Light Field Reconstruction

20154471 InJae Yu

CS580 Paper Presentation
2016. 06. 09

First Paper

- Layered Light Field Reconstruction for Defocus Blur
  - Siggraph 15

  - Sheared reconstruction filter of **Defocus Blur**
    - **Screen-Independent** filter
  - Reconstruction by compositing depth layers

Light Field and Irradiance

- Light Field : $(x_i, y_i, u_i, v_i) \longmapsto (z_i, l_i)$
  - (Pixel, Lens)->(Depth, Radiance)
- Partition the light field to have similar depth z

- $e_i(x)$: Irradiance at pixel x from layer i
  - integra $e(x) = \int l(x, u) a(u) du$ over the lens in certain layer

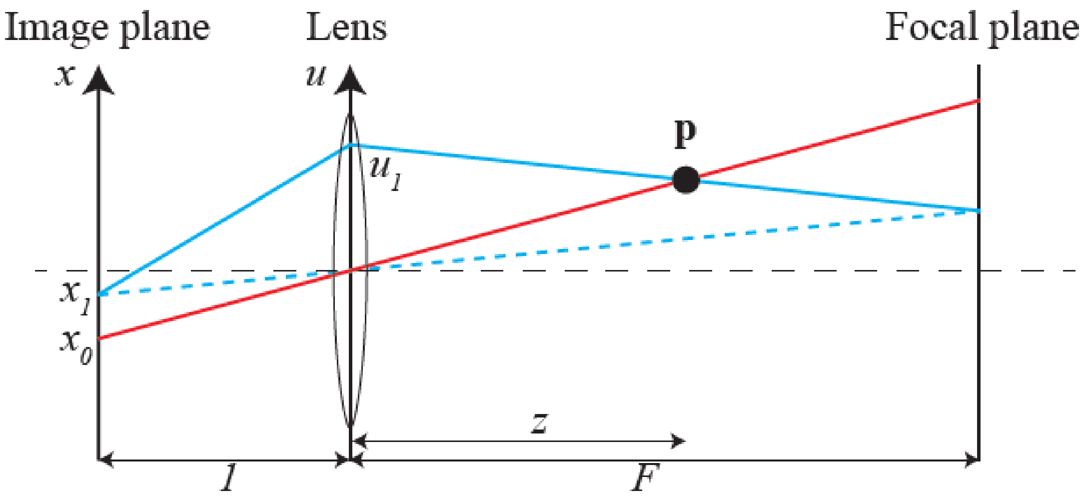
Aperture of camera

Radiance-Depth Relation

- With certain depth, radiance can be converted to function of x

$$x_0 = x_1 + \boxed{\frac{z-F}{zF}} u_1 = x_1 + c(z)u_1$$

$$l(x,u) = l(x + c(z)u, 0) = l^0(x + c(z)u)$$



Frequency Response of Radiance

- Fourier Analysis of radiance[Egan et al 2011]

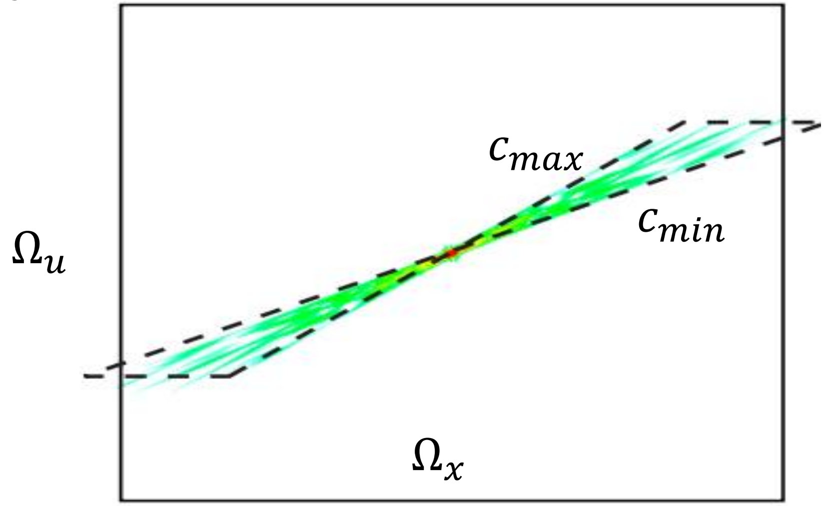
$$l(x,u) = l^0(x + c(z)u) \Longrightarrow L(\Omega_x, \Omega_u) = L^0(\Omega_x)\delta(\Omega_u - c(z)\Omega_x)$$

- Frequency response is sheared along the line

$$\Omega_u - c(z)\Omega_x = 0$$

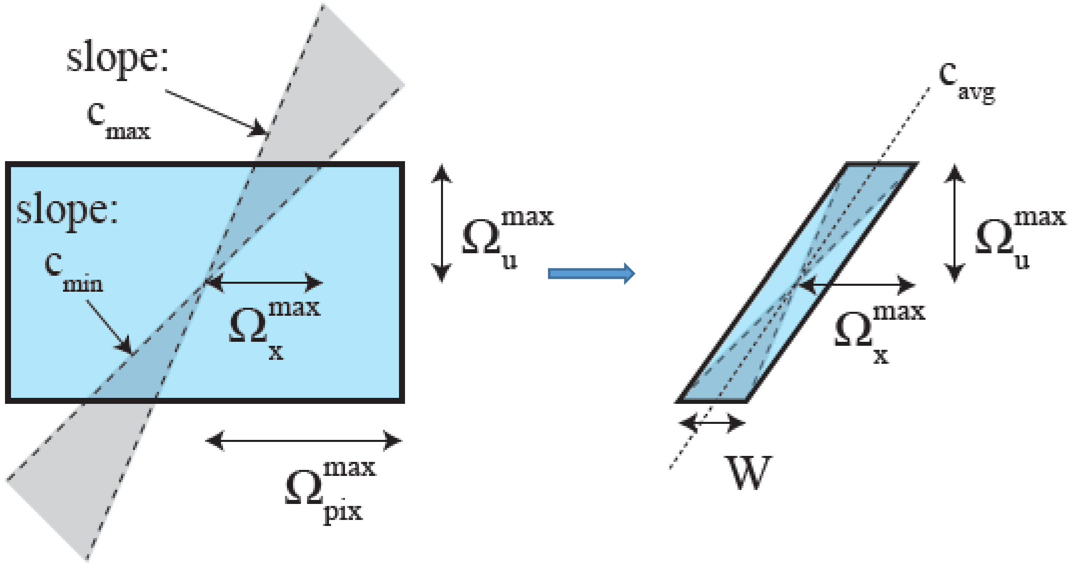
  - Calculate max and min value of $c(z)$ in each layer

