

# Fast Multidimensional Signal Processing with Shearlab.jl

Héctor Andrade Loarca

TU Berlin, BMS

2<sup>nd</sup> of July, 2017



# What is a signal?

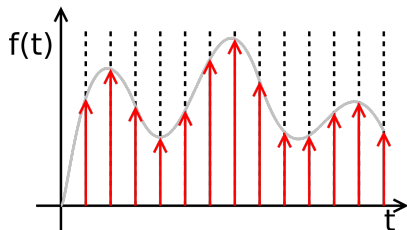
## Our definition

Function (or something that can be represented as) that contains information about the behavior or attributes of some phenomenon. It can be digital (discrete) or analog (continuous).

# What is a signal?

## Our definition

Function (or something that can be represented as) that contains information about the behavior or attributes of some phenomenon. It can be digital (discrete) or analog (continuous).



**Figure:** Digital and continuous one-dimensional signals

# What is a signal?

## Our definition

Function (or something that can be represented as) that contains information about the behavior or attributes of some phenomenon. It can be digital (discrete) or analog (continuous).

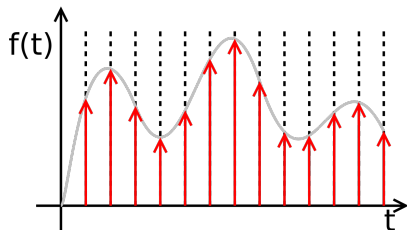


Figure: Digital and continuous one-dimensional signals

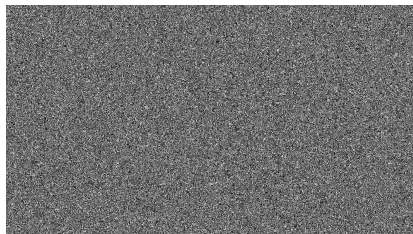


Figure: White noise, not a signal

# Sparse representations of signals

- ▶ Relevant information in structured data is sparse, due the high correlation of its elements.

# Sparse representations of signals

- ▶ Relevant information in structured data is sparse, due the high correlation of its elements.
- ▶ Goal: Find the right dictionary to represent optimally our data.

The diagram illustrates the equation  $x = D \hat{\alpha}$ . On the left, a vertical blue bar represents the signal  $x$ . This is equal to a matrix multiplication. The matrix is the Sparse Dictionary  $D$ , shown as a collection of vertical bars of different colors (red, blue, yellow, and green) with ellipses indicating more columns. This matrix is multiplied by a vertical vector of boxes representing the Sparse Coefficients  $\hat{\alpha}$ . The vector has colored boxes (red, blue, yellow, and green) corresponding to the columns in the dictionary, with ellipses indicating more coefficients. The red, blue, and yellow boxes are highlighted with colored outlines, indicating non-zero coefficients.

$x = \begin{bmatrix} \text{red} & \text{red} & \text{red} & \text{red} & \text{red} & \text{red} & \text{blue} & \text{blue} & \text{blue} & \text{blue} & \text{yellow} & \text{yellow} & \dots & \text{green} & \text{green} & \text{green} & \text{green} \end{bmatrix} \cdot \begin{bmatrix} \text{red} \\ \text{red} \\ \text{red} \\ \text{red} \\ \text{red} \\ \text{red} \\ \text{blue} \\ \text{blue} \\ \text{blue} \\ \text{blue} \\ \text{yellow} \\ \text{yellow} \\ \dots \\ \text{green} \\ \text{green} \\ \text{green} \\ \text{green} \end{bmatrix}$

