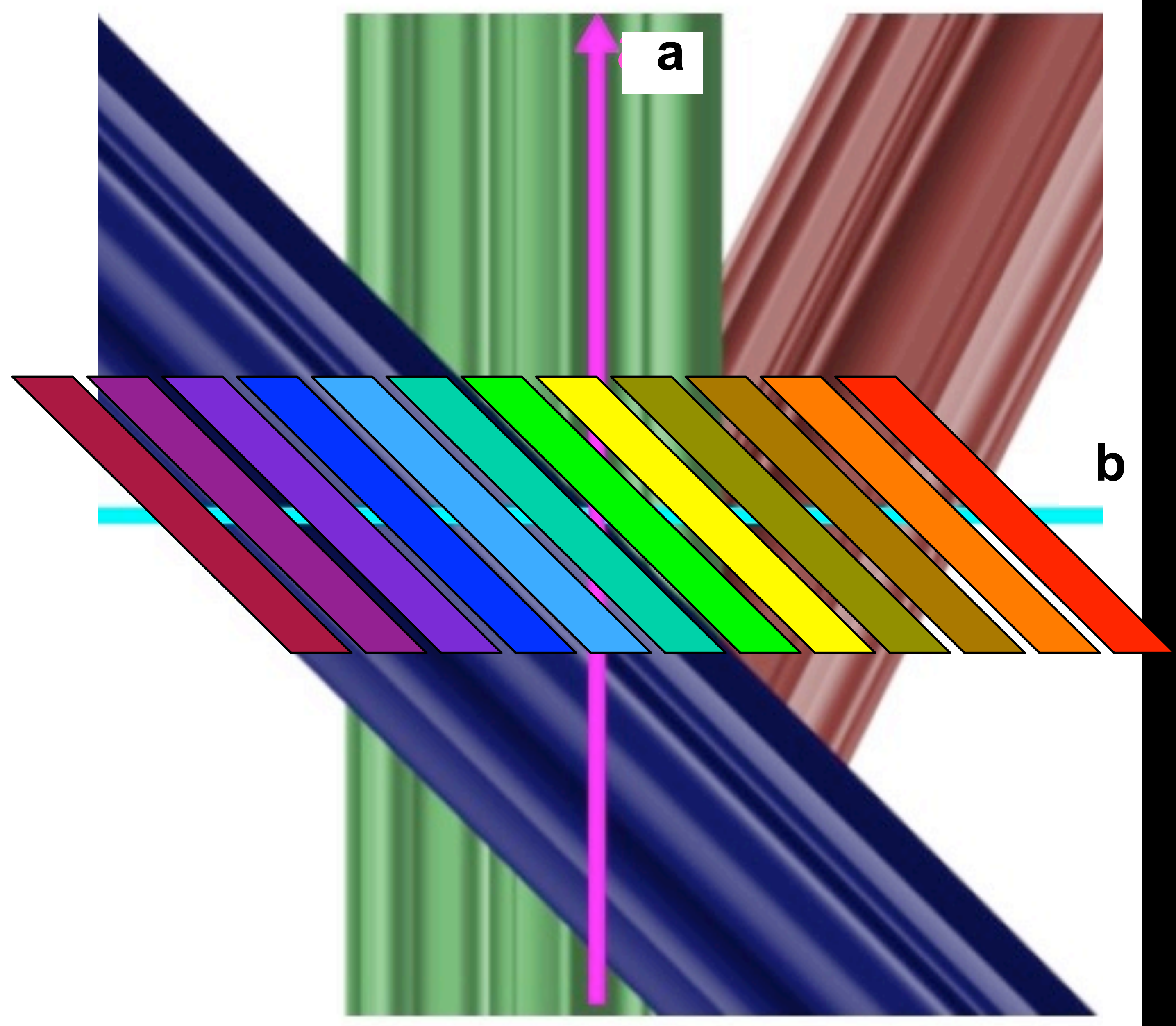
2 plane parameterization [Levoy and Hanrahan 96]