- Bound the frequency with sheared filter



Filter Design

- Axis Aligned Filter & Sheared Filter
  - Axis Aligned : **Easy to compute**, require **many samples**
  - **Sheared Filter** : **Hard to compute**, require **small samples**



Screen-Independent Filter

- Previous Sheared Filter [Egan et al 2011]

$$w_{\text{shear}}(x,u) = w(x;\sigma_x)\boxed{w(u + \frac{x}{c_{\text{avg}}};\sigma_u)}$$

Per-pixel Defined Filter

- Convert the aperture filter to be **independent from $\boldsymbol{x}$**

$$\begin{aligned} w_{\text{shear}}(x,u) &= w(x;\sigma_x)\, w(u + \frac{x}{c_{\text{avg}}};\sigma_u) \\ &= \underbrace{w(x + \eta\, u;\sigma'_x)}_{w_x}\,\boxed{\underbrace{w(u;\sigma'_u)}_{w_u}}. \end{aligned}$$

Pixel Independent

Computational Efficiency

- Irradiance of light field at **one depth layer**

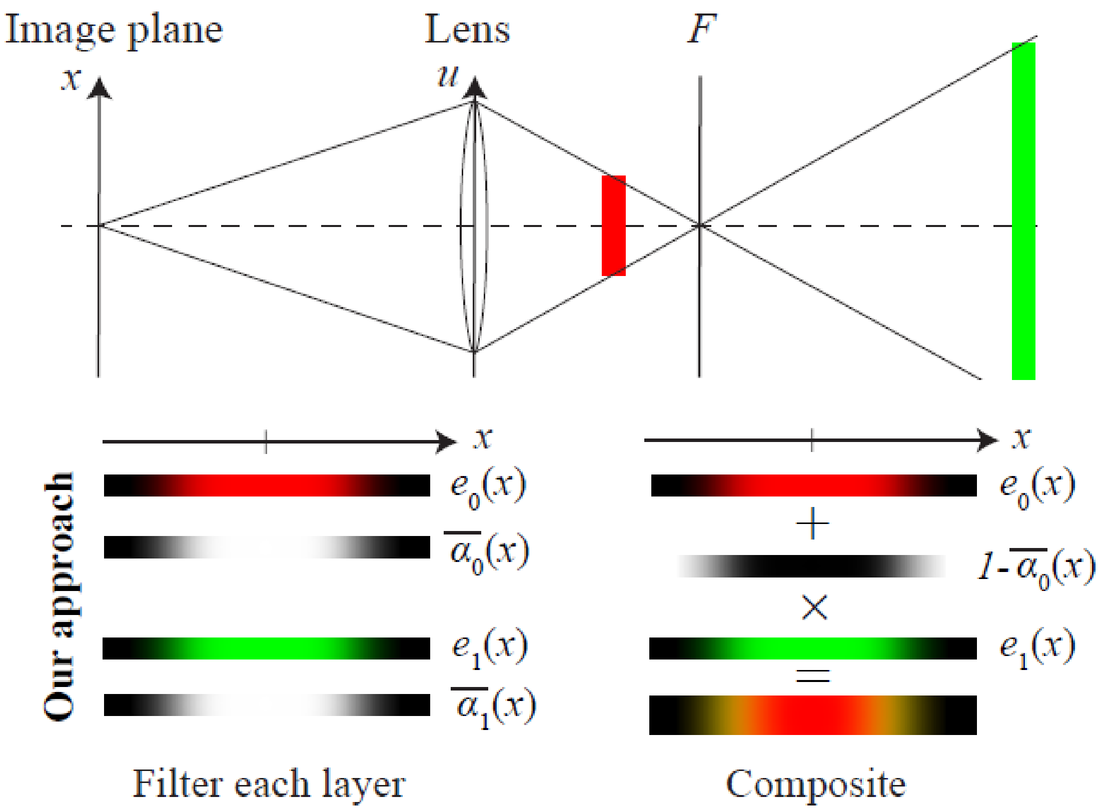
$$
\begin{aligned}
e(x) &\approx \iint l(x', u) w_{\text{shear}}(x' - x, u)\, du\, dx' \\
&= \iint l(x', u) w_x((x' - x) + \eta\, u) w_u(u)\, du\, dx' \\
&\quad [\,\text{Let } q = x' + \eta\, u,\ dq = dx'\,] \\
&= \int \underbrace{\left[\int w_u(u) l(q - \eta\, u, u)\, du\right]}_{I_l(q)} w_x(q - x)\, dq \\
&= \int I_l(q) w_x(x - q)\, dq \;=\; (I_l * w_x)(x),
\end{aligned}
$$

Perform only once for each layer

1. Multiply radiance sample $l(x_i, u_i)$ with $w_u(u_i)$
2. Accumulate $l(x_i, u_i) w_u(u_i)$ at the pixel $q \rightarrow I_l$
3. Convolution $(I_l * w_x)(x)$, I_l is reused for each pixel

Depth Layer Composition

- Different layers may occlude each other
  - Weighted(Opacity $\overline{\alpha_i(x)}$) sum of irradiance from each layer



Layer Opacity

- $\overline{a_i(x)}$: Visibility of x at Layer i
  - *Integration of visibility* $a_j(x, u)$ *over lens*

$$\bar{\alpha}_j(x) = \int \alpha_j(x, u) a(u) du$$

- Same as irradiance calculation

$$e(x) = \int l(x, u) a(u)\, du$$

  - Calculate irradiance and visibility at once, using same sheared filter

Result

- Compared methods
  - TLFR, Lehtinen et al.'s [2011]
    - Temporal light field reconstruction for rendering distribution effects
  - OurAA, Mehta et al.'s [2012]
    - Axis-Aligned Filtering for Interactive Sampled Soft Shadows