Sparse Dictionary  $D$

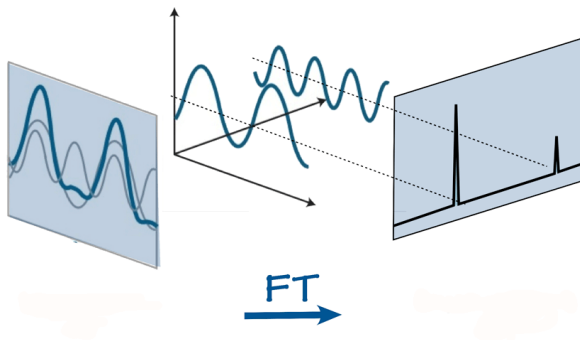
Sparse Coefficients  $\hat{\alpha}$

# Fourier Transform (Fourier, 1822)

$$\hat{f}(\omega) := \int_{\mathbb{R}^n} f(x) e^{-i\langle x, \omega \rangle} dx$$

# Fourier Transform (Fourier, 1822)

$$\hat{f}(\omega) := \int_{\mathbb{R}^n} f(x) e^{-i\langle x, \omega \rangle} dx$$



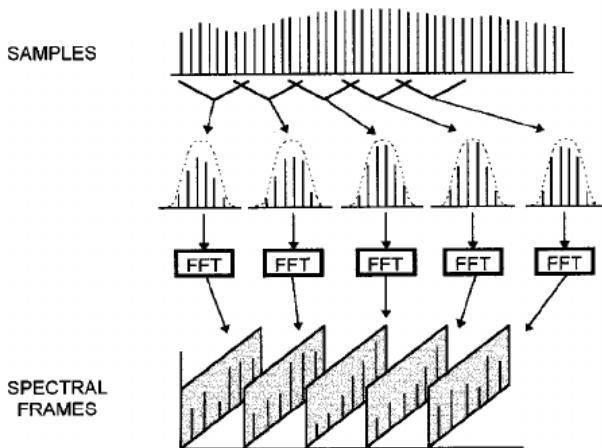


# Short Time Fourier Transform (Gabor, 1946)

$$S_g f(t, \omega) = \int_{\mathbb{R}} f(x) \overline{g(x-t)} e^{-ix\omega} dx$$

# Short Time Fourier Transform (Gabor,1946)

$$S_g f(t, \omega) = \int_{\mathbb{R}} f(x) \overline{g(x-t)} e^{-ix\omega} dx$$



# Wavelet Transform (Morlet and Grossman, 1984)

$$\begin{aligned}\mathcal{W}_\psi f(a, b) &= \int_{\mathbb{R}} f(t) a^{-\frac{1}{2}} \overline{\psi\left(\frac{t-b}{a}\right)} dt \\ &= (f * D_a \bar{\psi}^*)(b), \quad (a, b) \in \mathbb{R}^+ \times \mathbb{R}\end{aligned}$$

where

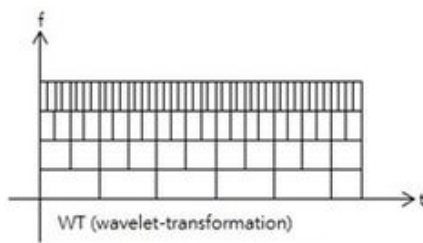
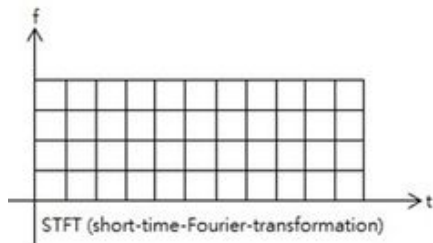
$$\int_0^\infty \frac{|\hat{\psi}(\omega)|^2}{\omega} d\omega < \infty$$

# Wavelet Transform (Morlet and Grossman, 1984)

$$\begin{aligned}\mathcal{W}_\psi f(a, b) &= \int_{\mathbb{R}} f(t) a^{-\frac{1}{2}} \overline{\psi\left(\frac{t-b}{a}\right)} dt \\ &= (f * D_a \overline{\psi^*})(b), \quad (a, b) \in \mathbb{R}^+ \times \mathbb{R}\end{aligned}$$