Lens, focused at blue object

flatworld 1D scene



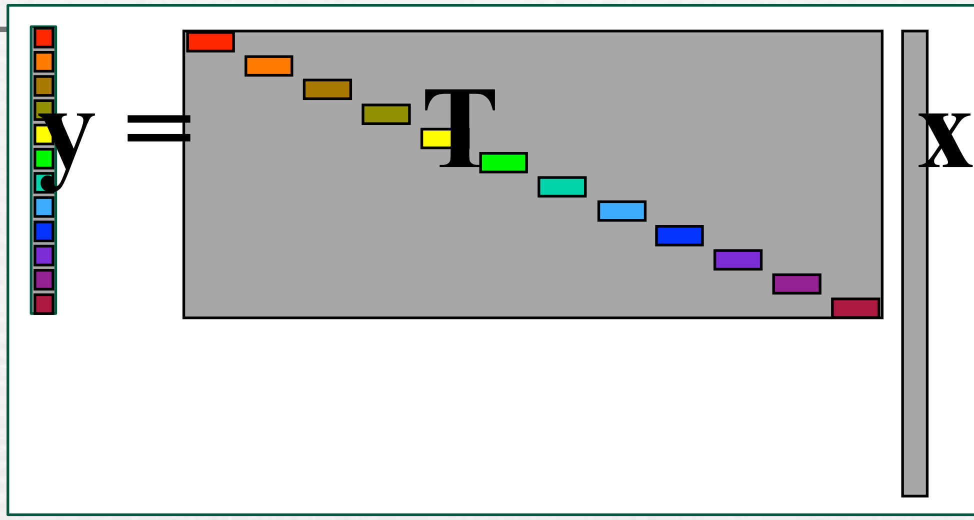
2D lightfield



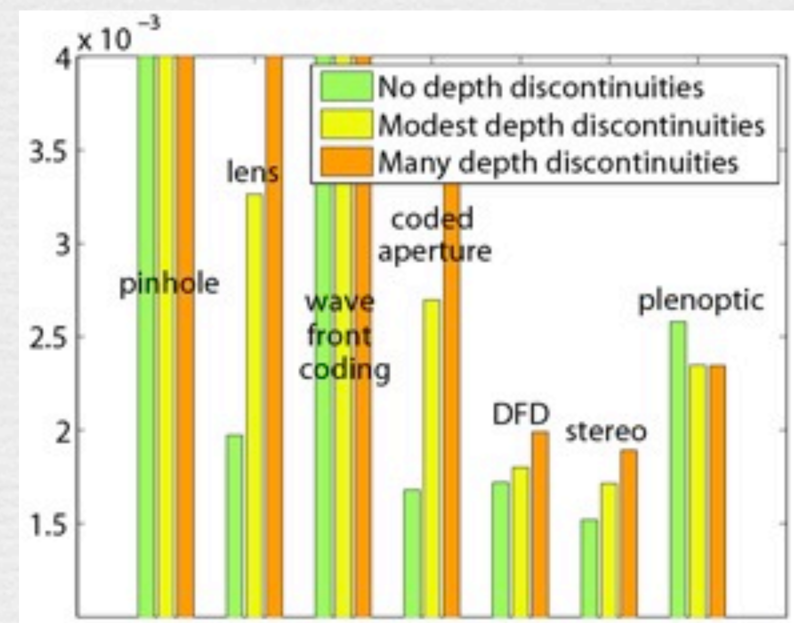
Similarity with motion blur

- Motion blur
  - Domain: 3D space time
  - Slope = 1/speed
  - OTF = slice
- Depth of field
  - Domain: 4D light field
  - Slope = 1/depth
  - OTF = slice
- And very similar to ambiguity function
  - [Zhang, Accardi, Oh]

Bayesian lightfield imaging

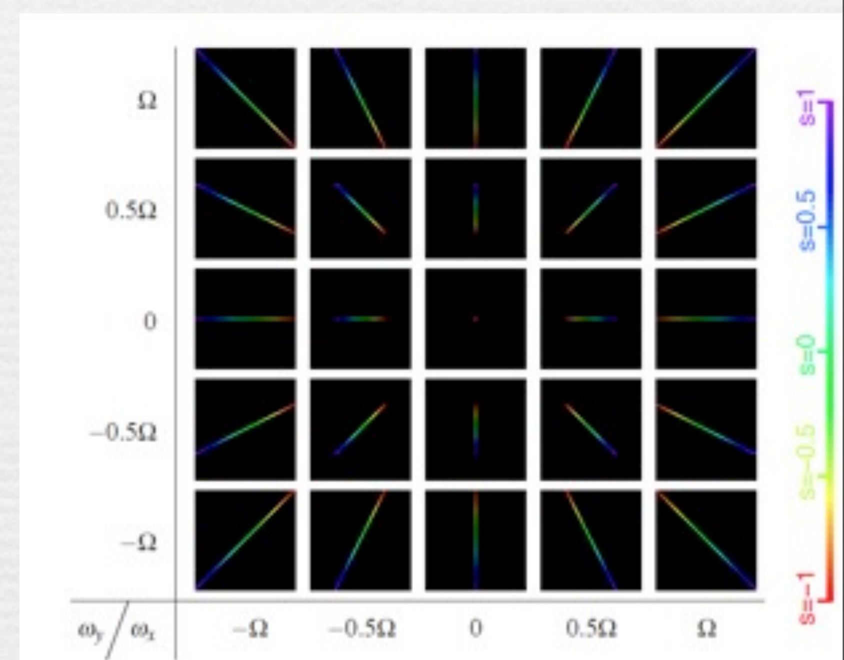


- *[Levin et al. ECCV 08]*
- Show that wavefront coding = parabola in light field
  - This inspired our motion-invariant photography
- Model imaging as linear light field projection
- New prior on light field
- Camera decoding: Bayesian inference problem
- Framework and software for comparison across camera configurations in flatland