  - Filtering after generating 8 samples per pixel

	TLFR		OUR			
Scene	CPU	GPU A	CPU	GPU A	GPU B	Avg. Layers
DRAGON	73920	12721	7472	31.1	131.2	3.5
CITADEL	74963	14023	9130	39.5	150.8	4.8
SAN MIGUEL	113907	22333	14072	75.9	269.9	9.5

(a)
(b)
(c)
(d)
Input
OurAA
TLFR
Our
256 spp
Input
OurAA
TLFR
Our
256 spp

Conclusion

- Screen-Independent Sheared Filter
  - Computation enhancement

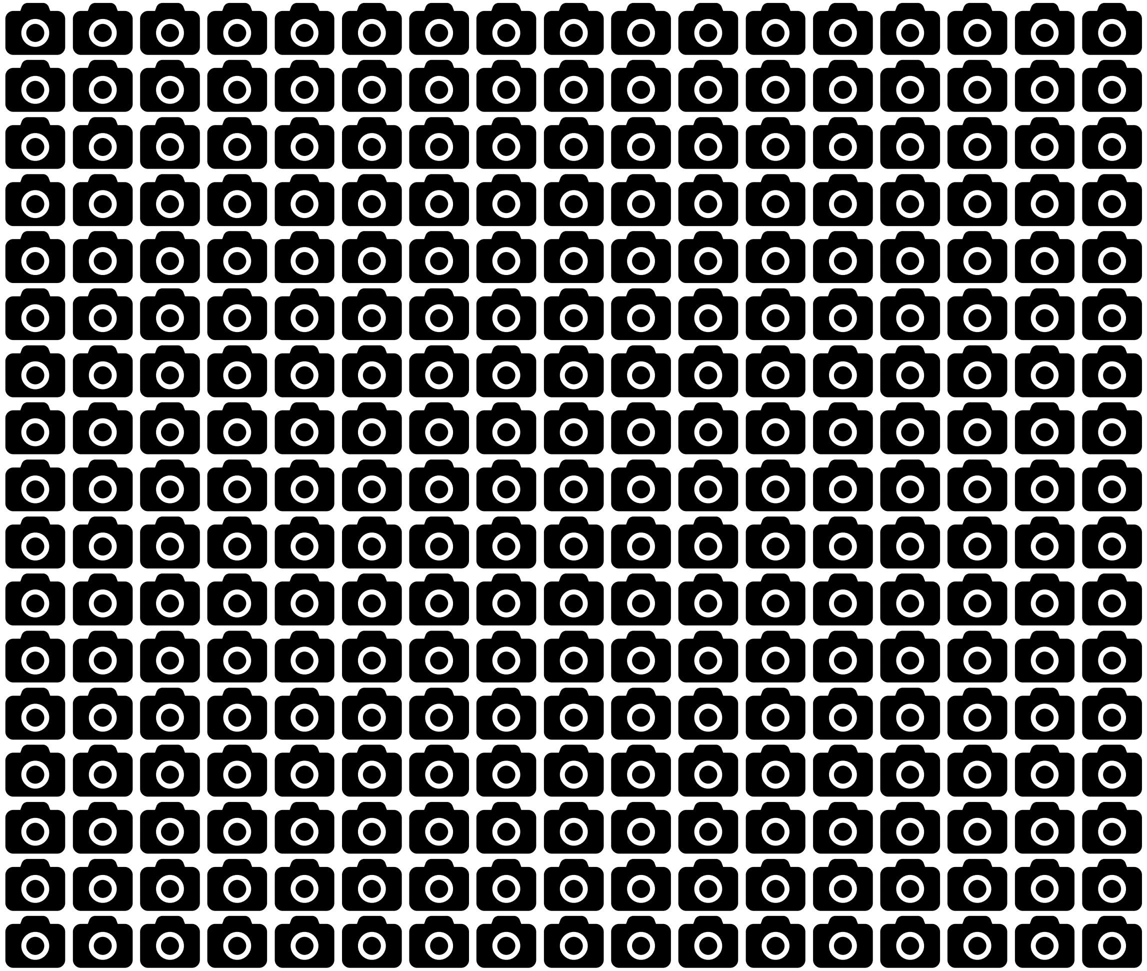
- Depth Layer Composition
  - Calculate with irradiance simultaneously

Second Paper

- Light Field Reconstruction Using Sparsity in the Continuous Fourier Domain
  - Siggraph 15

  - Light Field reconstruction for **viewpoint images**
  - Construct **sparse continuous Fourier transform** of light field

