EE123 Spring 2015 Discussion Section 8

Frank Ong

Plan

- 2D DSP
- Digital Light Field Imaging

2D signal processing

- In many cases, 2D processing is **separable** => 1D processing on both sides

  – 2D Forward DFT:

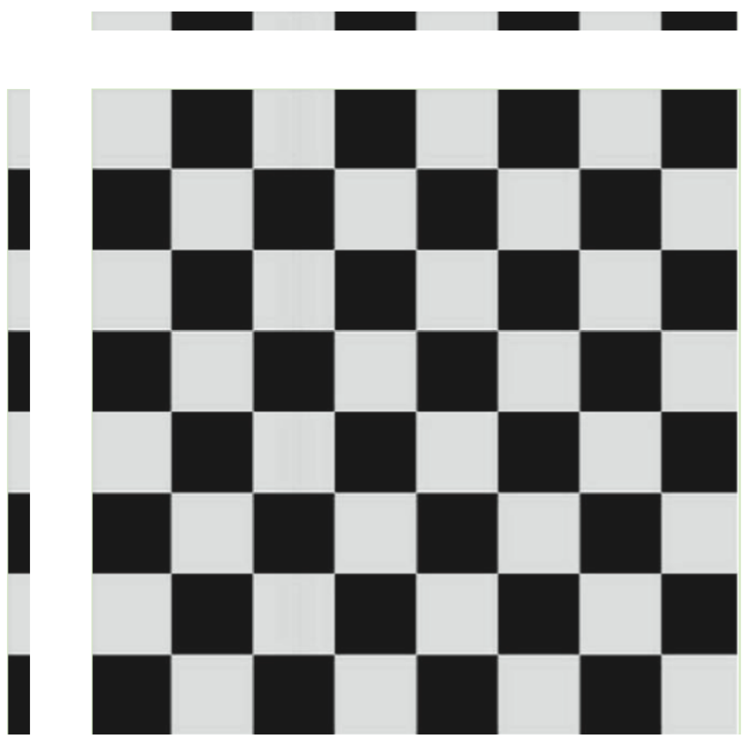
$$F[k_x, k_y] = \sum_{n_x=0}^{N-1} \sum_{n_y=0}^{M-1} f[n_x, n_y] e^{-j2\pi(n_x k_x/N + n_y k_y/M)}$$

  –2D Inverse DFT:

$$f[n_x, n_y] = \frac{1}{NM} \sum_{k_x=0}^{N-1} \sum_{k_y=0}^{M-1} F[k_x, k_y] e^{+j2\pi(n_x k_x/N + n_y k_y/M)}$$

2D signal processing

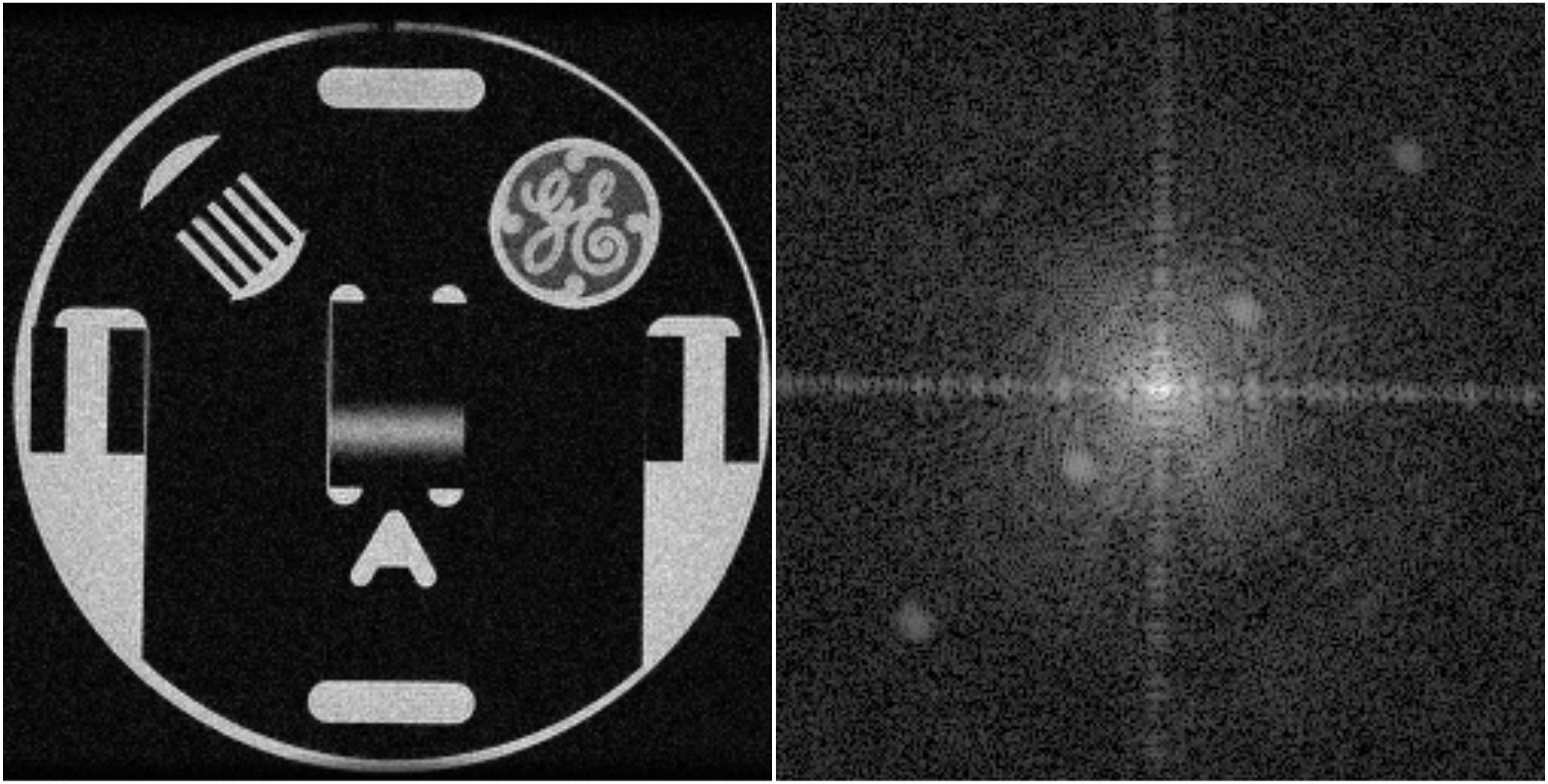
- Almost all 2D signals are **not separable**