where

$$\int_0^\infty \frac{|\hat{\psi}(\omega)|^2}{\omega} d\omega < \infty$$



# Cartoon-like functions

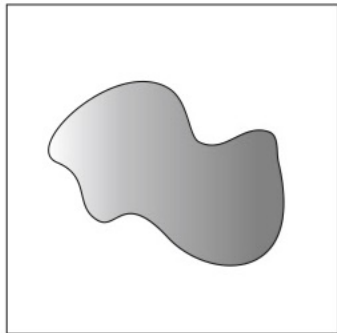
## Definition

Let  $f : \mathbb{R}^2 \rightarrow \mathbb{C}$ ,  $f \in \mathcal{E}^2(\mathbb{R}^2)$  if  $f = f_0 + \chi_B f_1$ , with  $B \subset [0, 1]^2$ ,  $\partial B \in C^2$  and with bounded curvature. Moreover,  $f_i \in C^2(\mathbb{R}^2)$  with  $\|f_i\|_{C^2} \leq 1$  and  $\text{supp} f_i \subset [0, 1]^2$  for  $i = 0, 1$ .

# Cartoon-like functions

## Definition

Let  $f : \mathbb{R}^2 \rightarrow \mathbb{C}$ ,  $f \in \mathcal{E}^2(\mathbb{R}^2)$  if  $f = f_0 + \chi_B f_1$ , with  $B \subset [0, 1]^2$ ,  $\partial B \in C^2$  and with bounded curvature. Moreover,  $f_i \in C^2(\mathbb{R}^2)$  with  $\|f_i\|_{C^2} \leq 1$  and  $\text{supp} f_i \subset [0, 1]^2$  for  $i = 0, 1$ .



# Optimal error for 2D signals

## Best N-term approx. error (Donoho, 2001)

Let  $\{\psi_\lambda\}_{\lambda \in \Lambda} \subset L^2(\mathbb{R}^2)$  a frame. The optimal best N-Term approximation error for any  $f \in \mathcal{R}^2(\mathbb{R}^2)$  is

$$\sigma_N(f, \{\psi_\lambda\}_{\lambda \in \Lambda}) = O(N^{-1})$$

# Optimal error for 2D signals

## Best N-term approx. error (Donoho, 2001)

Let  $\{\psi_\lambda\}_{\lambda \in \Lambda} \subset L^2(\mathbb{R}^2)$  a frame. The optimal best N-Term approximation error for any  $f \in \mathcal{R}^2(\mathbb{R}^2)$  is

$$\sigma_N(f, \{\psi_\lambda\}_{\lambda \in \Lambda}) = O(N^{-1})$$

## Error of 2D-wavelets

$$\sigma_N(f, \{\psi_{j,m}\}_{j,m}) \sim N^{-1/2}$$



# Optimal error for 2D signals

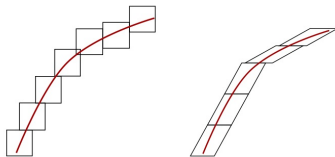
## Best N-term approx. error (Donoho, 2001)

Let  $\{\psi_\lambda\}_{\lambda \in \Lambda} \subset L^2(\mathbb{R}^2)$  a frame. The optimal best N-Term approximation error for any  $f \in \mathcal{R}^2(\mathbb{R}^2)$  is

$$\sigma_N(f, \{\psi_\lambda\}_{\lambda \in \Lambda}) = O(N^{-1})$$

## Error of 2D-wavelets

$$\sigma_N(f, \{\psi_{j,m}\}_{j,m}) \sim N^{-1/2}$$



# How to solve this? Scaling and Shearing

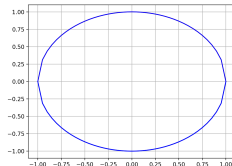
- *Scaling*

$$A_j := \begin{pmatrix} 2^j & 0 \\ 0 & 2^{j/2} \end{pmatrix}$$

# How to solve this? Scaling and Shearing

## ► *Scaling*

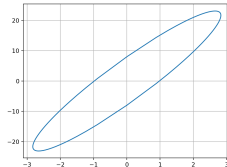
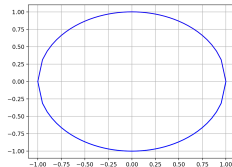
$$A_j := \begin{pmatrix} 2^j & 0 \\ 0 & 2^{j/2} \end{pmatrix}$$



