

## **Sampling and Reconstruction of Visual Appearance: From Denoising to View Synthesis**

CSE 274 [Fall 2022], Lecture 9

Ravi Ramamoorthi

<http://www.cs.ucsd.edu/~ravir>



1

## **Applications**

- Monte Carlo Rendering
- *Light Transport Acquisition / Many Light Rendering*
- Light Fields and Computational Photography
- View Synthesis
- Animation/Simulation (not covered in course)
- Introduce concepts of sparsity, coherence, compressive sensing for reconstruction

2

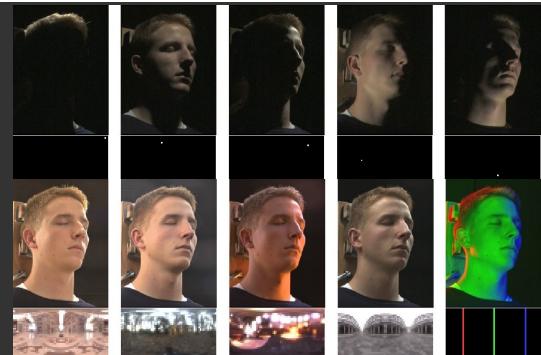
## **Acquiring Reflectance Field of Human Face [Debevec et al. SIGGRAPH 00]**