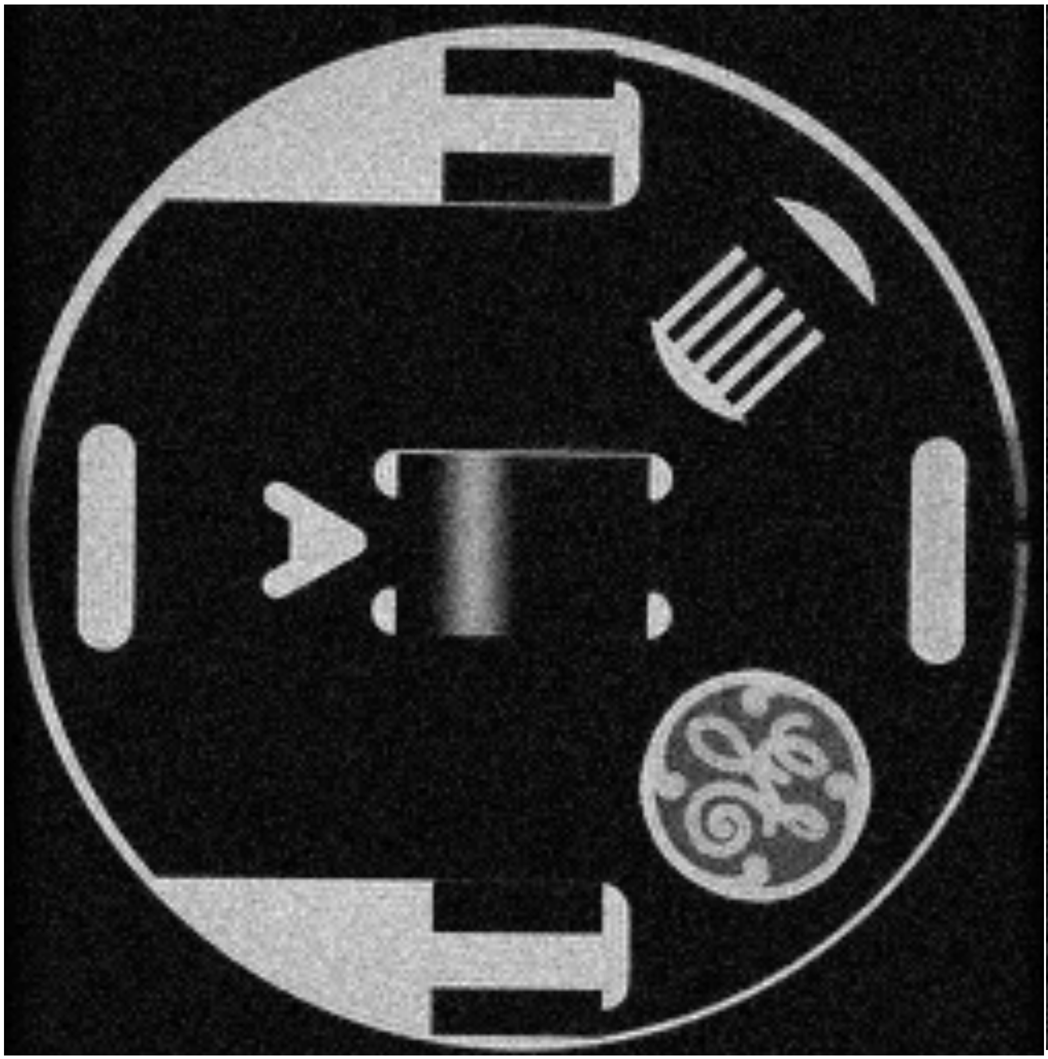
Separable image

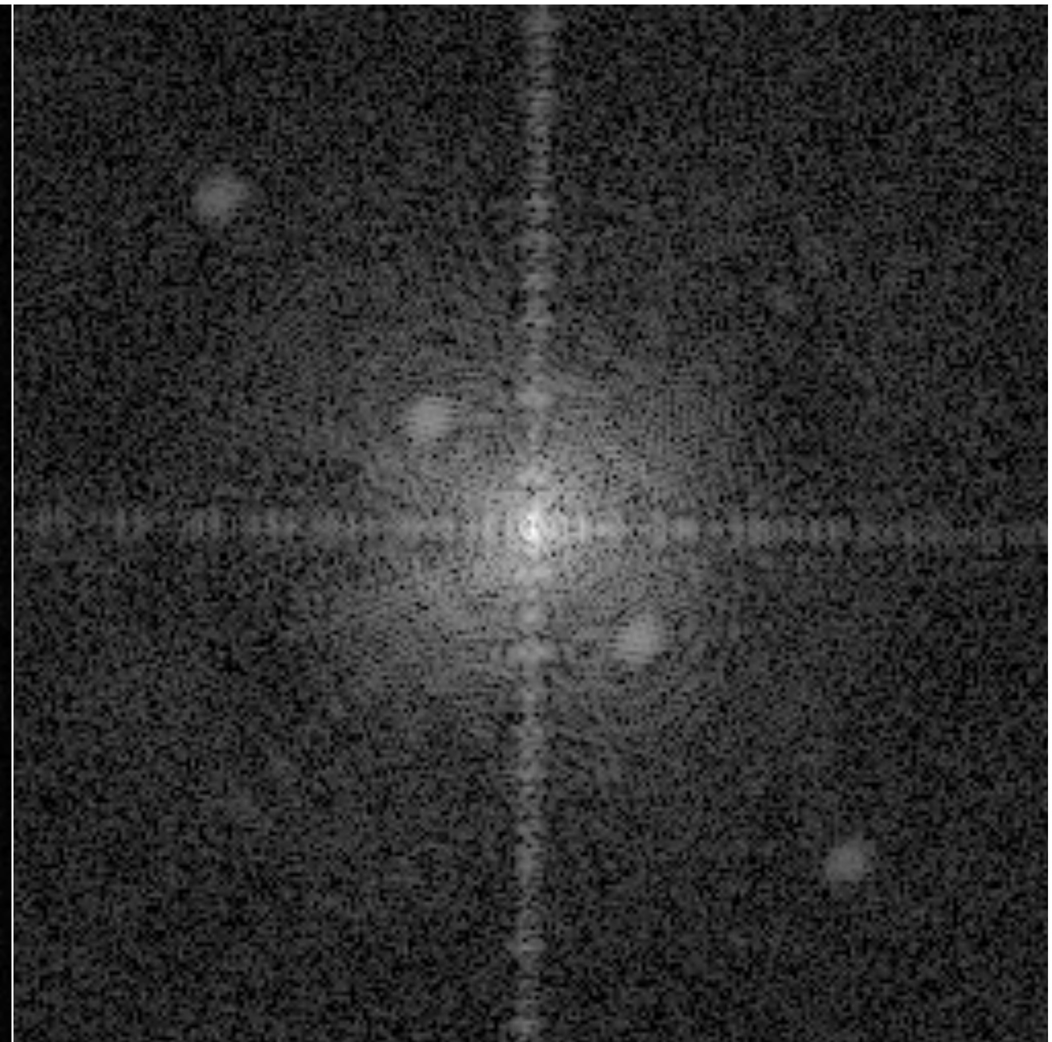
Not separable image

2D vs 1D Differences: Rotation



2D vs 1D Differences: Rotation

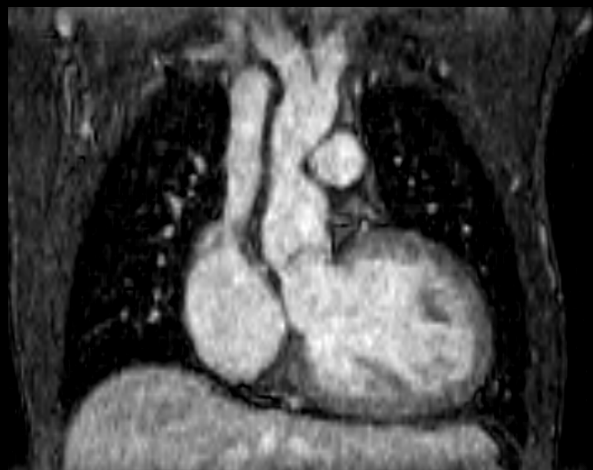


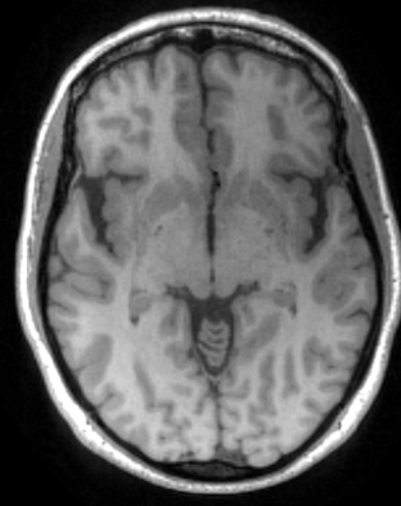


-> leads to projection slice theorem

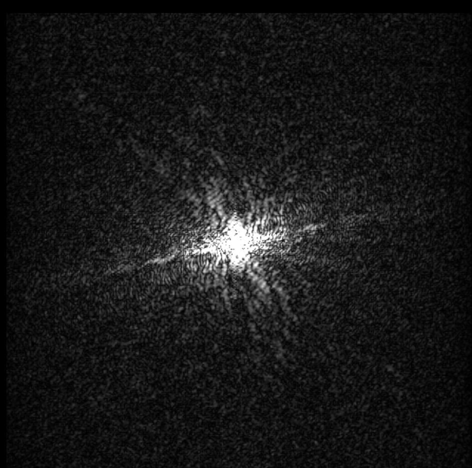