# How to solve this? Scaling and Shearing

## ► *Scaling*

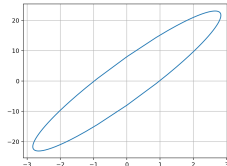
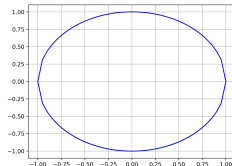
$$A_j := \begin{pmatrix} 2^j & 0 \\ 0 & 2^{j/2} \end{pmatrix}$$



# How to solve this? Scaling and Shearing

## ► *Scaling*

$$A_j := \begin{pmatrix} 2^j & 0 \\ 0 & 2^{j/2} \end{pmatrix}$$



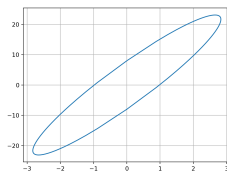
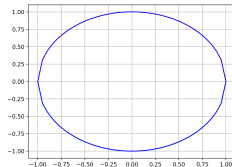
## ► *Shearing*

$$S_k := \begin{pmatrix} 1 & k \\ 0 & 1 \end{pmatrix}$$

# How to solve this? Scaling and Shearing

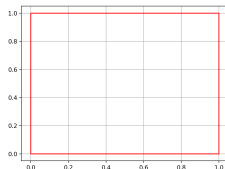
## ► *Scaling*

$$A_j := \begin{pmatrix} 2^j & 0 \\ 0 & 2^{j/2} \end{pmatrix}$$



## ► *Shearing*

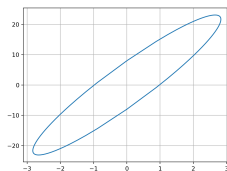
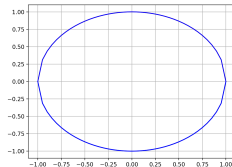
$$S_k := \begin{pmatrix} 1 & k \\ 0 & 1 \end{pmatrix}$$



# How to solve this? Scaling and Shearing

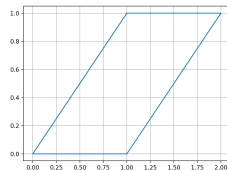
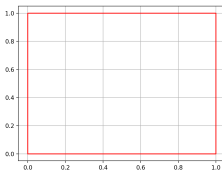
## ► *Scaling*

$$A_j := \begin{pmatrix} 2^j & 0 \\ 0 & 2^{j/2} \end{pmatrix}$$



## ► *Shearing*

$$S_k := \begin{pmatrix} 1 & k \\ 0 & 1 \end{pmatrix}$$



## Classical Shearlet Transform

$$\langle f, \psi_{j,k,m} \rangle = \int_{\mathbb{R}^2} f(x) \overline{\psi_{j,k,m}(x)} dx$$



# Shearlet Transform (Kutyniok, Guo, Labate, 2005)

## Classical Shearlet Transform

$$\langle f, \psi_{j,k,m} \rangle = \int_{\mathbb{R}^2} f(x) \overline{\psi_{j,k,m}(x)} dx$$

where

$$\mathcal{SH}(\psi) = \{\psi_{j,k,m}(x) = 2^{3j/4} \psi(S_k A_j x - m) : (j, k) \in \mathbb{Z}^2, m \in \mathbb{Z}^2\}$$

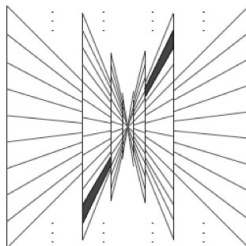
# Shearlet Transform (Kutyniok, Guo, Labate, 2005)