Illuminate subject from many incident directions



3

## **Example Images**



4

## **Motivation: Image-based Relighting**



Sample Lighting Directions

5

## **Motivation: Image-based Relighting**



Sample Lighting Directions

6

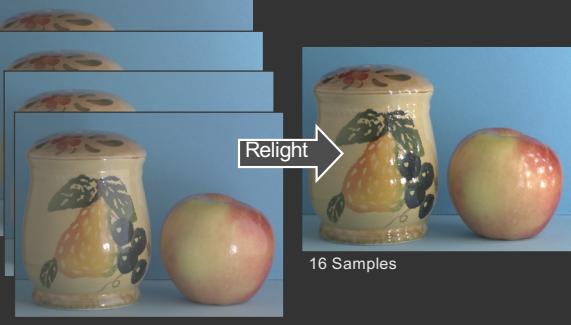
### Motivation: Image-based Relighting



Sample Lighting Directions

7

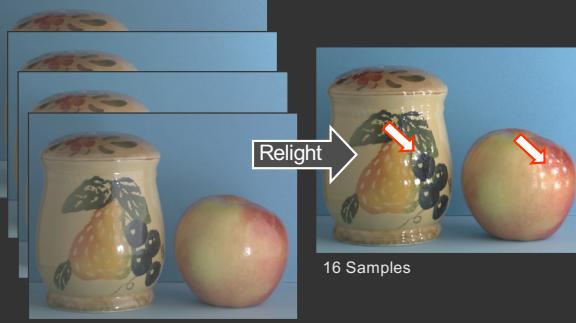
### Motivation: Image-based Relighting



Sample Lighting Directions

8

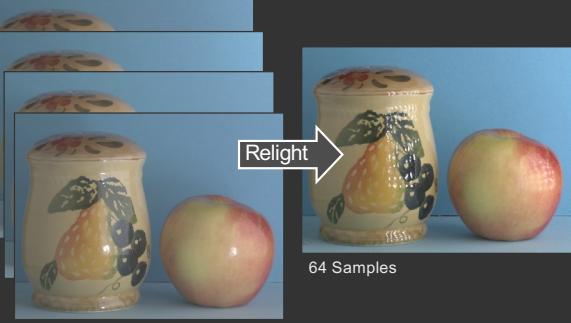
### Motivation: Image-based Relighting



Sample Lighting Directions

9

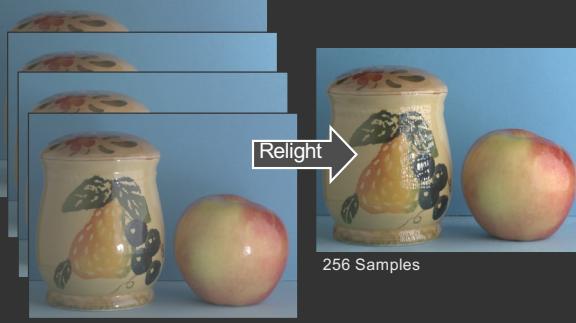
### Motivation: Image-based Relighting



Sample Lighting Directions

10

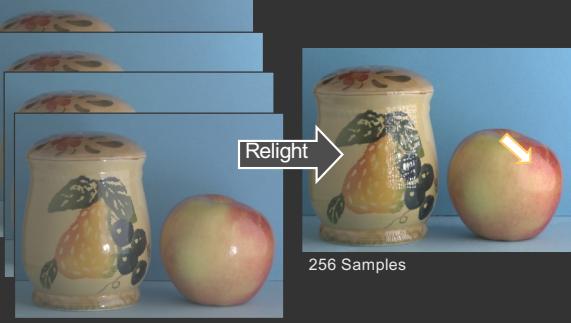
### Motivation: Image-based Relighting



Sample Lighting Directions

11

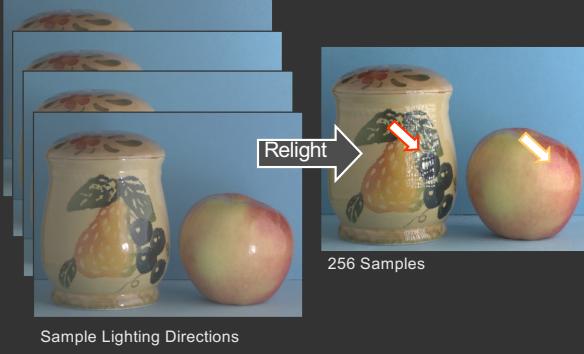
### Motivation: Image-based Relighting



Sample Lighting Directions

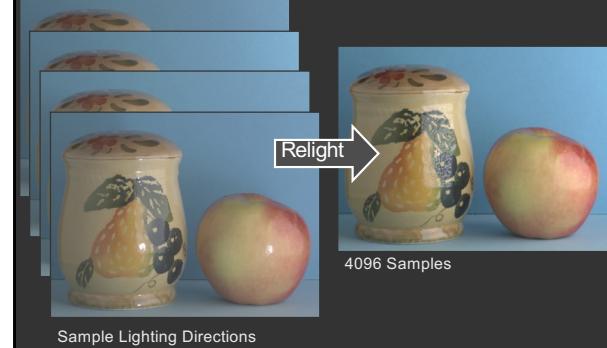
12

### Motivation: Image-based Relighting



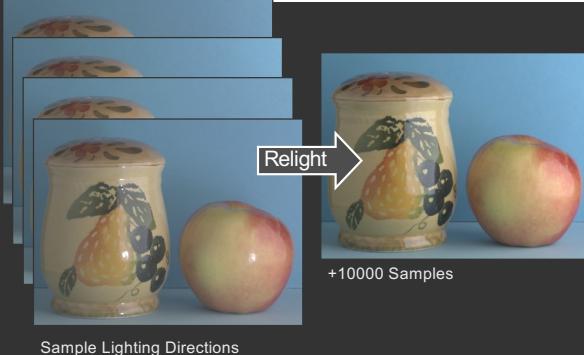
13

### Motivation: Image-based Relighting



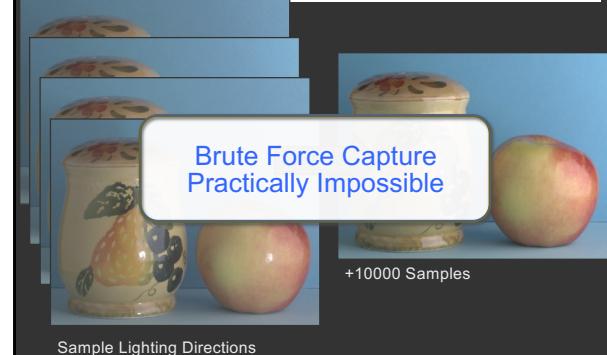
14

### Motivation: Image-based Relighting



15

### Motivation: Image-based Relighting



16

### Relighting as a Matrix-Vector Multiply

$$\begin{bmatrix}
 P_1 \\
 P_2 \\
 P_3 \\
 \vdots \\
 P_N
 \end{bmatrix}
 \begin{bmatrix}
 T_{11} & T_{12} & \cdots & T_{1M} \\
 T_{21} & T_{22} & \cdots & T_{2M} \\
 T_{31} & T_{32} & \cdots & T_{3M} \\
 \vdots & \vdots & \ddots & \vdots \\
 T_{N1} & T_{N2} & \cdots & T_{NM}
 \end{bmatrix}
 \begin{bmatrix}
 L_1 \\
 L_2 \\
 \vdots \\
 L_M
 \end{bmatrix}$$

17

### Relighting as a Matrix-Vector Multiply

$$\begin{array}{c}
 \text{Output Image (Pixel Vector)} \\
 \begin{bmatrix}
 P_1 \\
 P_2 \\
 P_3 \\
 \vdots \\
 P_N
 \end{bmatrix} \\
 = \\
 \begin{array}{c}
 \text{Input Lighting (Cubemap Vector)} \\
 \begin{bmatrix}
 T_{11} & T_{12} & \cdots & T_{1M} \\
 T_{21} & T_{22} & \cdots & T_{2M} \\
 T_{31} & T_{32} & \cdots & T_{3M} \\
 \vdots & \vdots & \ddots & \vdots \\
 T_{N1} & T_{N2} & \cdots & T_{NM}
 \end{bmatrix}
 \end{array} \\
 \begin{array}{c}
 \text{Precomputed Transport Matrix} \\
 \begin{bmatrix}
 L_1 \\
 L_2 \\
 \vdots \\
 L_M
 \end{bmatrix}
 \end{array}
 \end{array}$$

18

### Matrix Columns (Images)

$$\begin{bmatrix} T_{11} & T_{12} & \cdots & T_{1M} \\ T_{21} & T_{22} & \cdots & T_{2M} \\ T_{31} & T_{32} & \cdots & T_{3M} \\ \vdots & \vdots & \ddots & \vdots \\ T_{N1} & T_{N2} & \cdots & T_{NM} \end{bmatrix} \quad \begin{array}{c} \text{A 3D scene with a plant} \\ \text{represented as a matrix of columns} \end{array}$$

19

### (Pre)compute: Ray-Trace Image Cols

$$\begin{bmatrix} T_{11} & T_{12} & \cdots & T_{1M} \\ T_{21} & T_{22} & \cdots & T_{2M} \\ T_{31} & T_{32} & \cdots & T_{3M} \\ \vdots & \vdots & \ddots & \vdots \\ T_{N1} & T_{N2} & \cdots & T_{NM} \end{bmatrix} \quad \begin{array}{c} \text{A 3D scene with a teapot} \\ \text{represented as a matrix of columns} \end{array}$$