Match the images with their DFTs

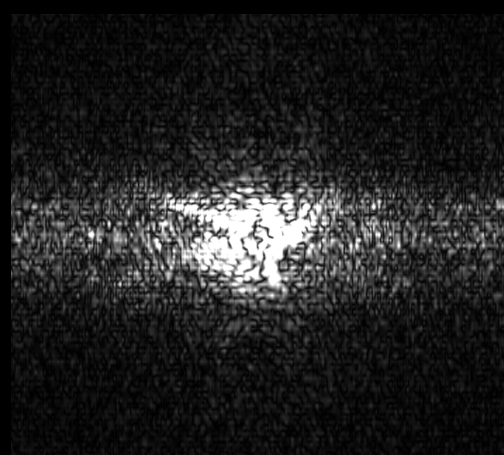
Images:

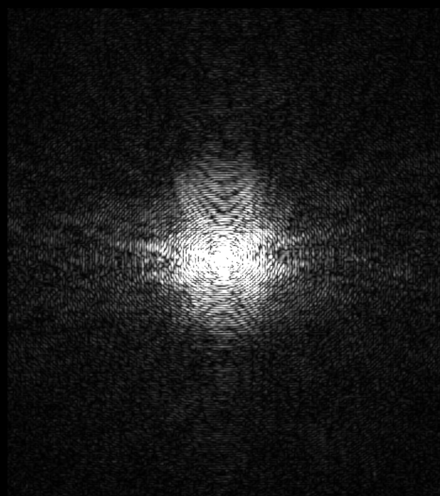




DFTs:

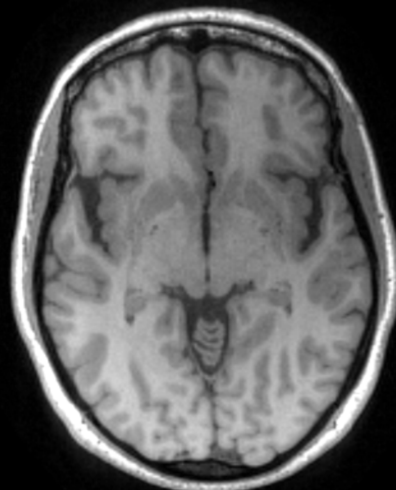




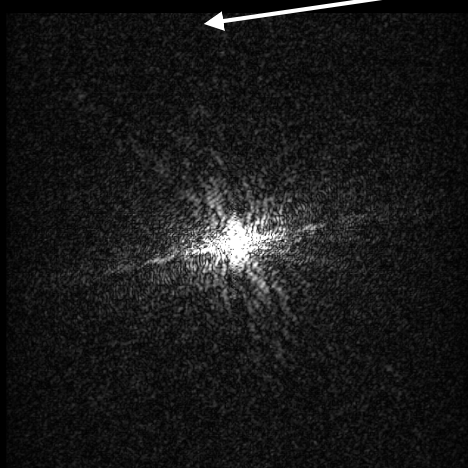


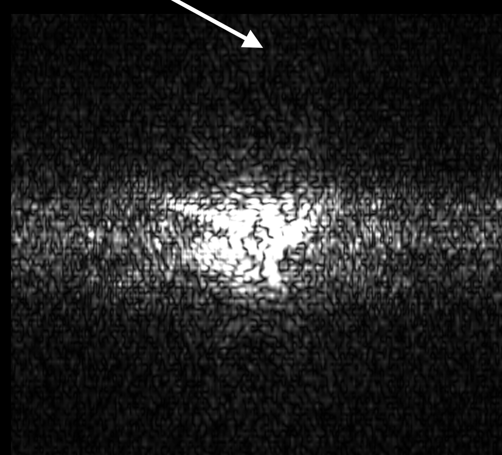
Match the images with their DFTs

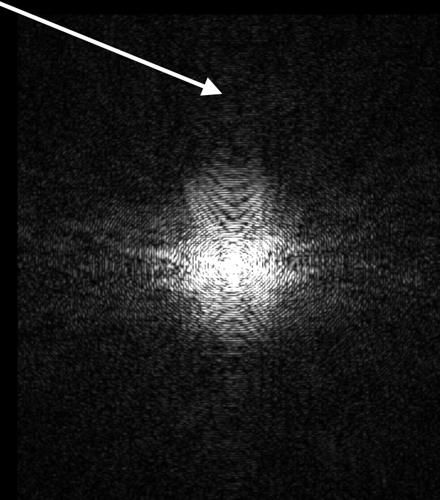
Images:



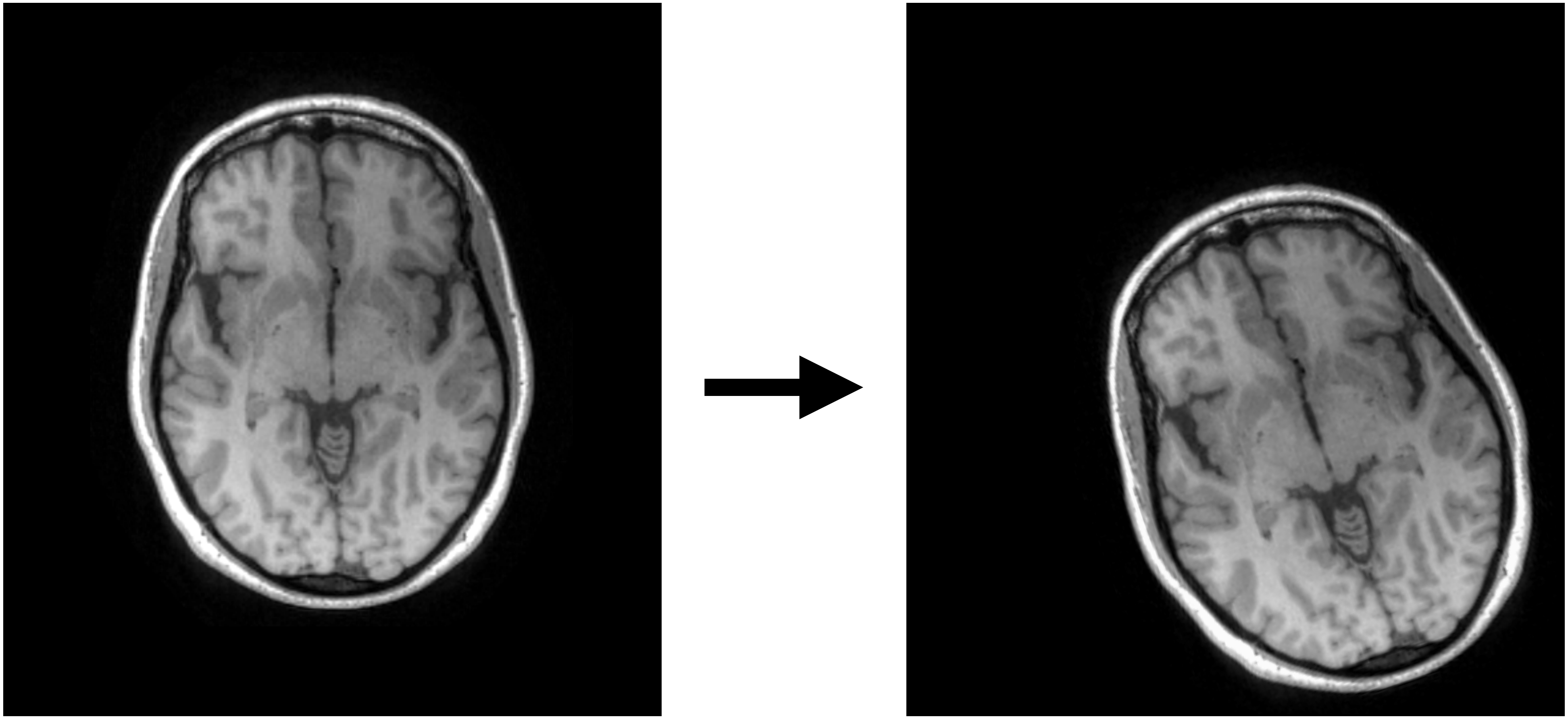
DFTs:







Rigid motion estimation

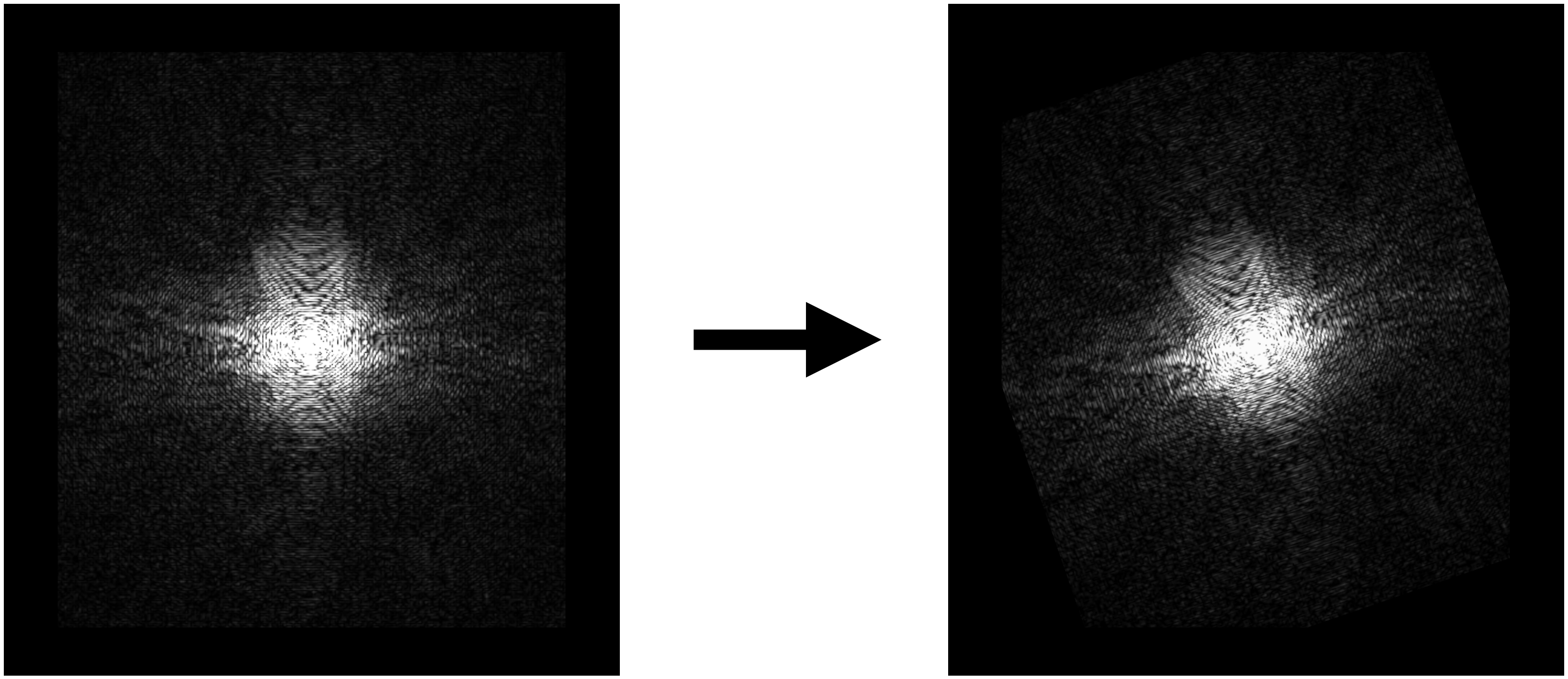


The image on the left undergoes a rigid motion transform (rotate + shift) and produces the image on the right

How can we recover the rotation angle and shift?