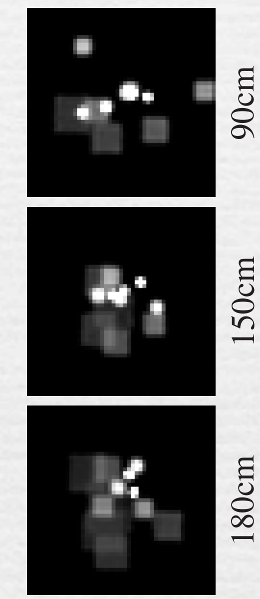
4D frequency analysis of depth of field

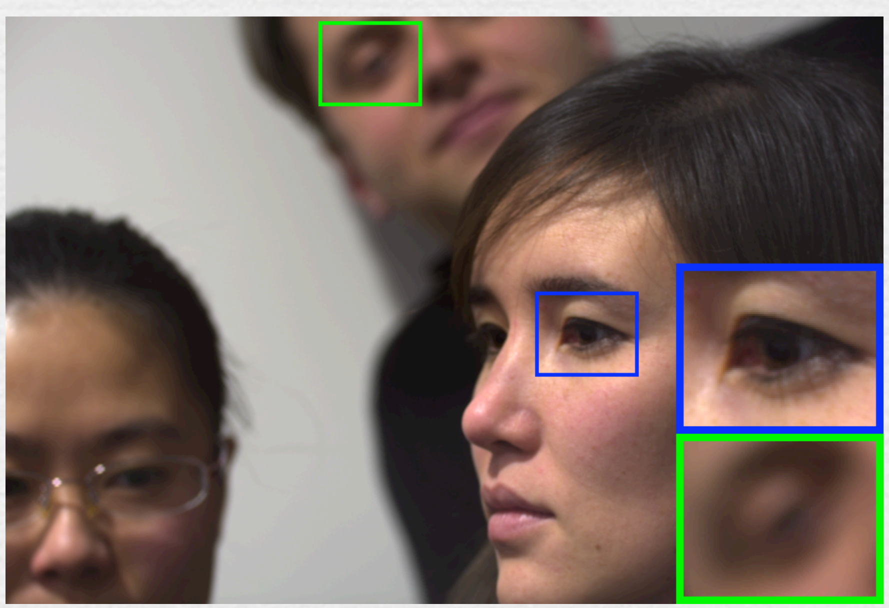
- *Siggraph 2009, with Levin, Hasinoff, Green & Freeman*
- Upper bound
- Dimensionality gap: only a 3D subset of the 4D light field spectrum is useful
  - Images focused at all depths: 1D family of 2D images (3D)
  - Full light field: 4D
- Previous designs spend energy outside this 3D subset



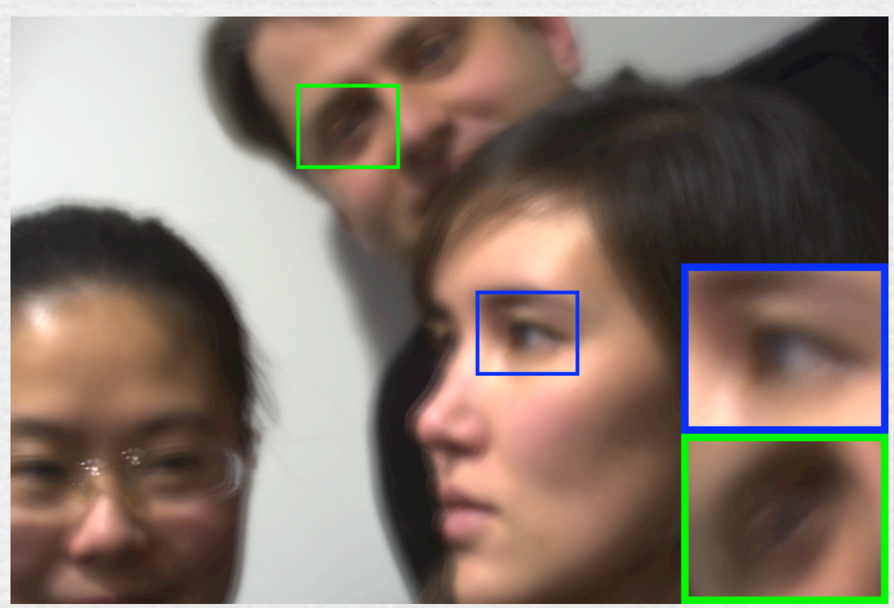
Lattice focal lens

- Takes into account dimensionality gap
- Lattice of lens elements with different powers
- samples the focusing distances
- See Anat levin's talk on Thursday