20

### (Pre)compute 2: Rasterize Matrix Rows

$$\begin{bmatrix} T_{11} & T_{12} & \cdots & T_{1M} \\ T_{21} & T_{22} & \cdots & T_{2M} \\ T_{31} & T_{32} & \cdots & T_{3M} \\ \vdots & \vdots & \ddots & \vdots \\ T_{N1} & T_{N2} & \cdots & T_{NM} \end{bmatrix} \quad \begin{array}{c} \text{A 3D scene with a plant} \\ \text{represented as a matrix of rows} \end{array}$$

21

### Outline

- *Matrix Row-Column Sampling (Many Lights)*  
(clustering for matrix completion of light transport)
- Compressive Sensing for Light Transport
- Matrix Completion

Hasan, Pellacini, Bala SIGGRAPH 07

22

### Complex Illumination: A Challenge



23

### Conversion to Many Lights

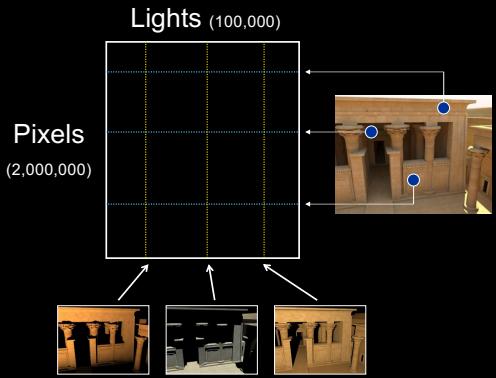
- Area, indirect, sun/sky



Courtesy Walter et al., Lightcuts, SIGGRAPH 05/06

24

## A Matrix Interpretation



25

## Problem Statement

- Compute sum of columns

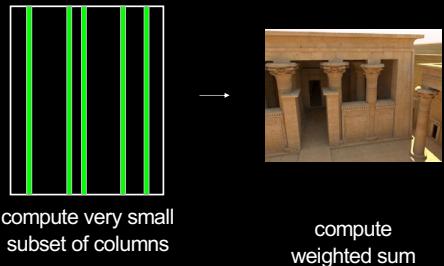
$$\text{Lights} = \sum (\text{Pixels})$$

The diagram shows a large image of a room with columns on the left, with three blue dots pointing to specific pixels. To the right is a large matrix with 100,000 columns and 2,000,000 rows. Arrows point from the image to the matrix and from the matrix to the equation.

26

## Image as a Weighted Column Sum

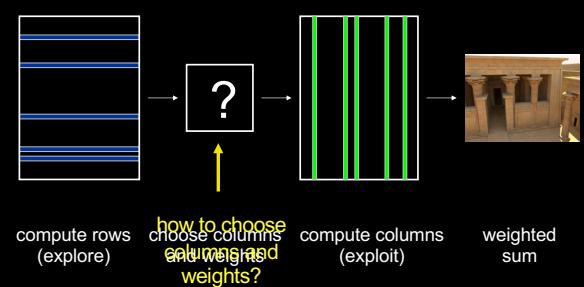
- The following is possible:



- Use rows to choose a good set of columns!

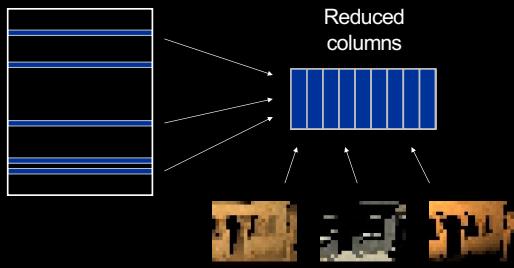
27

## Exploration and Exploitation



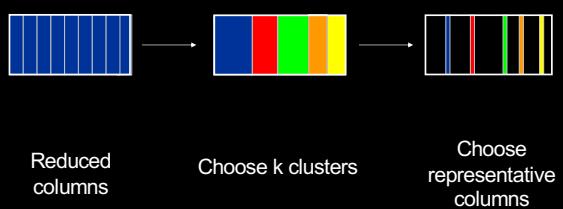
28

## Reduced Matrix



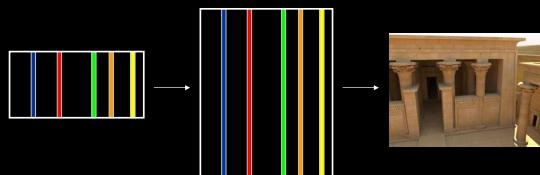
29

## Clustering Approach



30

## Reduced to Full

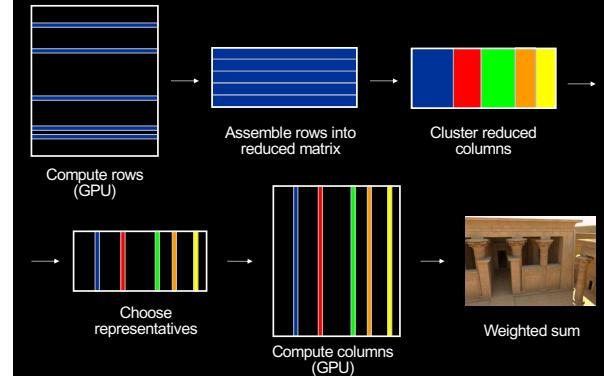


Representative columns

Use the same representatives for the full matrix

Weighted sum

## Full Algorithm



Assemble rows into reduced matrix

Cluster reduced columns

Choose representatives

Compute columns (GPU)