Goal



Reduced Number of Cameras

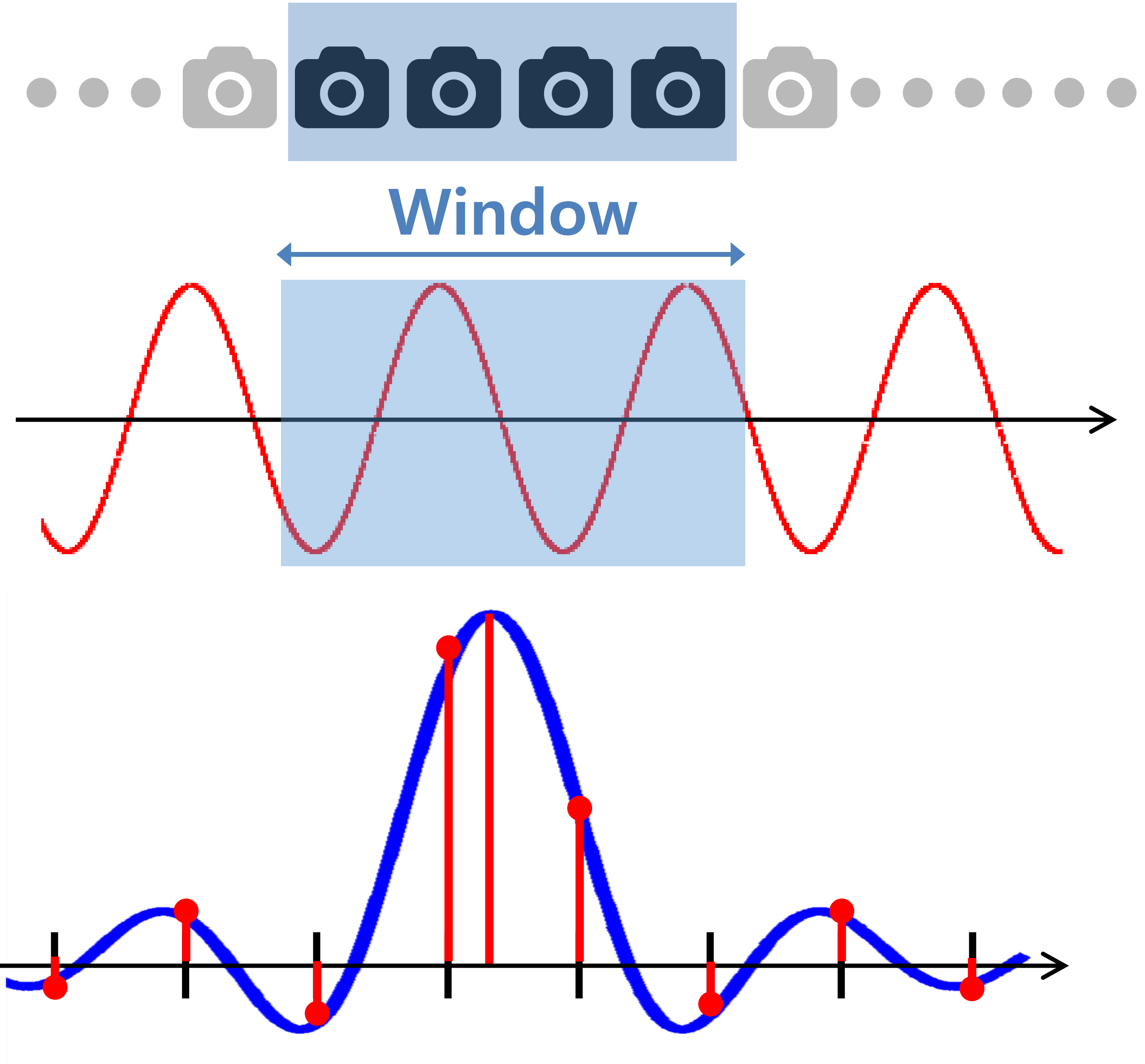
Dense Array of Cameras

$$O(n^2)$$

$$O(n)$$

[Slides from Author]

Continuous and Discrete Fourier Transform



SignalA

Continuous Frequency

Discrete Frequency

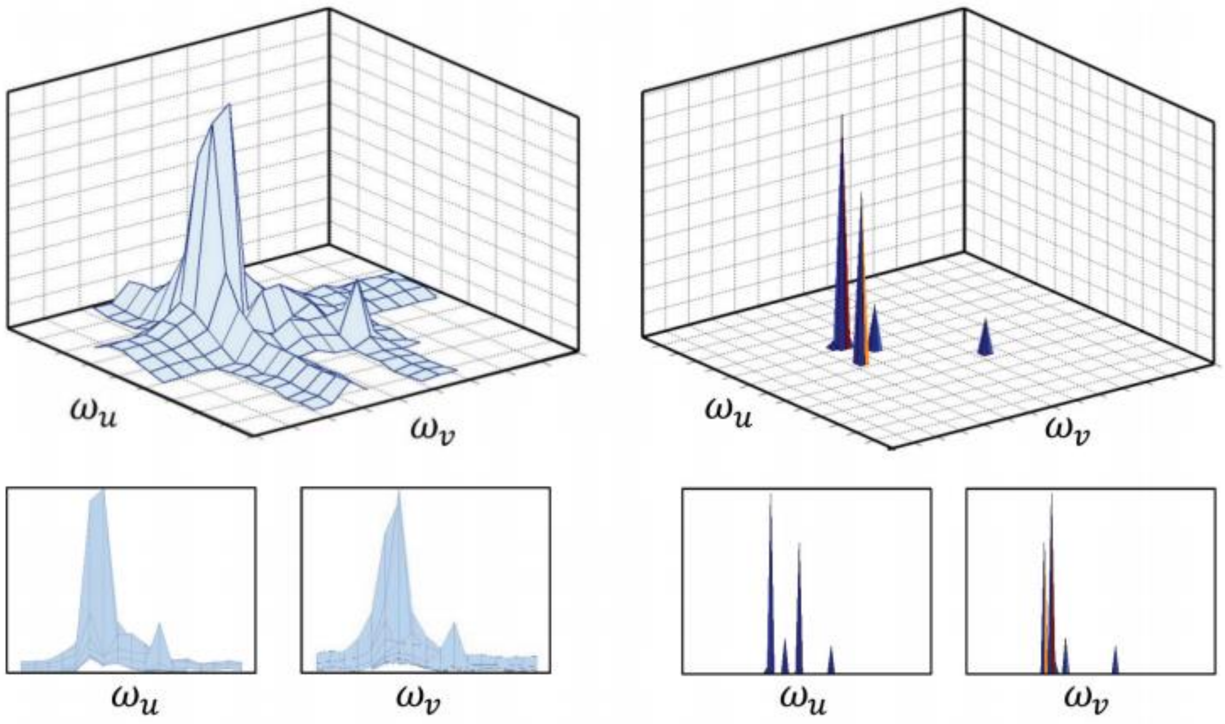
Multiply by Box Window

Convolve with Sinc Function

[Slides from Author]