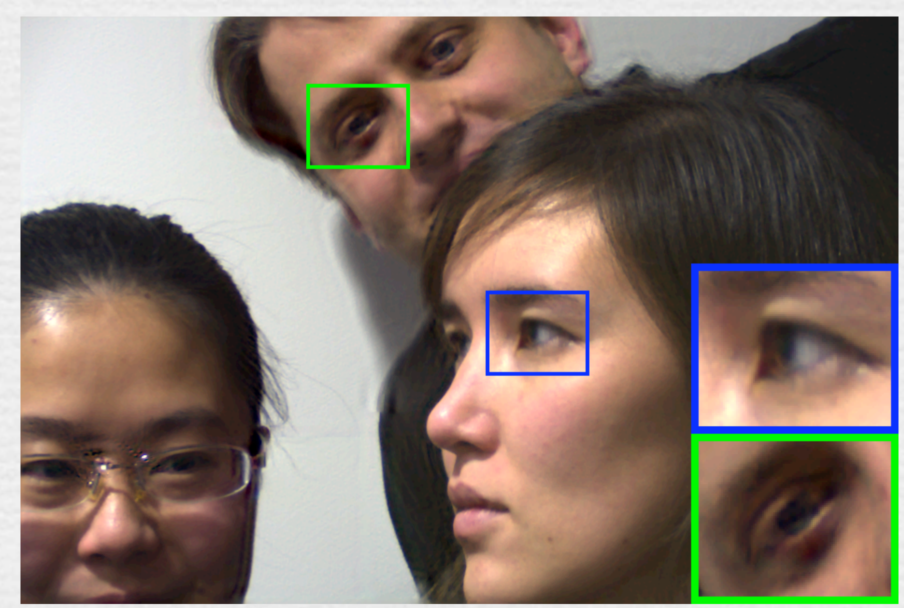




Standard lens image



Our lattice-focal lens: input

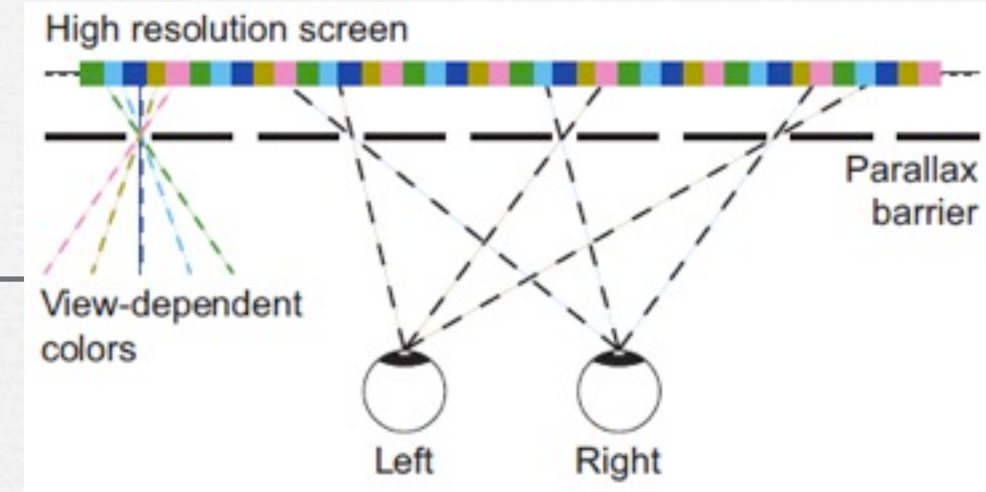


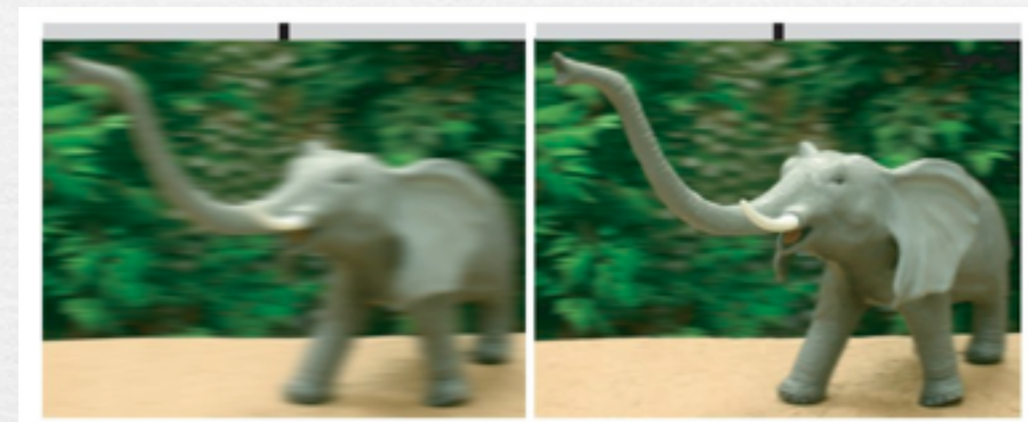
Lattice-focal lens: all-focused output

3D Display

[Zwicker et al. 06]

- 3D Displays
  - lenticular or barrier
  - Pretty much 4D light field displays
- 4D aliasing
  - Depth of field problem as well!