Weighted sum

31

32

## Results

- We show 5 scenes:



- Show reference and 5x difference image
- All scenes have 100,000+ lights
- Timings
  - NVidia GeForce 8800 GTX
  - Light / surface sample creation not included

33

## Results: Kitchen

- 388k polygons
- Mostly indirect illumination
- Glossy surfaces
- Indirect shadows



Our result: 13.5 sec  
(432 rows + 864 columns)



Reference: 13 min  
(using all 100k lights)

34

## Results: Temple

- 2.1m polygons
- Mostly indirect & sky illumination
- Indirect shadows



Our result: 16.9 sec  
(300 rows + 900 columns)

Reference: 20 min  
(using all 100k lights)

35

## Results: Trees

- 328k polygons
- Complex incoherent geometry



Our result: 2.9 sec  
(100 rows + 200 columns)

Reference: 14 min  
(using all 100k lights)

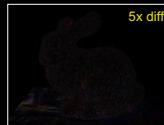
36

## Results: Bunny

- 869k polygons
- Incoherent geometry
- High-frequency lighting
- Kajiya-Kay hair shader



Our result: 3.8 sec  
(100 rows + 200 columns)



Reference: 10 min  
(using all 100k lights)

## Results: Grand Central

- 1.5m polygons
- Point lights between stone blocks



Our result: 24.2 sec  
(588 rows + 1176 columns)



Reference: 44 min  
(using all 100k lights)

37

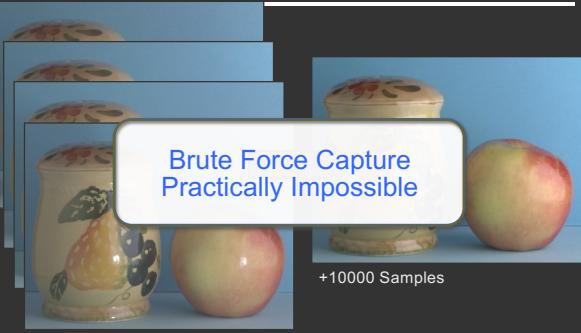
## Outline

- Matrix Row-Column Sampling (Many Lights)  
(clustering for matrix completion of light transport)
- *Compressive Sensing for Light Transport*
- Matrix Completion

Gu et al. ECCV 08  
Peers et al. SIGGRAPH 09  
Sen and Darabi EG 09 (reading)

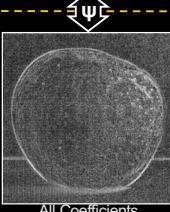
39

## Motivation: Image-based Relighting

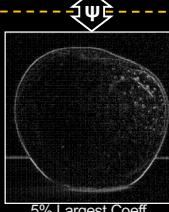


40

## Compressible / Sparseness

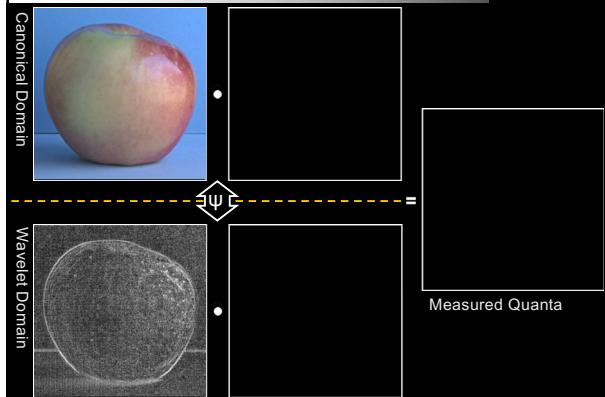


All Coefficients



5% Largest Coeff.