## Classical Shearlet Transform

$$\langle f, \psi_{j,k,m} \rangle = \int_{\mathbb{R}^2} f(x) \overline{\psi_{j,k,m}(x)} dx$$

where

$$\mathcal{SH}(\psi) = \{\psi_{j,k,m}(x) = 2^{3j/4} \psi(S_k A_j x - m) : (j, k) \in \mathbb{Z}^2, m \in \mathbb{Z}^2\}$$

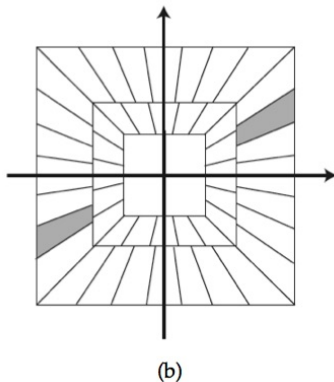
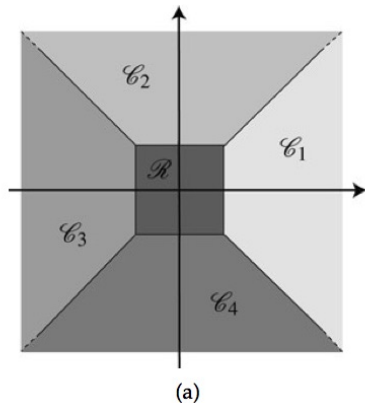


# Cone-based shearlet transform

$$\mathcal{SH}(\phi, \psi, \tilde{\psi}, c) := \mathcal{P}_{\mathcal{R}}\Phi(\phi, c1) \cup \mathcal{P}_{\mathcal{C}_1}\Psi(\psi, c) \cup \mathcal{P}_{\mathcal{C}_2}\tilde{\Psi}(\tilde{\psi}, c)$$

# Cone-based shearlet transform

$$\mathcal{SH}(\phi, \psi, \tilde{\psi}, c) := \mathcal{P}_{\mathcal{R}}\Phi(\phi, c1) \cup \mathcal{P}_{\mathcal{C}_1}\Psi(\psi, c) \cup \mathcal{P}_{\mathcal{C}_2}\tilde{\Psi}(\tilde{\psi}, c)$$



# Separable and non-separable generator

- ▶ *Separable*

$$\psi^{\text{sep}}(x_1, x_2) = \psi_1(x_1)\phi_1(x_2)$$

# Separable and non-separable generator

## ► *Separable*

$$\psi^{\text{sep}}(x_1, x_2) = \psi_1(x_1)\phi_1(x_2)$$

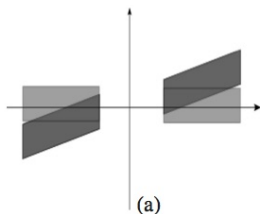
## ► *Non-separable*

$$\hat{\psi}^{\text{non}}(\xi) = P\left(\frac{\xi_1}{2}, \xi_2\right) \hat{\psi}^{\text{sep}}(\xi)$$

# Separable and non-separable generator

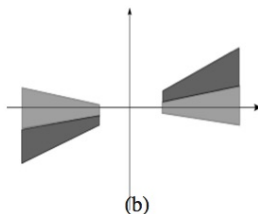
## ► *Separable*

$$\psi^{\text{sep}}(x_1, x_2) = \psi_1(x_1)\phi_1(x_2)$$



## ► *Non-separable*

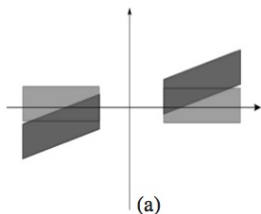
$$\hat{\psi}^{\text{non}}(\xi) = P\left(\frac{\xi_1}{2}, \xi_2\right) \hat{\psi}^{\text{sep}}(\xi)$$



# Separable and non-separable generator

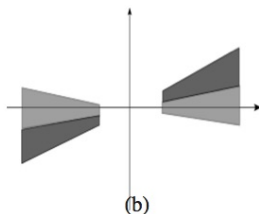
## ► Separable

$$\psi^{\text{sep}}(x_1, x_2) = \psi_1(x_1)\phi_1(x_2)$$



## ► Non-separable

$$\hat{\psi}^{\text{non}}(\xi) = P\left(\frac{\xi_1}{2}, \xi_2\right) \hat{\psi}^{\text{sep}}(\xi)$$