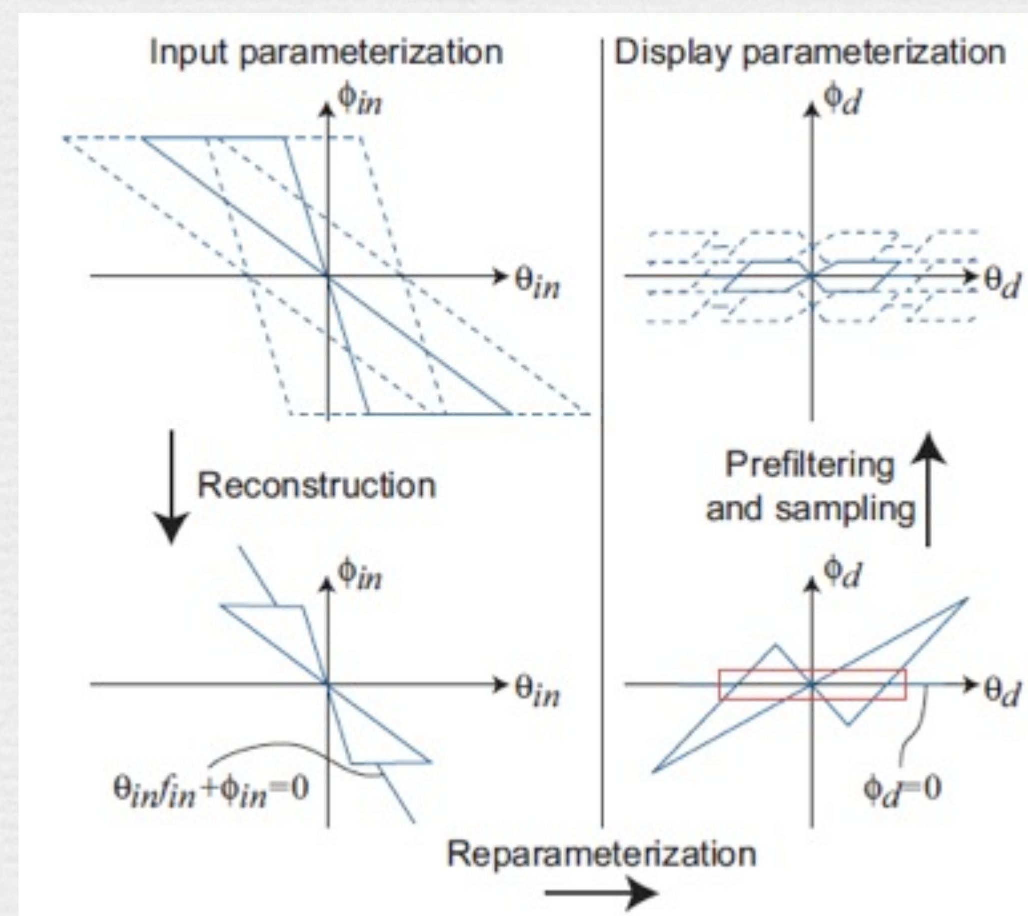
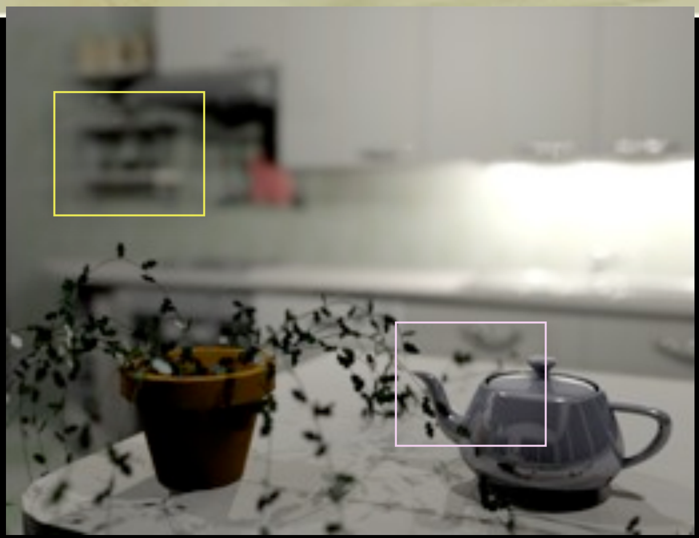
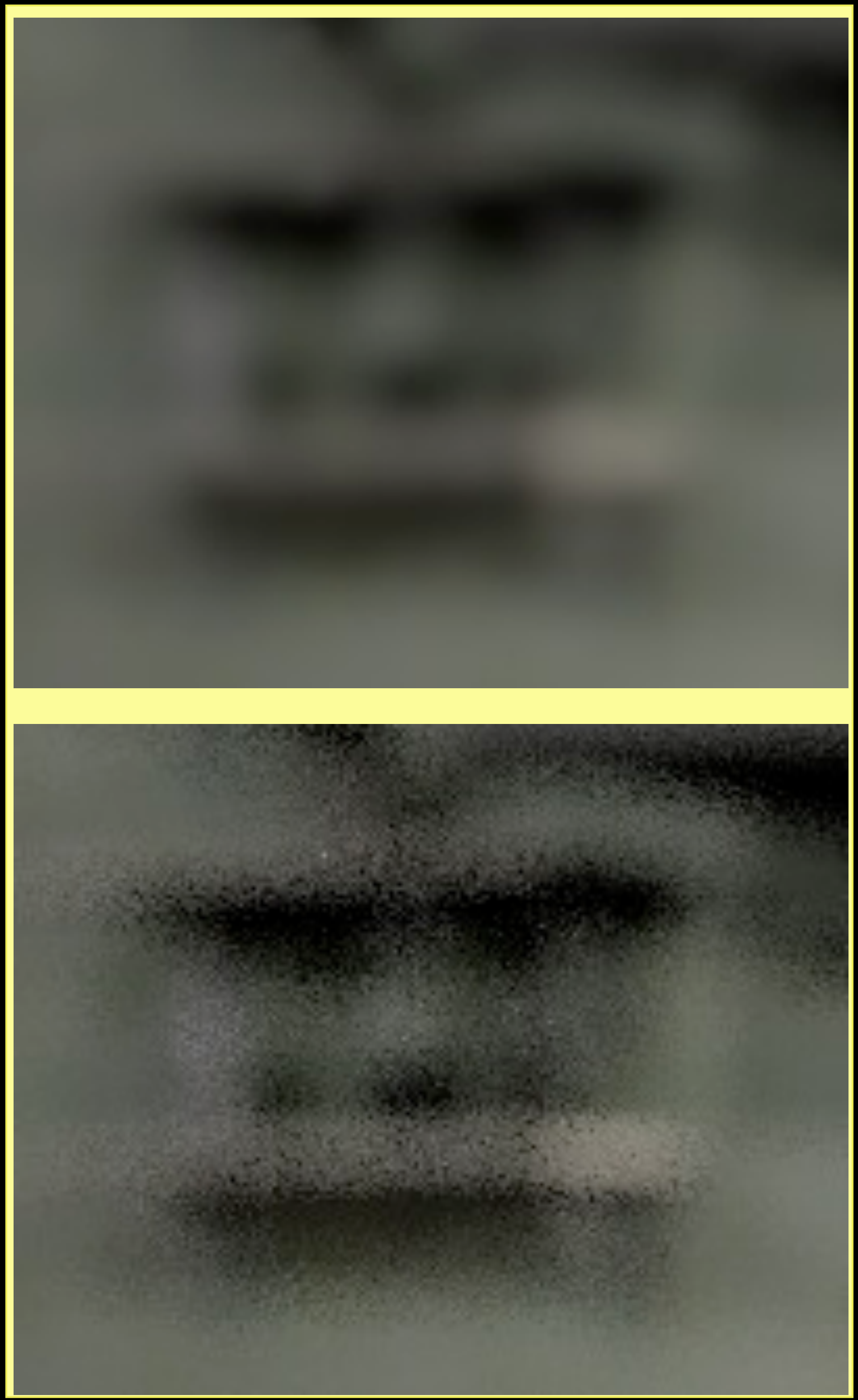