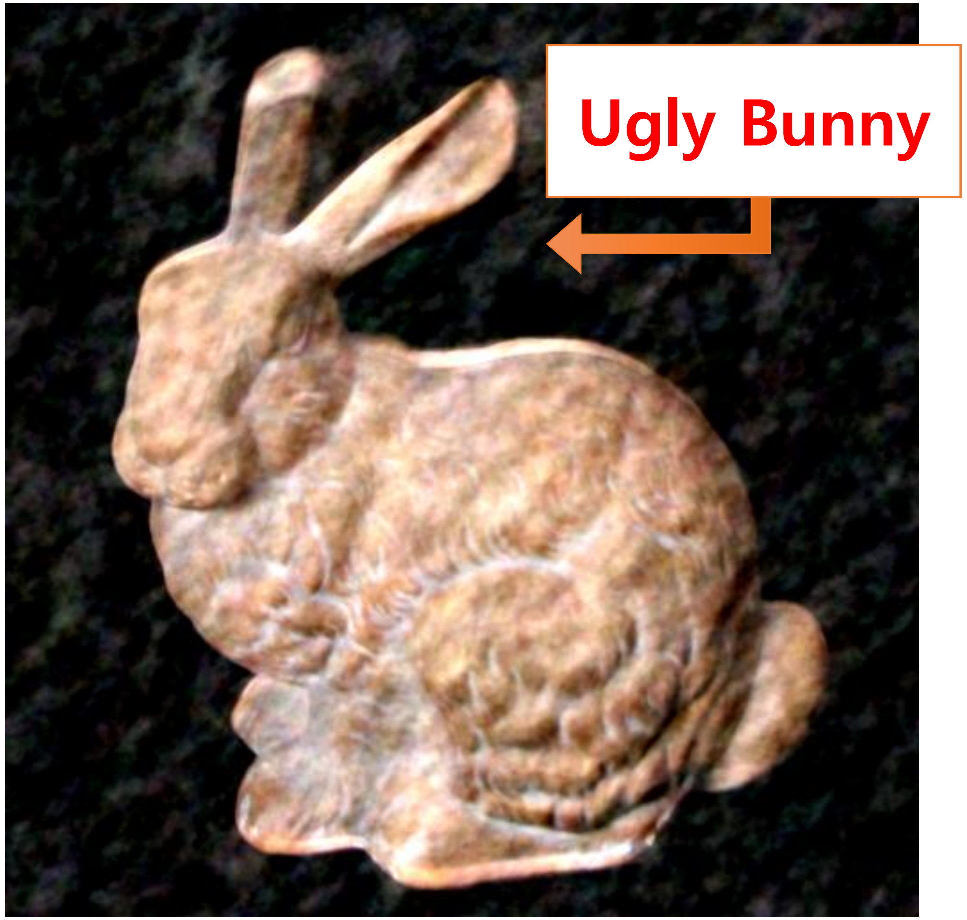
Problem of DFT

- Sparsity of light field spectrum in discrete domain is much less sparse than continuous domain

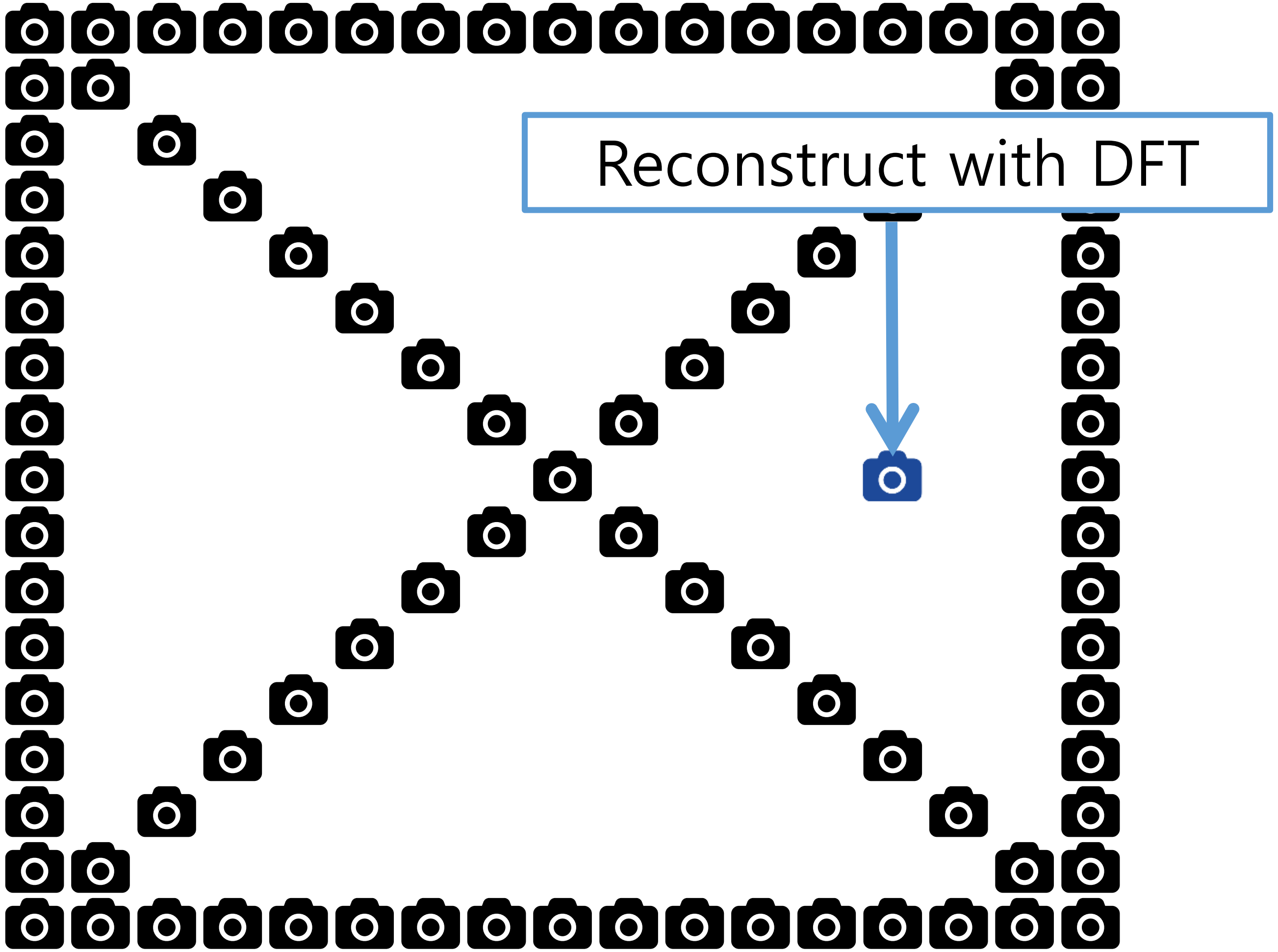


(a) discrete spectrum (b) continuous spectrum

Traditional Sparse Recovery



Sparse Recovery



Captured Viewpoints

[Slides from Author]