## ► Best $N$ -term approximation error

$$\sigma_N(f, \{\psi_{j,k,m}\}_{j,k,m}) \sim N^{-1}(\log(N))^{3/2}$$



## ► Matlab

- FFST- Fast Finite Shearlet Transform (Häuser, Steidl, TU Keiserlautern)  
<http://www.mathematik.uni-kl.de/imagepro/software/ffst/>
- 2D/3D Shearlet Toolbox (D. Labate, University of Houston)  
<https://www.math.uh.edu/~dlabate/software.html>
- **Shearlab3D** (G. Kutyniok, W.-Q.Lim, R. Reisenhoffer, TU Berlin)  
<http://www.shearlab.org/>

## ► Matlab

- FFST- Fast Finite Shearlet Transform (Häuser, Steidl, TU Keiserlautern)  
<http://www.mathematik.uni-kl.de/imagepro/software/ffst/>
- 2D/3D Shearlet Toolbox (D. Labate, University of Houston)  
<https://www.math.uh.edu/~dlabate/software.html>
- **Shearlab3D** (G. Kutyniok, W.-Q.Lim, R. Reisenhofer, TU Berlin)  
<http://www.shearlab.org/>

## ► Python

- pyShearLab (Stefan Loock, U Göttingen)  
<http://na.math.uni-goettingen.de/pyshearlab/>

## ▶ Matlab

- ▶ FFST- Fast Finite Shearlet Transform (Häuser, Steidl, TU Keiserlautern)  
<http://www.mathematik.uni-kl.de/imagepro/software/ffst/>
- ▶ 2D/3D Shearlet Toolbox (D. Labate, University of Houston)  
<https://www.math.uh.edu/~dlabate/software.html>
- ▶ **Shearlab3D** (G. Kutyniok, W.-Q.Lim, R. Reisenhofer, TU Berlin)  
<http://www.shearlab.org/>

## ▶ Python

- ▶ pyShearLab (Stefan Loock, U Göttingen)  
<http://na.math.uni-goettingen.de/pyshearlab/>

## ▶ Julia

- ▶ **Shearlab.jl** (H. Andrade, TU Berlin)  
<https://github.com/arsenal9971/Shearlab.jl>

# Why Julia?

- ▶ Extensive use of `fft`, well implemented in Julia.

# Why Julia?

- ▶ Extensive use of `fft` , well implemented in Julia.
- ▶ Fast vectorization and loops as well as JIT-compilation.

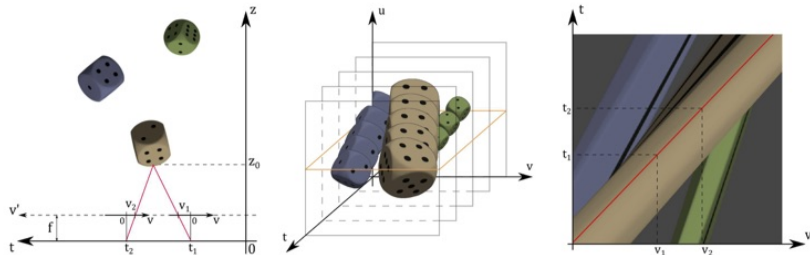
# Why Julia?

- ▶ Extensive use of `fft` , well implemented in Julia.
- ▶ Fast vectorization and loops as well as JIT-compilation.
- ▶ Plenty of image filtering, import and rescaling functions with `Images.jl` , `Wavelets.jl` .