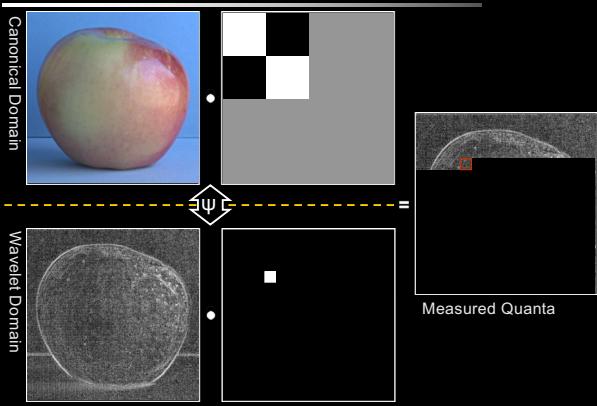
## Measurements



41

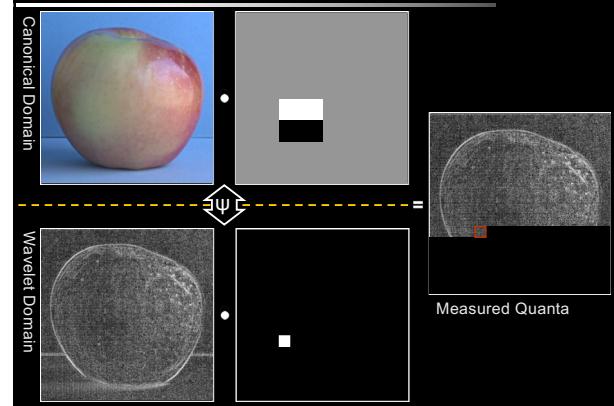
42

## Measurements



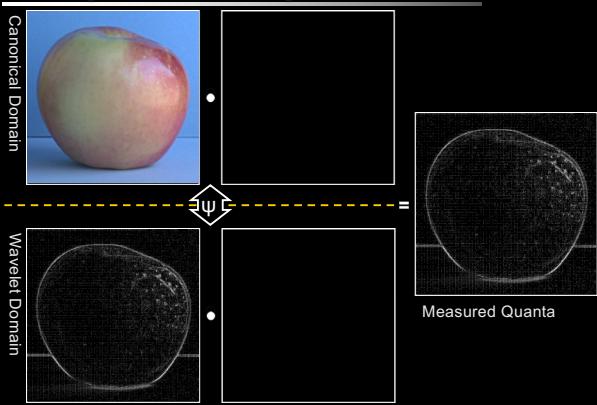
43

## Measurements



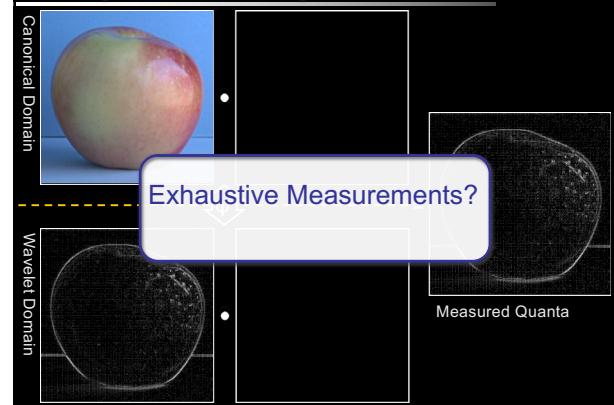
44

## Compressible / Sparseness



45

## Compressible / Sparseness



46

## Compressive Sensing: A Brief Introduction

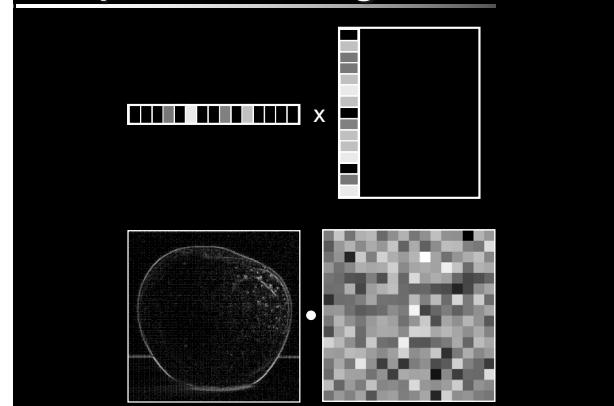
[Candes et al., 06][Donoho, 06]...

- Sparsity / Compressibility:
  - Signals can be represented as a few non-zero coefficients in an appropriately-chosen basis, e.g., wavelet, gradient, PCA.
- For sparse signals, acquire **measurements** (condensed representations of the signals) with **random projections**.

$$\mathbf{A} \begin{pmatrix} \text{Measurement Ensemble} \\ m \times n, \text{ where } m < n \end{pmatrix} \begin{pmatrix} \text{Signal} \\ n \times 1 \end{pmatrix} = \begin{pmatrix} \text{Measurements} \\ m \times 1 \end{pmatrix} \mathbf{b}$$