Rigid motion estimation

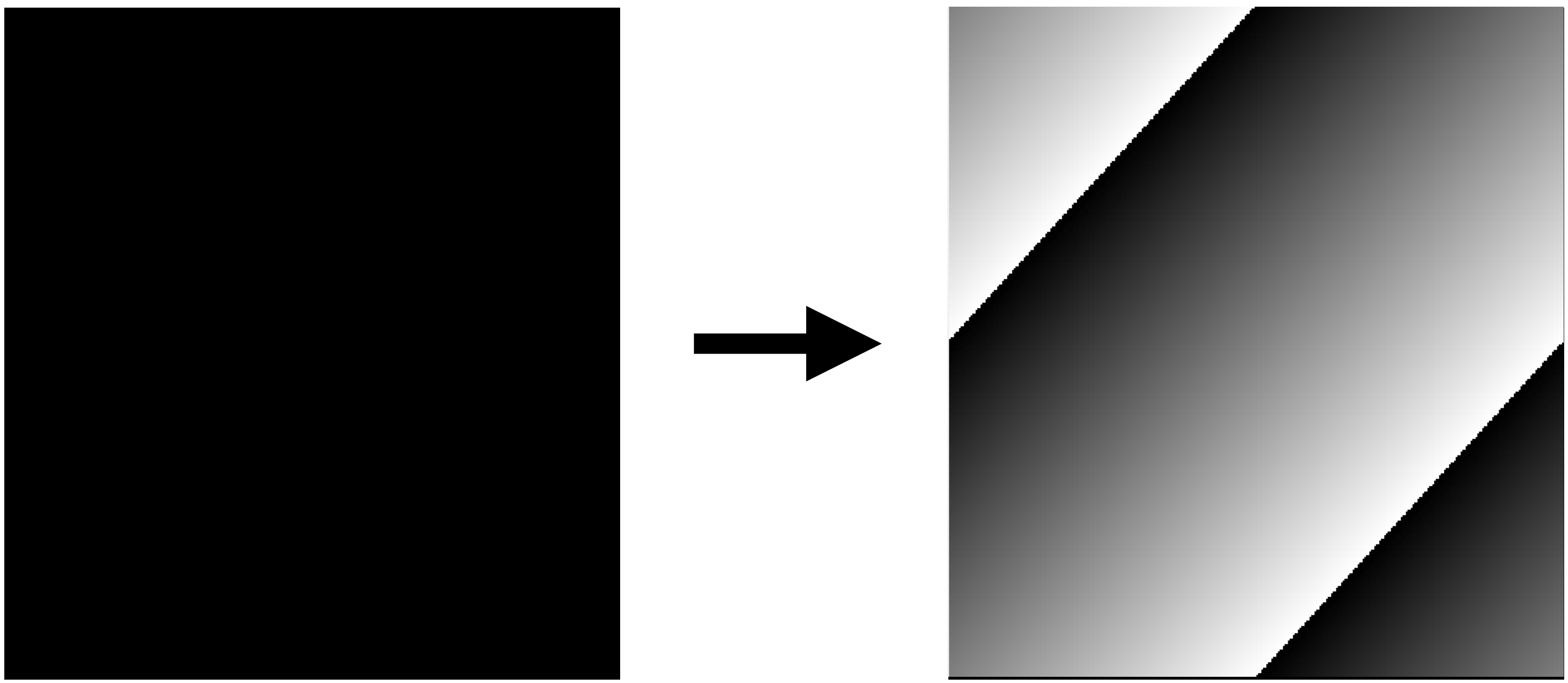
Fourier Transform



- Magnitude tells us the rotation
- Phase tells us the shift

Rigid motion estimation

Fourier Transform



- Magnitude tells us the rotation
- Phase tells us the shift

2D vs 1D Differences: Zeros-poles

- For 1D, all rational z-transforms can be factored into zeros and poles by **the fundamental theorem of algebra**

$$X(z) = \frac{N(z)}{D(z)} = \frac{\sum_{k=0}^{M} b_k z^{-k}}{\sum_{k=0}^{N} a_k z^{-k}} = \frac{b_0}{a_0} z^{-M+N} \frac{(z - z_1)(z - z_2) \cdots (z - z_M)}{(z - p_1)(z - p_2) \cdots (z - p_N)}$$

Any polynomial of degree **n** ... has **n** roots
but we may need to use complex numbers

2D vs 1D Differences: Zeros-poles

- For 2D, rational z-transforms do not have to have zeros and poles

- **The fundamental theorem for algebra does not exists for 2D!**

    - Make certain problem not possible for 2D

    - Make certain problem (Fourier phase retrieval) possible for 2D but not for 1D

Images taken from Ren Ng's thesis

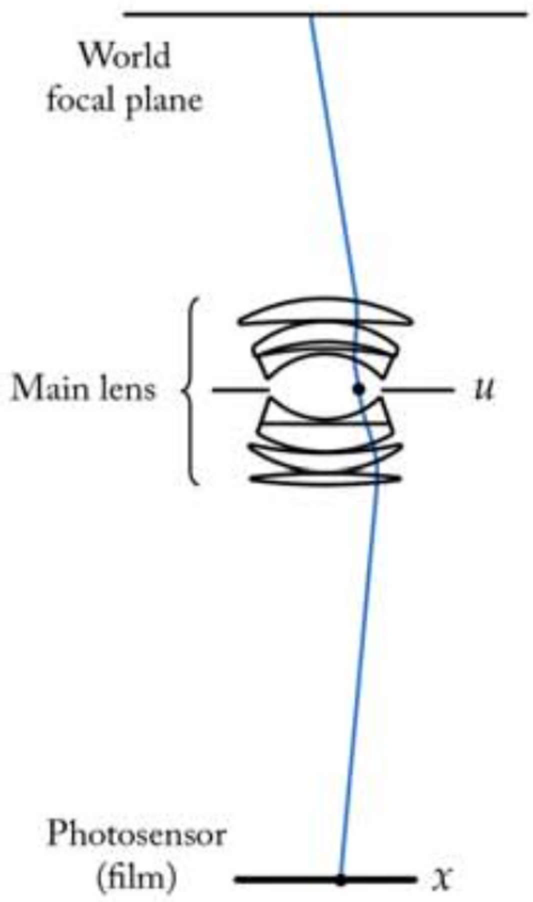
Digital Light Field Photography

- **Idea:** Refocus after you take the photo
- Uses projection slice theorem

Figure 1.3: Refocusing after the fact in digital light field photography.