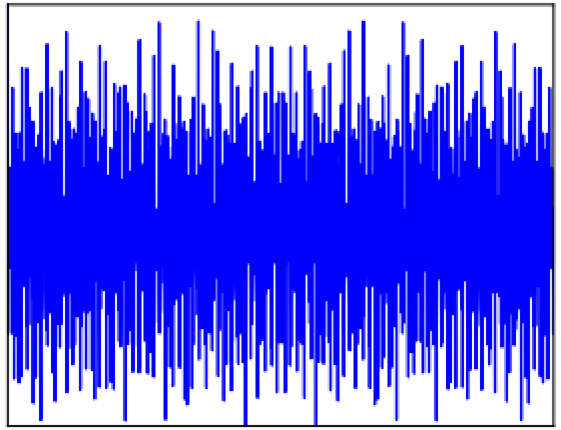
Sparse Continuous Fourier Spectrum

- CFT of k peaks

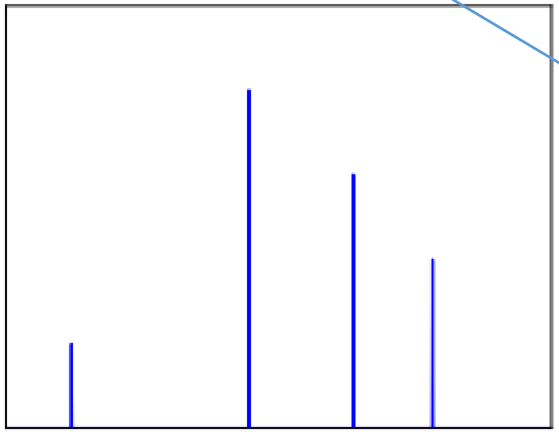
$$x(t) = \frac{1}{N}\sum_{i=0}^{k} a_i \exp\left(\frac{2\pi j t \omega_i}{N}\right)$$