Image synthesis: sparse sampling





Our algorithm
Adaptive sampling

Stratified
lens sampling
(70 lens samp/pixel)

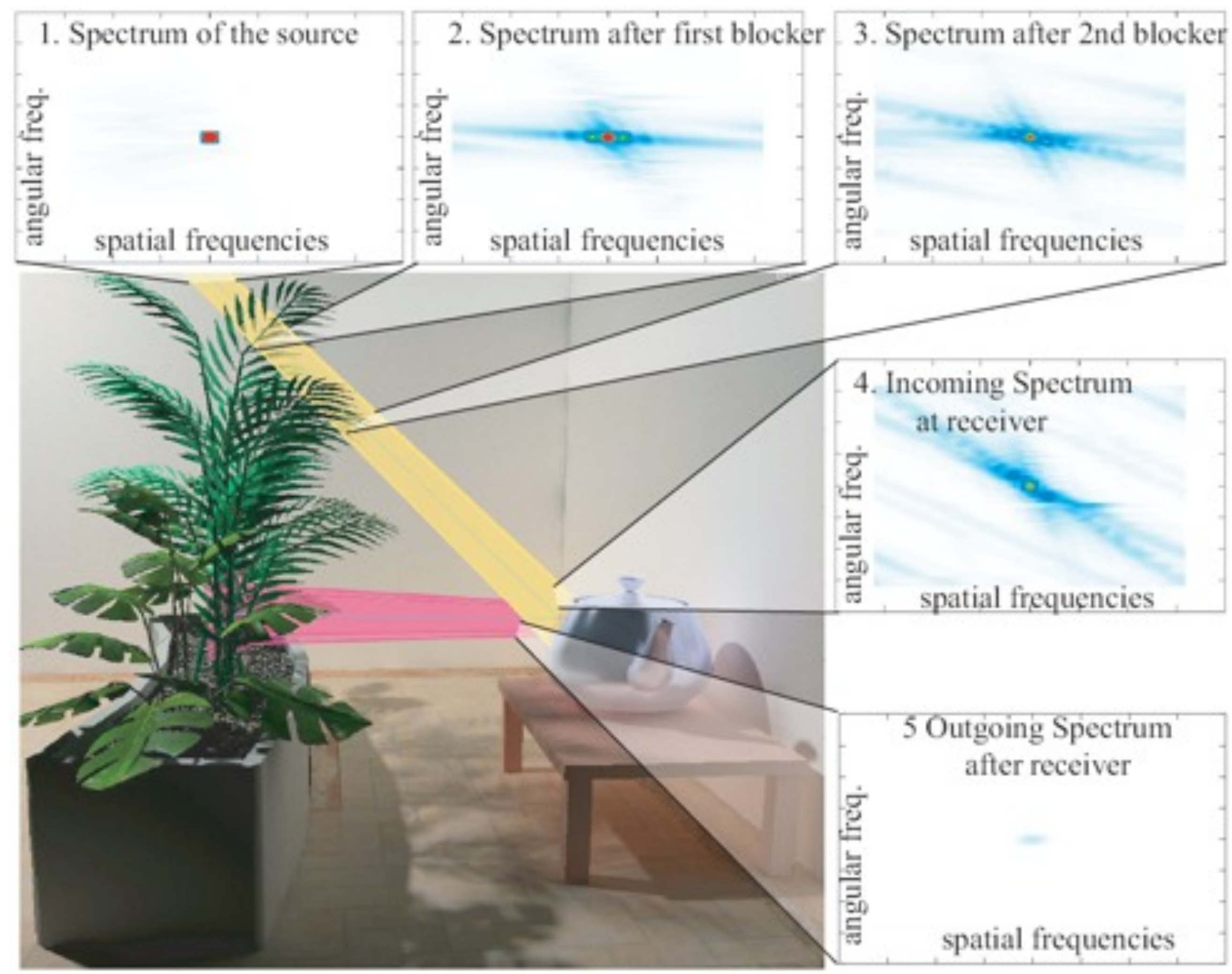
Dimensionalities

- Motion blur:
  - space of (linear) motions: 2D (parameterized by image-space velocity)
  - integration domain (time): 1D
  - Dimensionality mismatch => not enough integration dimensions => 1D motion only
- Depth of field
  - space of distances: 1D
  - integration domain (aperture): 2D
  - Dimensionality mismatch: too many integration dimensions => loss of optimality, only subset of 4D space is useful