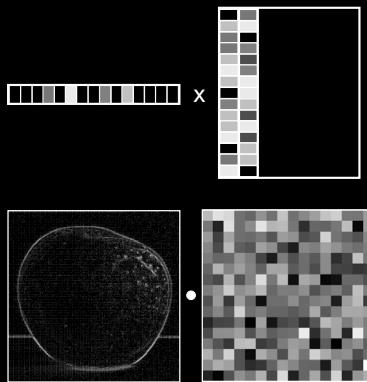
47

## Compressive Sensing



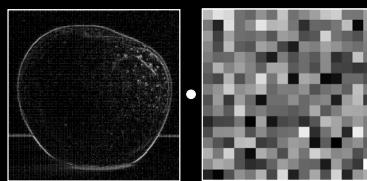
48

## Compressive Sensing



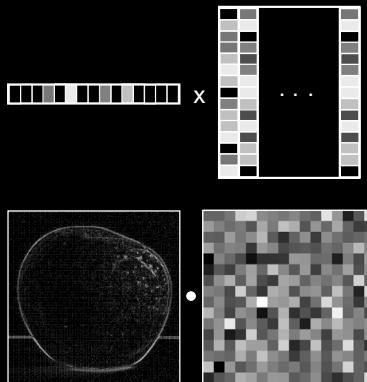
49

## Compressive Sensing



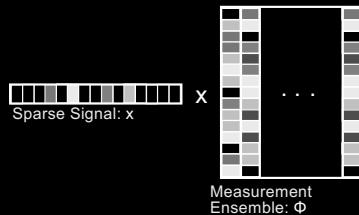
50

## Compressive Sensing



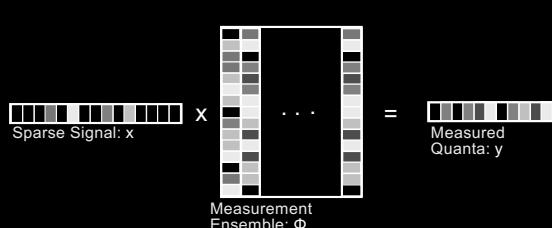
51

## Compressive Sensing



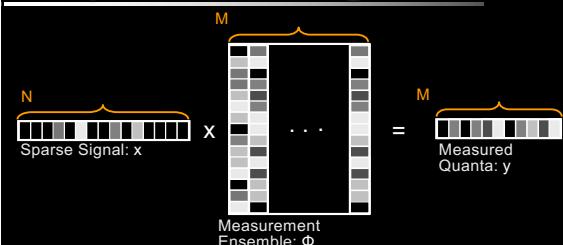
52

## Compressive Sensing



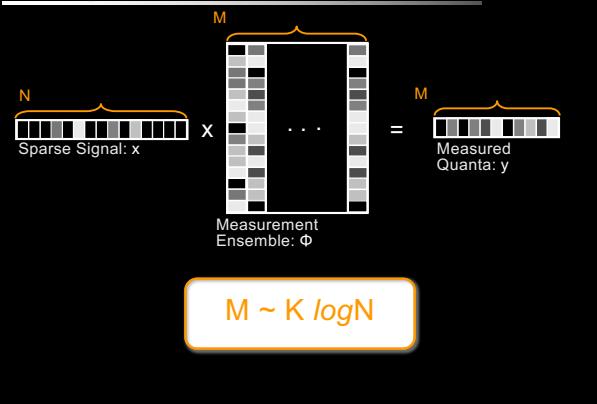
53

## Compressive Sensing



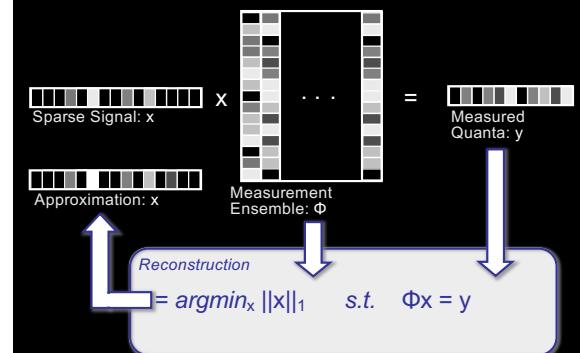
54

## Compressive Sensing



55

## Compressive Sensing



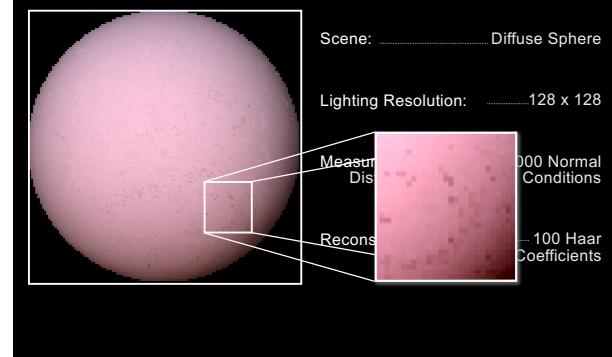
56

## Brute Force: Result



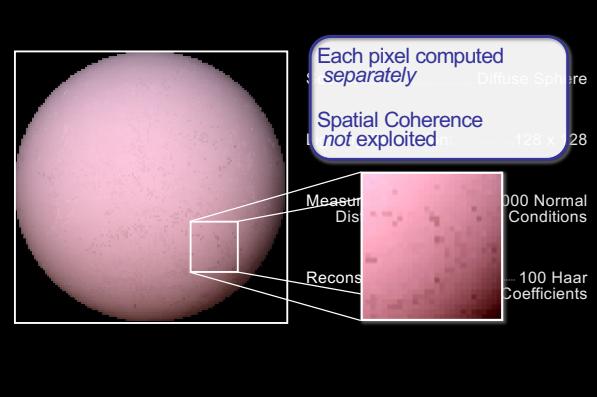
57

## Brute Force: Result



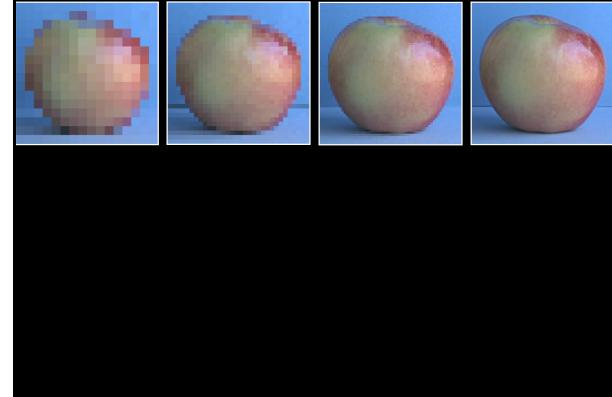
58

## Brute Force: Result

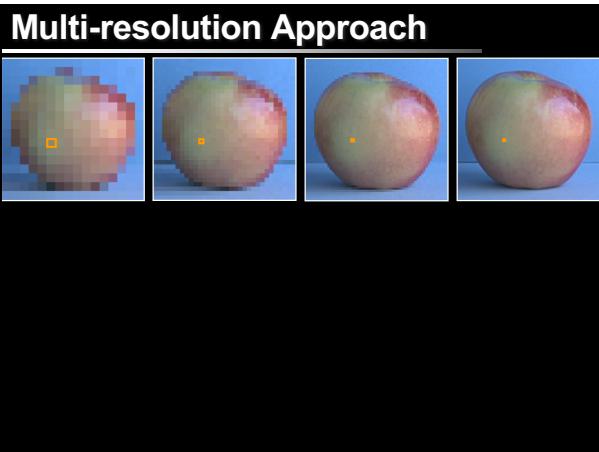


59

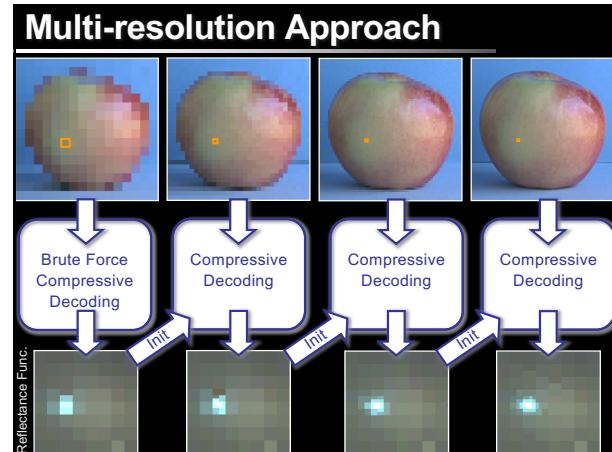
## Multi-resolution Approach



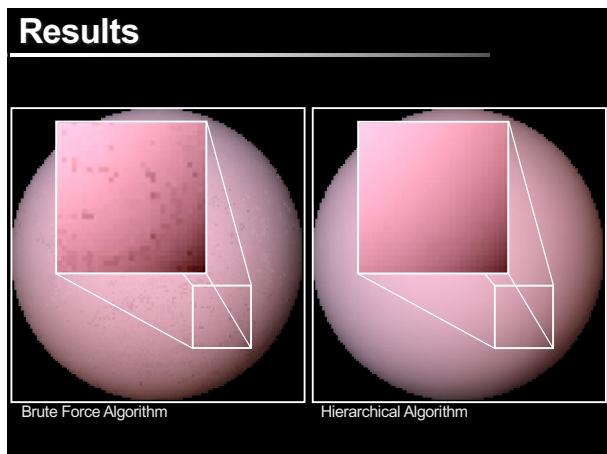
60



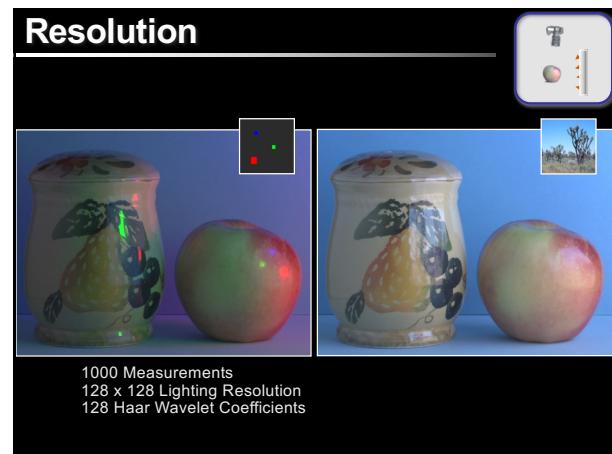