But how do we know

1. # of peaks
2. Coefficient of frequency
3. Frequency

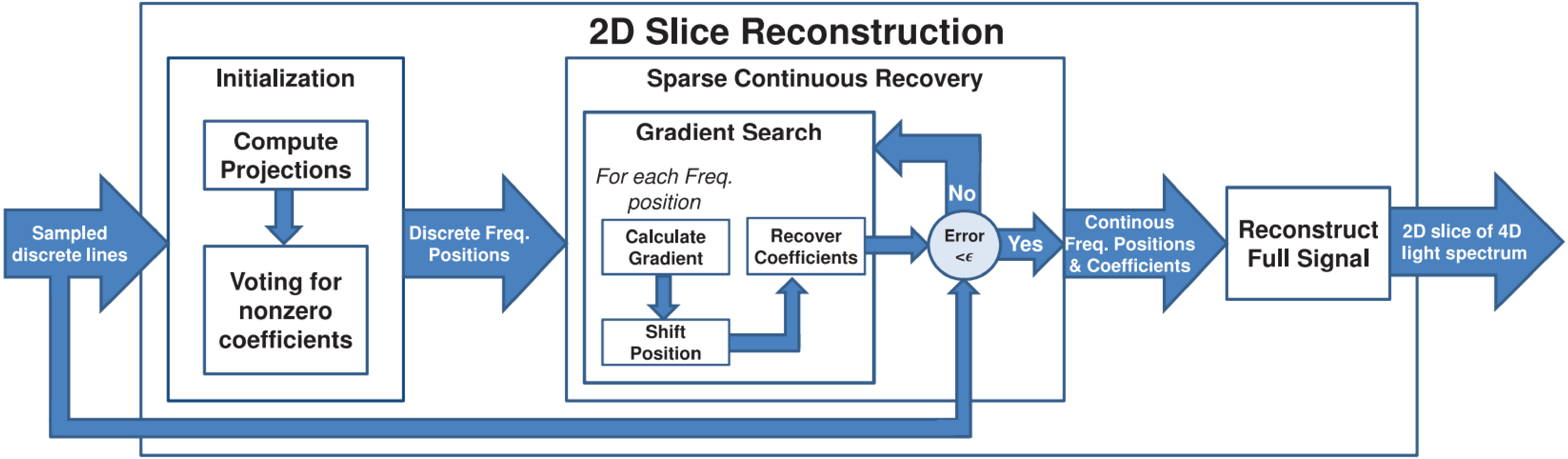


Signal



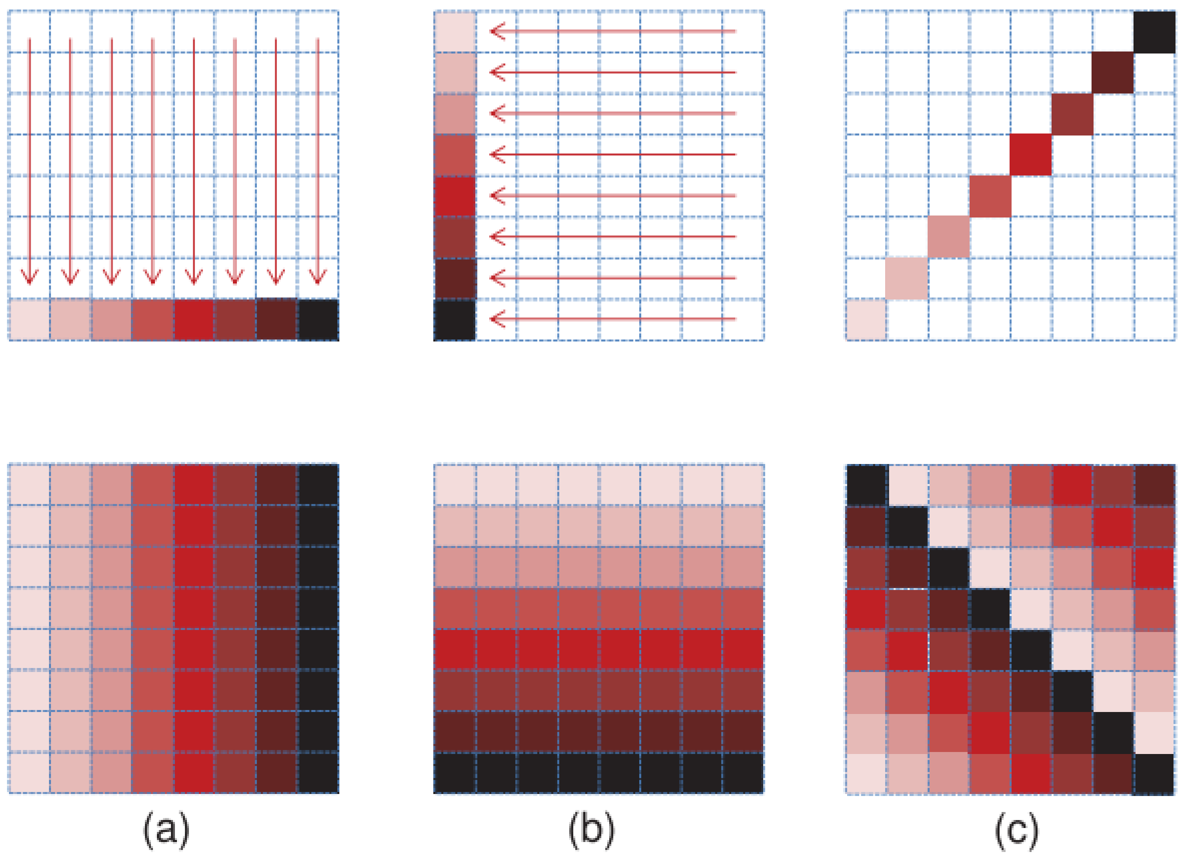
Frequency of k peaks

Overall Process



Initialization – Find the # of peaks

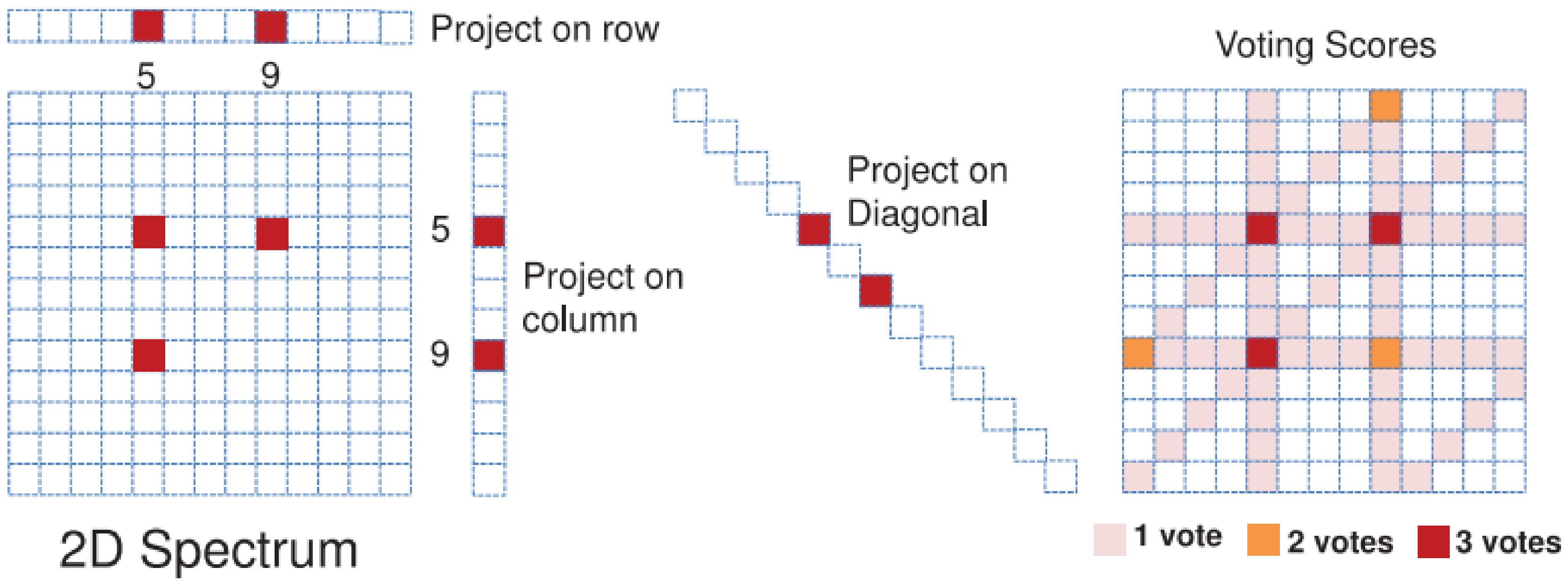
Compute the DFT of row, column and diagonal samples(O(n))



[Slides from Author]

Initialization – Find the # of peaks

Vote to recover the discrete position of the large frequencies



Sparse CFT Recovery

- Recovering Frequency Coefficient
  - Use ω_i derived from **Initialization Step**
  - Solve simple linear system