# Why Julia?

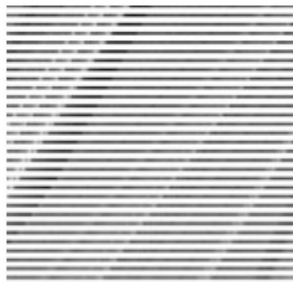
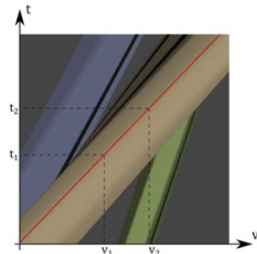
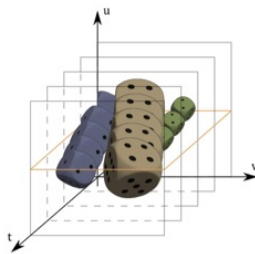
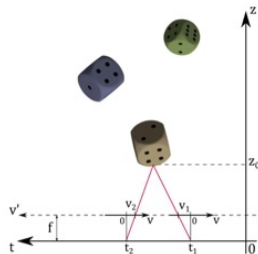
- ▶ Extensive use of `fft` , well implemented in Julia.
- ▶ Fast vectorization and loops as well as JIT-compilation.
- ▶ Plenty of image filtering, import and rescaling functions with `Images.jl` , `Wavelets.jl` .
- ▶ Support of multithreading and painless GPU processing with `ArrayFire.jl` .

## Current application: Light Field Recovery

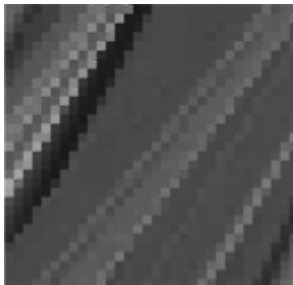
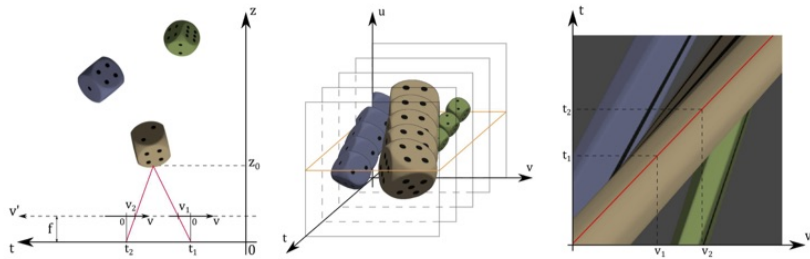




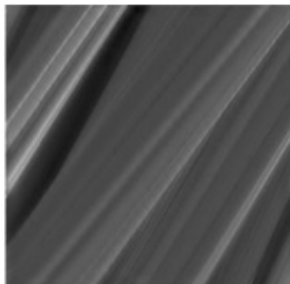
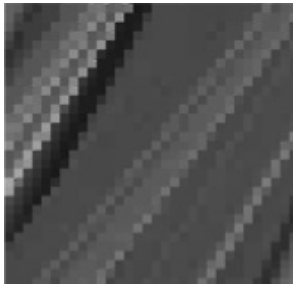
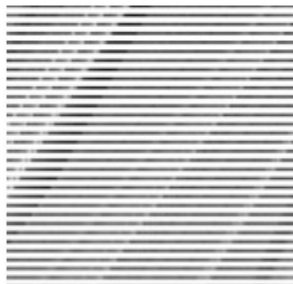
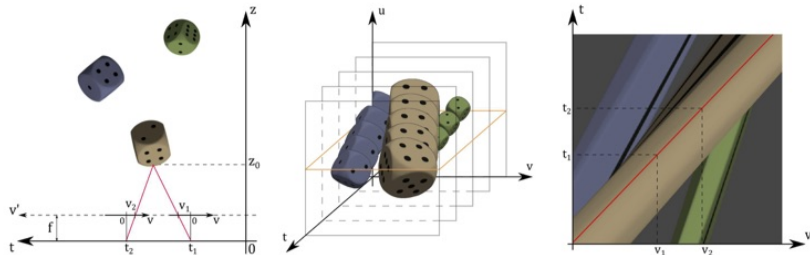
## Current application: Light Field Recovery



# Current application: Light Field Recovery



# Current application: Light Field Recovery



# Thanks!

## Questions?