61



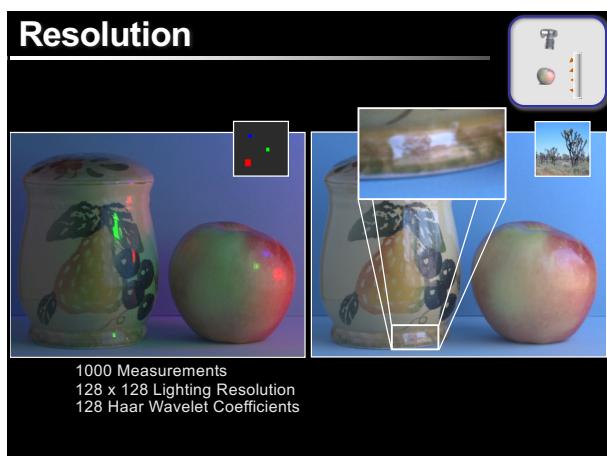
62



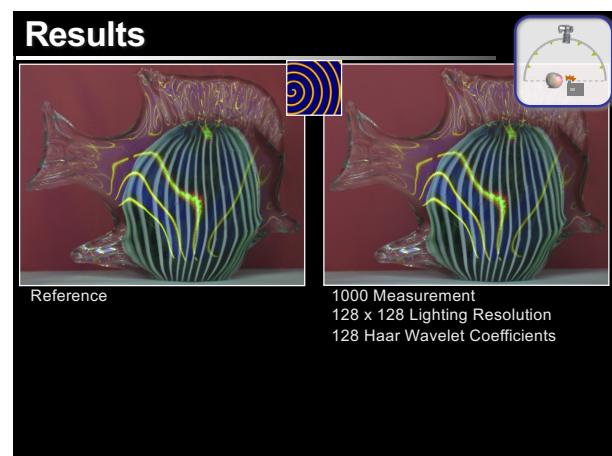
63



64

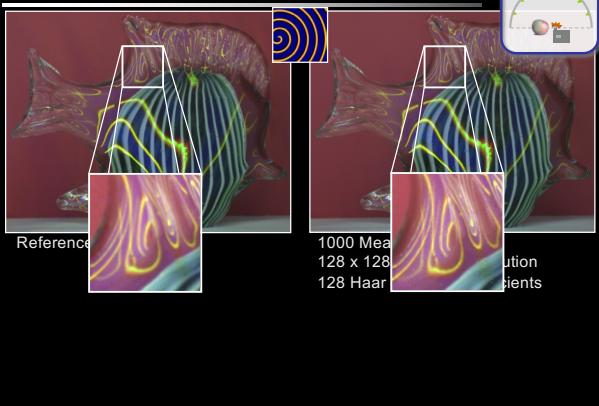


65



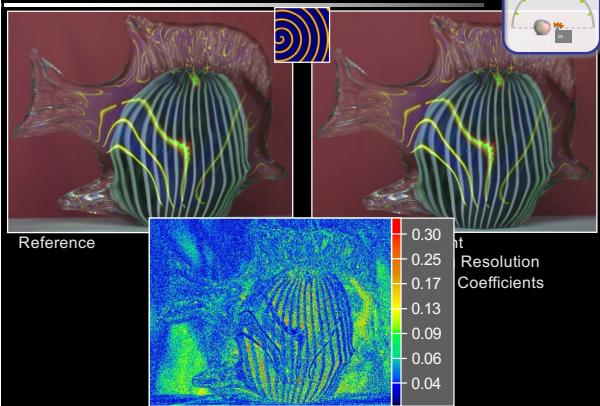
66

## Results



67

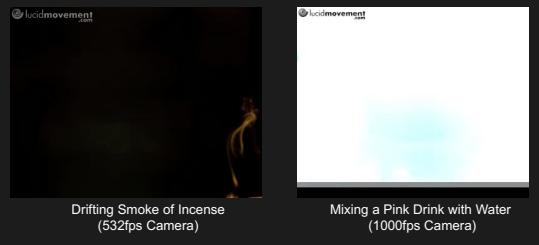
## Results



68

## Inhomogeneous Participating Media

Volume densities rather than boundary surfaces.  
Efficiency in acquisition is critical, especially for time-varying participating media.

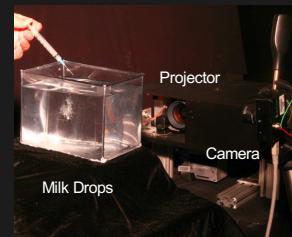


Video clips are from <http://www.lucidmovement.com>

69

## Compressive Structured Light

- Projector: DLP, 1024x768, 360 fps
- Camera: Dragonfly Express 8bit, 320x140 at 360 fps
- 24 measurements per time instance, and thus recover dynamic volumes up to  $360/24 = 15$  fps.

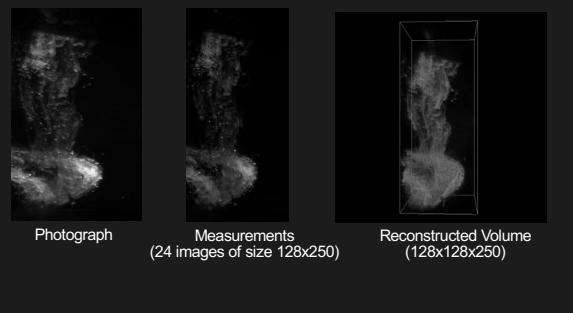


Gu, Nayar, Grinspun, Belhumeur, Ramamoorthi 08, 13

70

## Milk Dissolving: One Instance of time

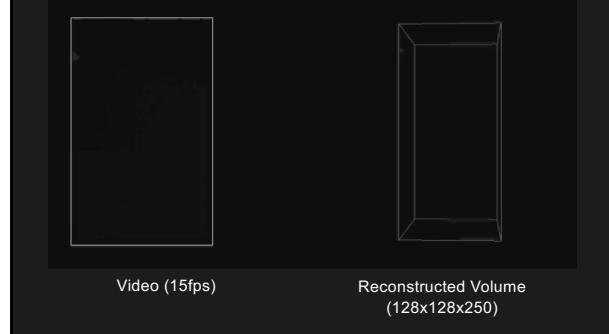
- Milk drops dissolving in a water tank.