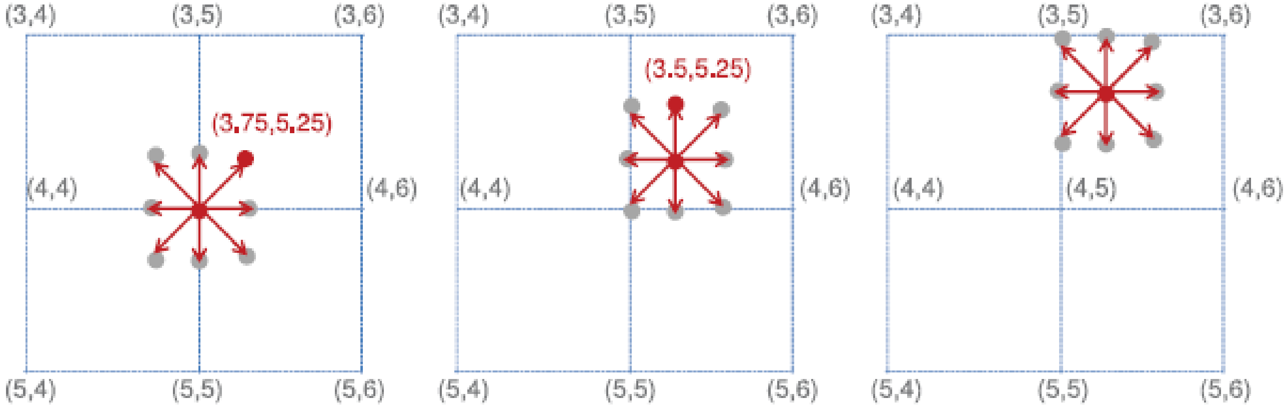
$$x(t) = \frac{1}{N}\sum_{i=0}^{k} a_i \exp\left(\frac{2\pi j t \omega_i}{N}\right)$$

Sparse CFT Recovery

- Recovering Frequency
  - Use Gradient Decent Algorithm

$$e = \sum_t \left\| x(t) - \frac{1}{N} \sum_{i=0}^{k} \tilde{a}_i \exp\left(\frac{2\pi j t \tilde{\omega}_i}{N} \right) \right\|^2$$

  - Adjust ω_i slightly until e converges



(a) iteration 1 (b) iteration 2 (c) iteration 3

Reconstruct Full Signal

- Reconstruct an image of unknown viewpoint u, v using CFT

$$L_{\omega_x,\omega_y}(u, v) = \sum_{(a,\omega_u,\omega_v)\in F} a \cdot \frac{1}{N} \exp\left(2j\pi \frac{u\omega_u + v\omega_v}{N}\right)$$

Original Captured Image

Our Reconstruction

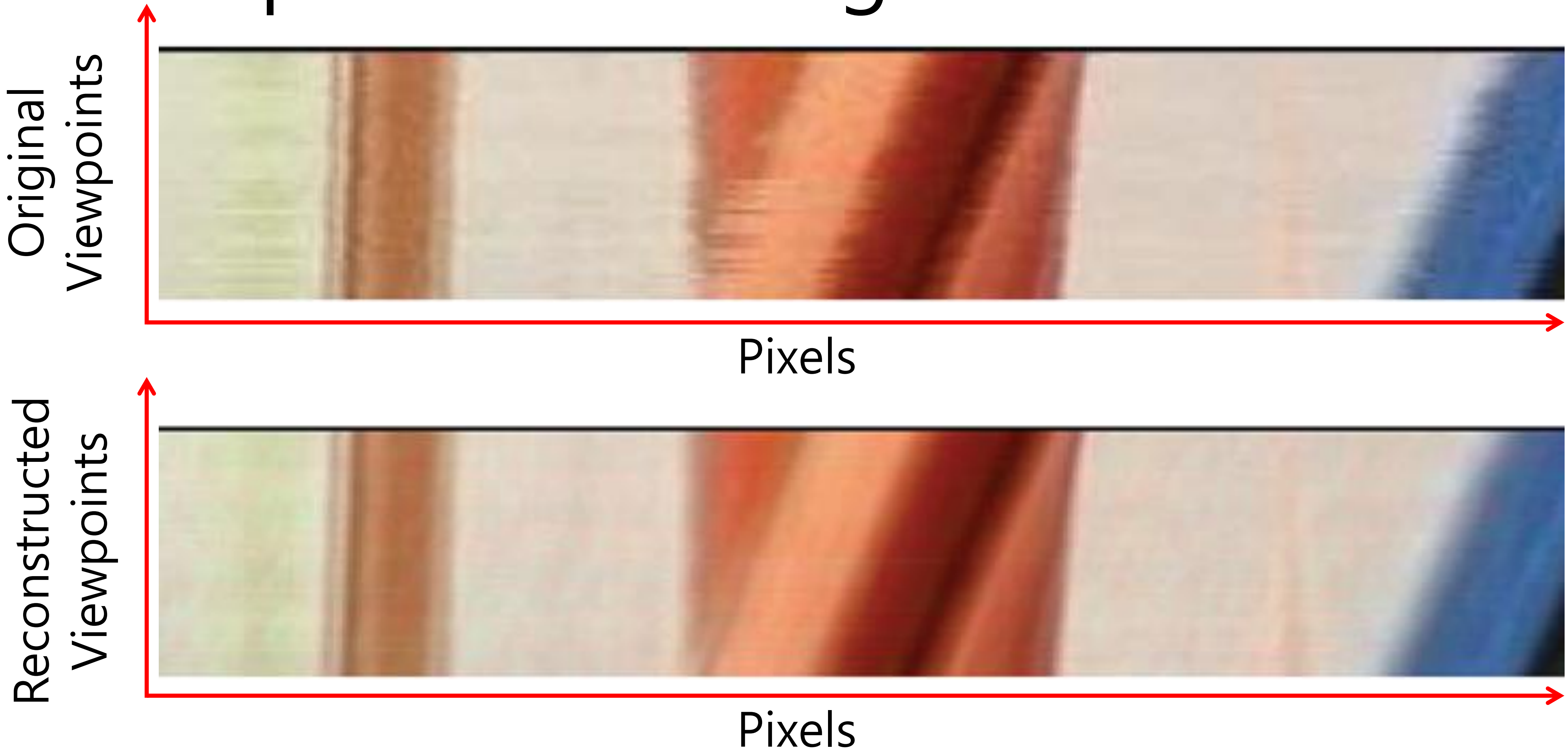
[Slides from Author]

Original Captured Image

Our Reconstruction

[Slides from Author]

Viewpoint Denoising



[Slides from Author]

Conclusion

- Sparsity in continuous vs discrete Fourier domain
- Reconstruction algorithm
  - Reduces capture cost by 70-90%
  - Non-Lambertian scenes
  - Viewpoint denoising

Thank You!