Images taken from Ren Ng's thesis

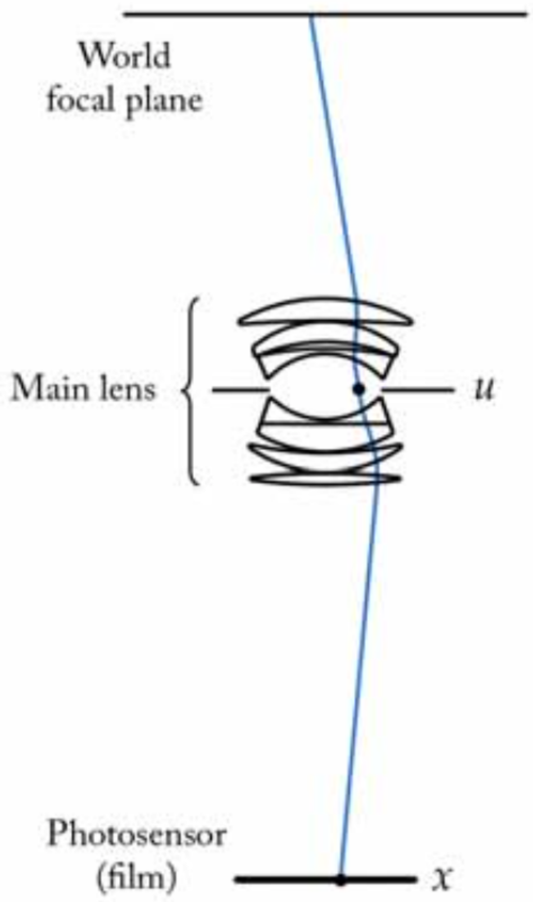
Light ray modeling

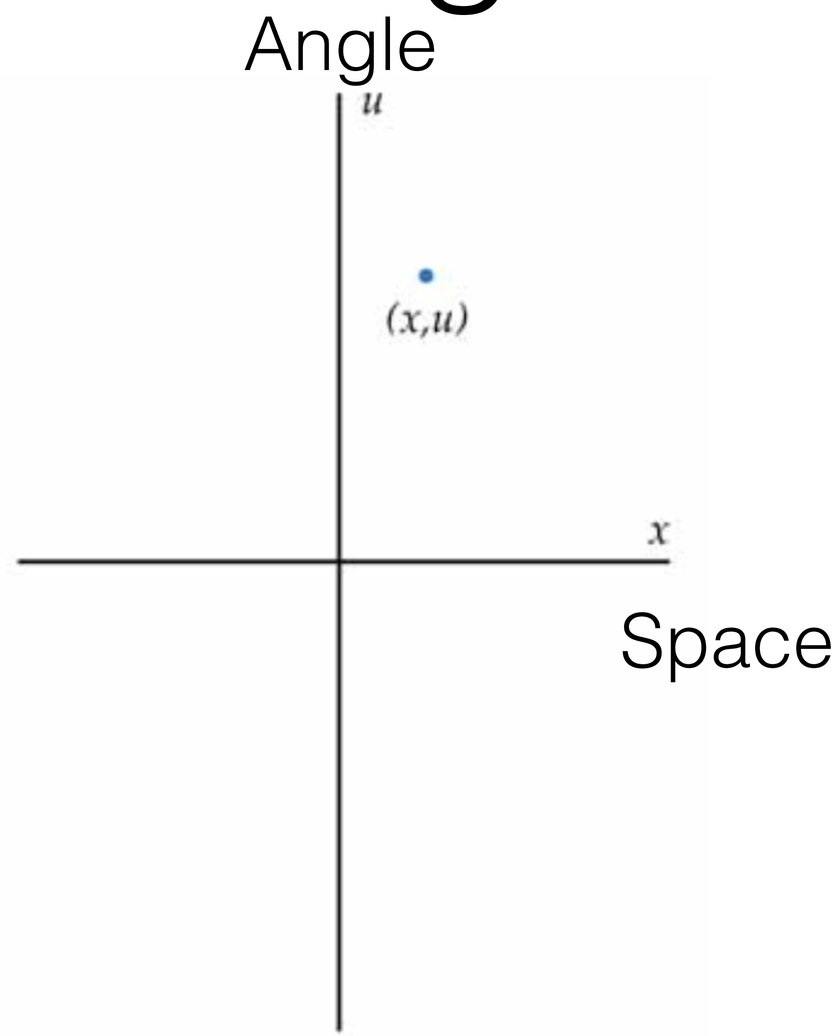


A light ray goes from one object to one pixel through a certain angle

Images taken from Ren Ng's thesis

Light ray modeling



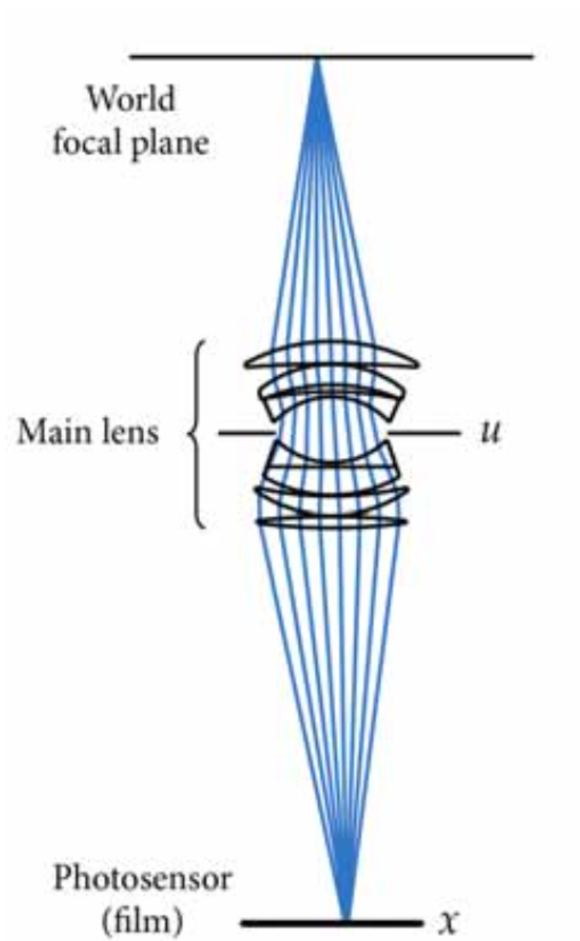


A light ray goes from one object to one pixel through a certain angle

Corresponding point on angle/space plot

Images taken from Ren Ng's thesis