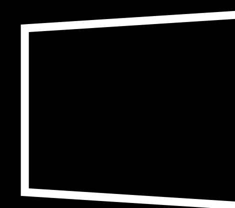
Fourier analysis of Light transport

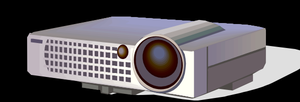
- 4D radiance signal in neighborhood of ray
- Light sources are input signal
- Interactions are filters/transform
  - Transport in free space
  - Visibility
  - shading by BRDF
  - Etc.

- *[Durand et al. 2005]*



Summary:
Frequency in space & angle



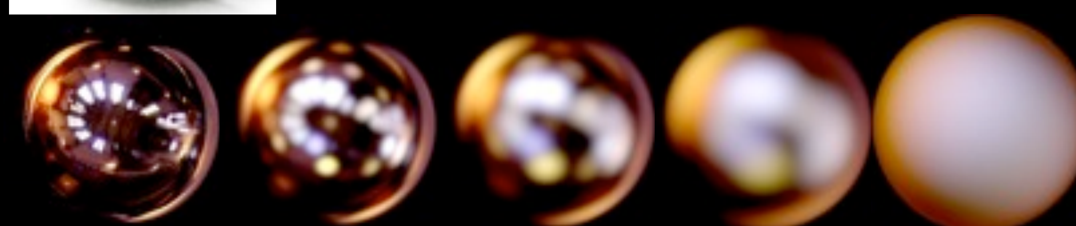


- Transport in free space
  → shear

- Occlusion
  → multipl./convolution

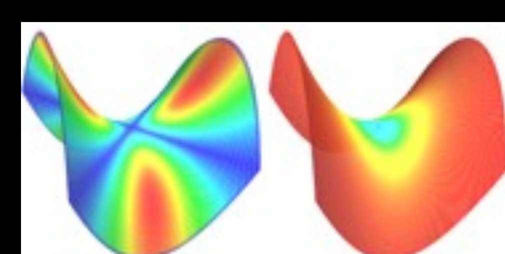
  Simpler in Ray space
- BRDF
  → convo./multipl.