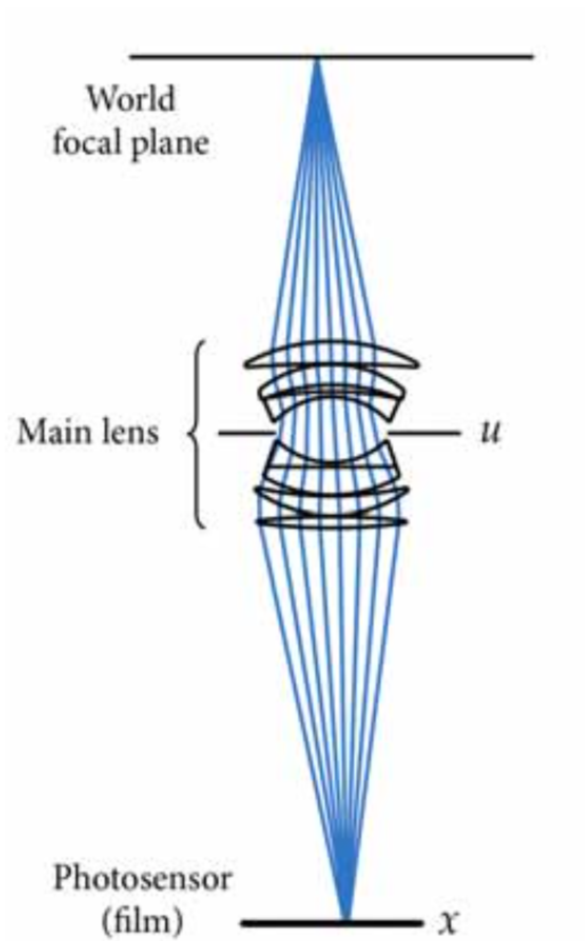
Light ray modeling



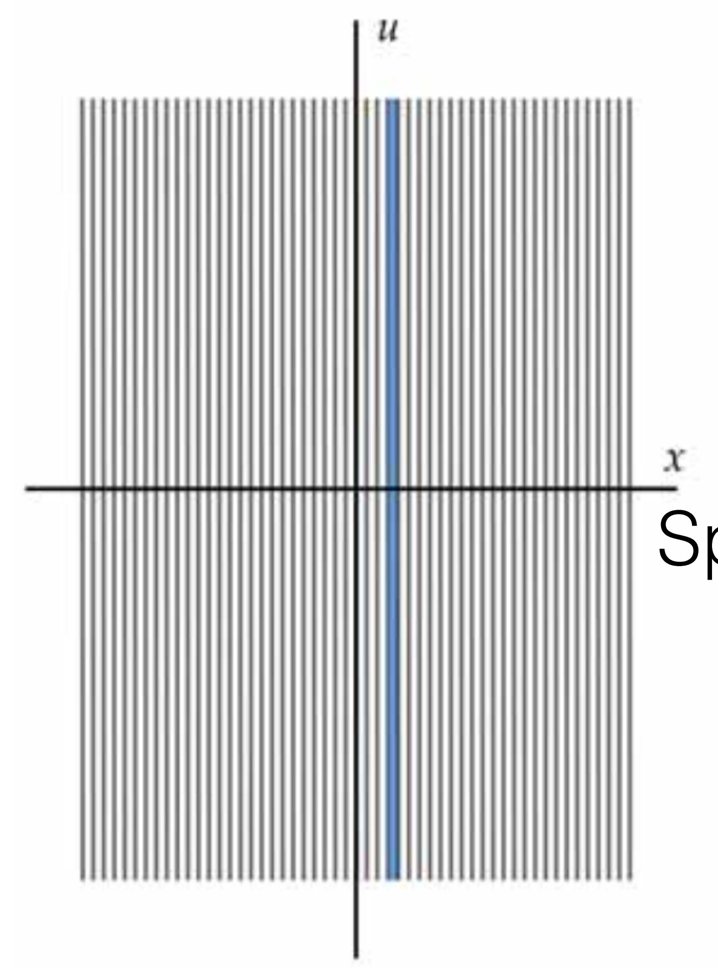
Many rays goes to one pixel
with different angles

Images taken from Ren Ng's thesis

Light ray modeling



Angle



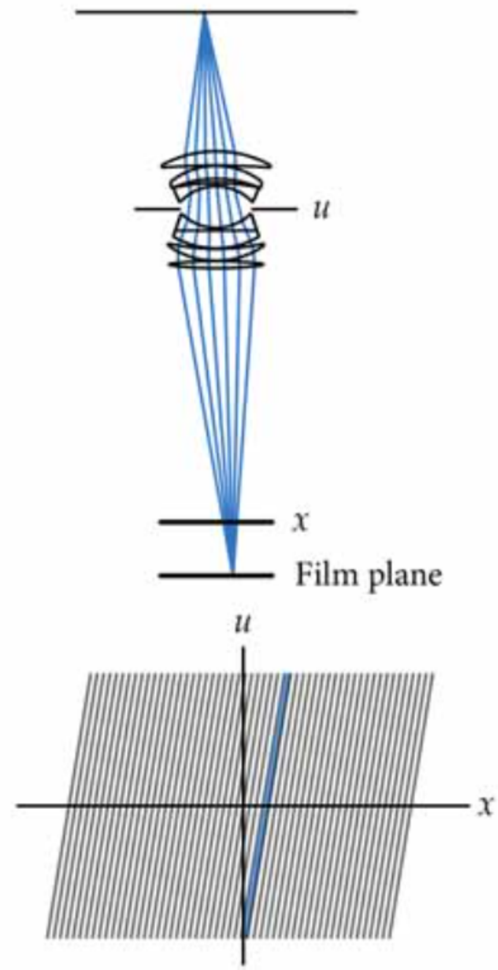
Space

Many rays goes to one pixel with different angles

Projection along angle direction !

Refocusing

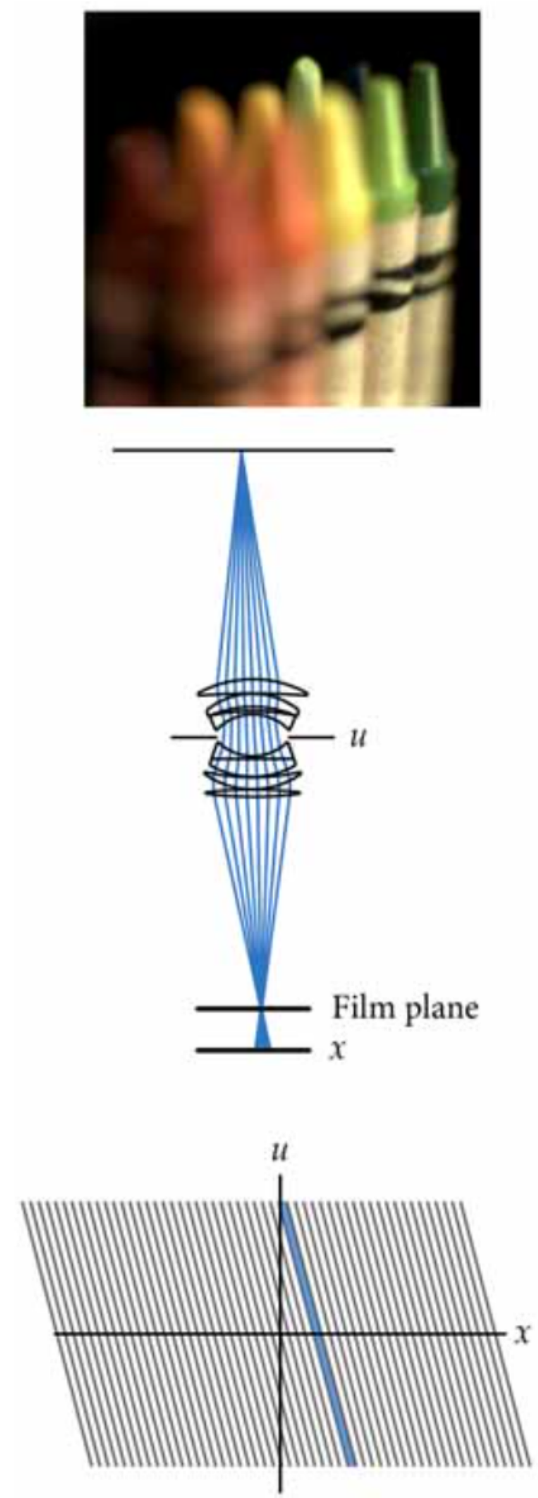
- Different focus corresponds to different tilts!