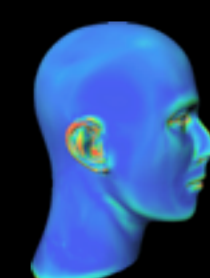
  Simpler in Fourier



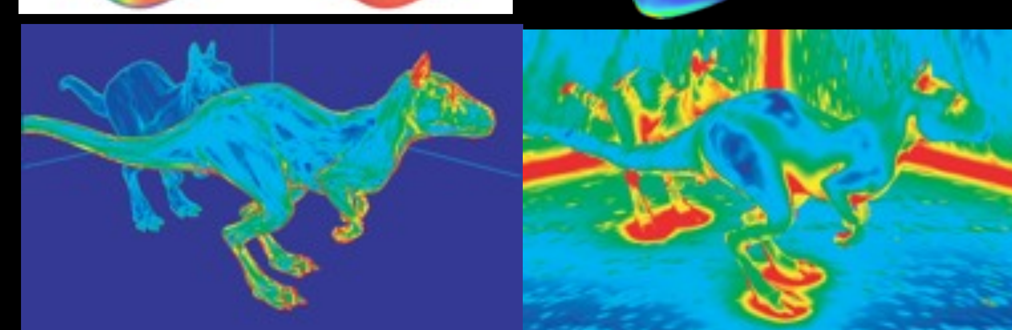
- Curvature

  → shear



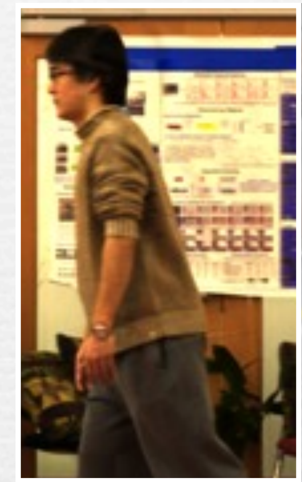


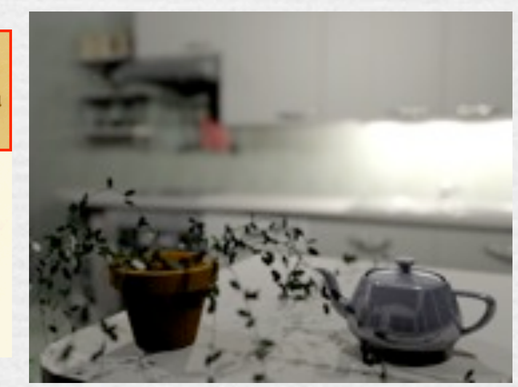
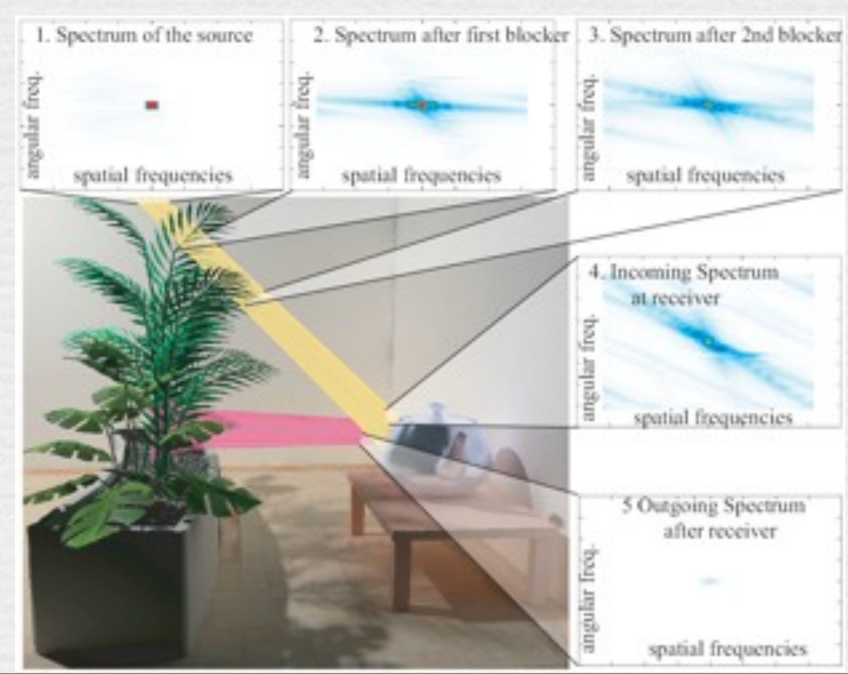
- Proof of concept:
  – Bandwidth prediction



Recap: Fourier analysis for

- Motion blur
    - 3D space time
    - Shear, slope = 1/speed
    - Motion-invariant photo, upper bound
    - Synthesis: sparse sampling, sheared filter
- Depth of field
    - 4D light field
    - Shear, slope = 1/depth
    - Dimensionality gap, upper bound
    - Lattice-focal lens
    - 3D display depth of field
    - Synthesis: sparse sampling
- Light transport in scenes



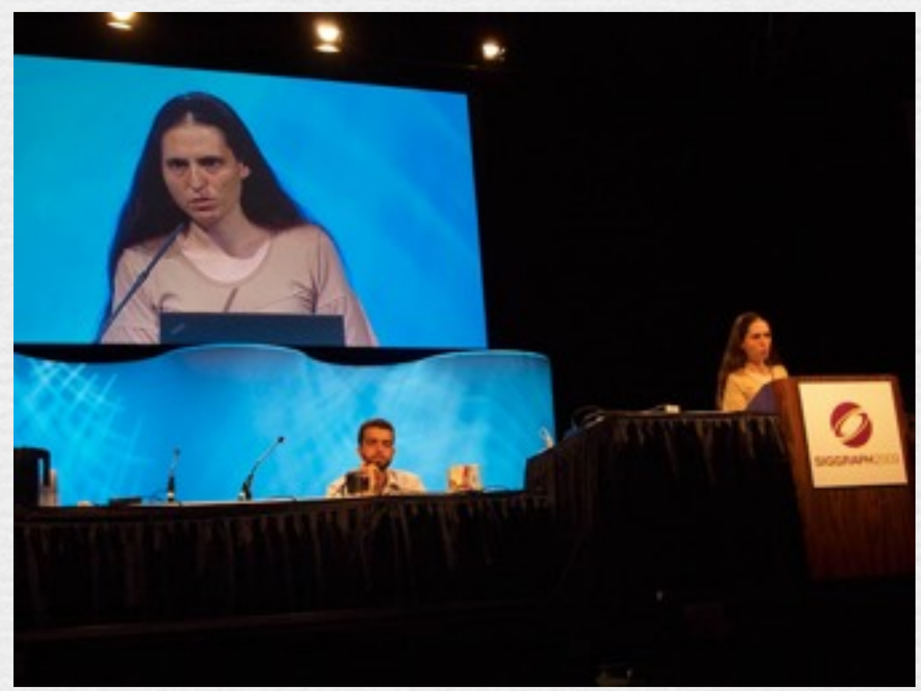




Acknowledgements

- Anat Levin, Bill Freeman, Paul Green, Sam Hasinoff, Rob Fergus, Kevin Egan, Ravi Ramamoorthi, Nicolas Holzschuch, Cyril Soler, Kartic Subr, Francois Sillion, Eric Chan, Matthias Zwicker, Wojciech Matusik, Yu-Ting Tseng, Taeg Sang Cho, Peter Sand

Motion-invariant photography: Levin et al.

Motion-blur image synthesis: Egan et al.



Space of cameras, depth of field, lattice-focal lens Levin et al.

Depth of field image synthesis: Soler et al.

- Funding: NSF, Quanta, Foxconn, Shell, INRIA