Images taken from Ren Ng's thesis

Refocusing

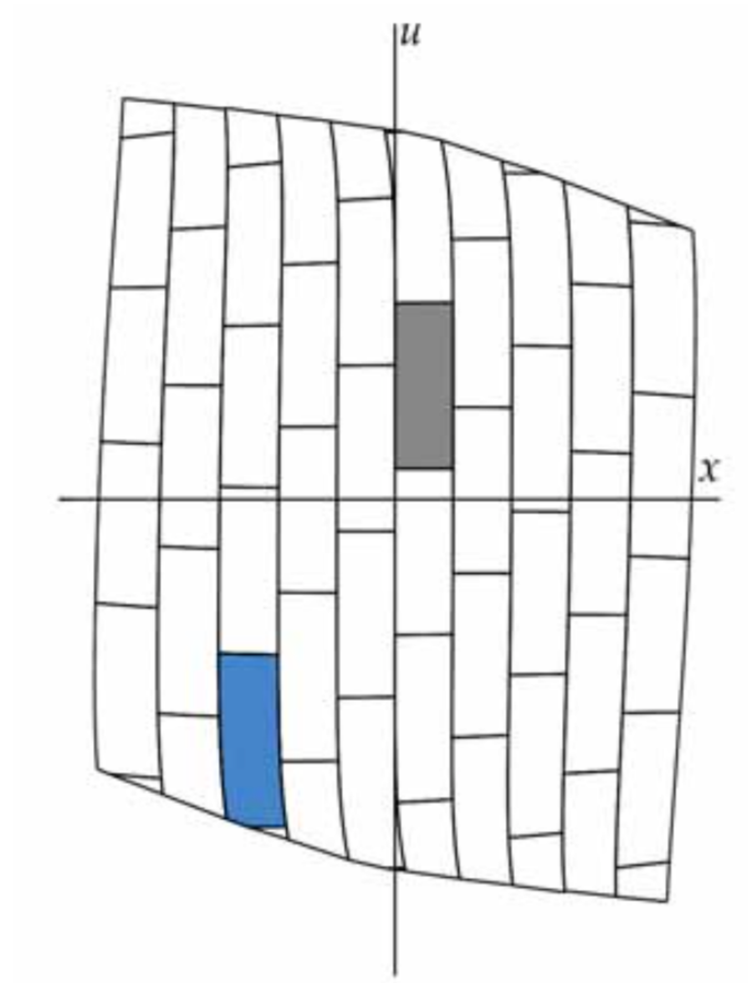
- Different focus corresponds to different tilts!



Images taken from Ren Ng's thesis

Light Field Cameras

- Captures samples in the angle/space domain
- Can refocus by doing projections along different directions



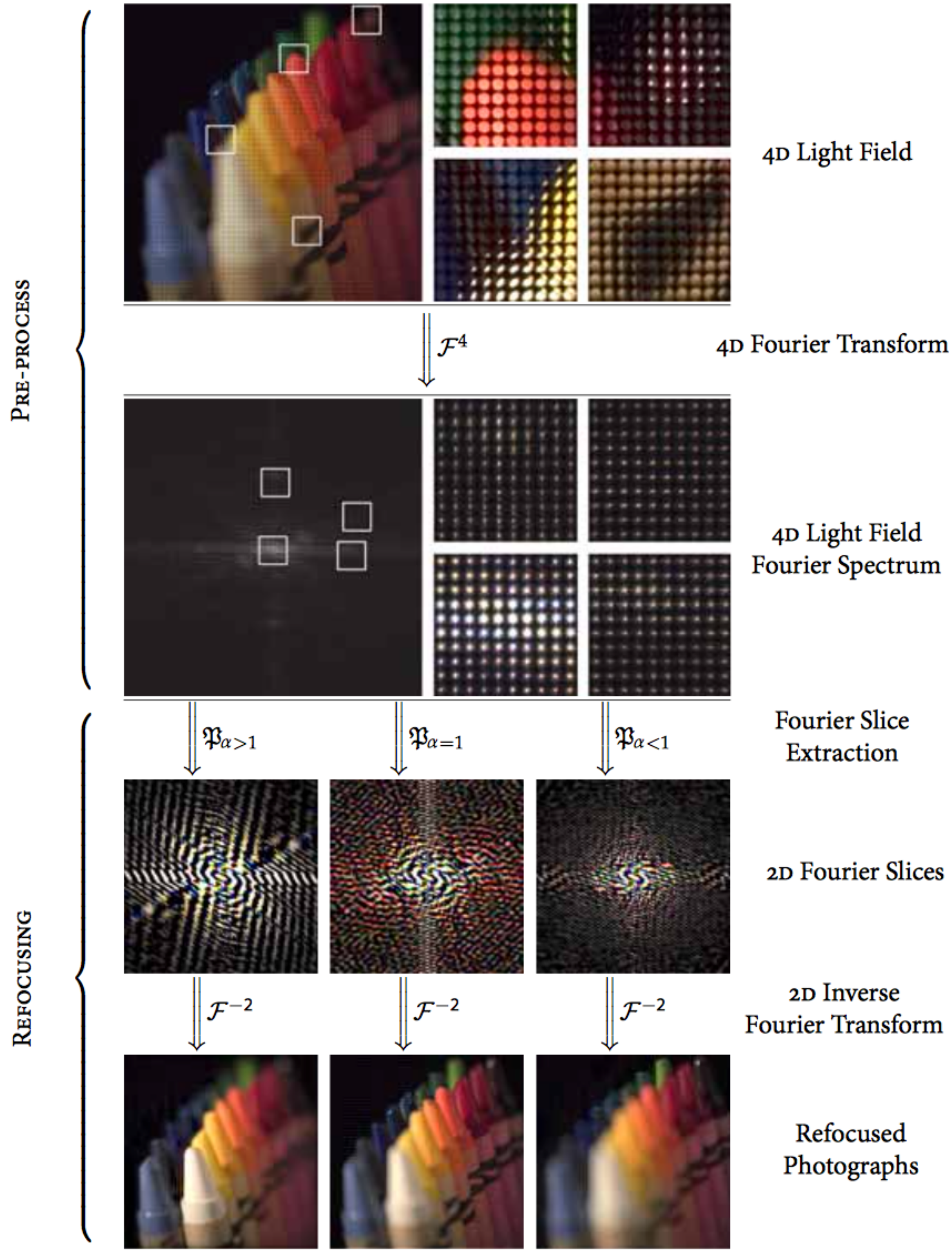
Light Field Cameras

- Captures samples in the angle/space domain
- Can refocus by doing projections along different directions

Light Field Cameras

- Captures samples in the angle/space domain
- Can refocus by doing projections along different directions

Projection slice theorem!


Pre-process
4D Light Field
$\mathcal{F}^4$
4D Fourier Transform
4D Light Field Fourier Spectrum
$\mathfrak{P}_{\alpha>1}$
$\mathfrak{P}_{\alpha=1}$
$\mathfrak{P}_{\alpha<1}$
Fourier Slice Extraction
2D Fourier Slices
$\mathcal{F}^{-2}$
$\mathcal{F}^{-2}$
$\mathcal{F}^{-2}$
2D Inverse Fourier Transform
Refocused Photographs
Refocusing