71

## Milk Dissolving: Time-varying Volume

- Milk drops dissolving in a water tank.



72

## Outline

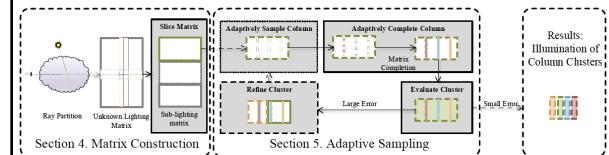
- Matrix Row-Column Sampling (Many Lights)  
(clustering for matrix completion of light transport)
- Compressive Sensing for Light Transport
- **Matrix Completion**
  - Extension to compressive sensing: Low rank matrices
  - Minimize matrix norm (rank), given some entries
  - Combine many ideas seen previously

Huo et al. SIGGRAPH Asia 16

73

## Outline

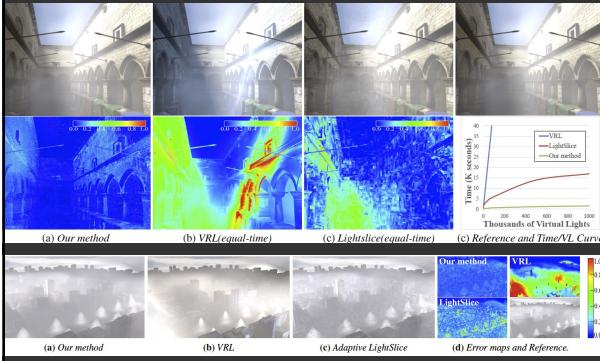
- **Matrix Completion**
  - Extension to compressive sensing: Low rank matrices
  - Minimize matrix norm (rank), given some entries
  - Combine many ideas seen previously



Huo et al. SIGGRAPH Asia 16

74

## Results (Participating Media)



75

## Summary

- Light Transport for Acquisition, Many Light Rendering
- Compressive Sensing for projected patterns
- Matrix Completion for many light rendering
- Leverages popular ideas in applied math
- Consider all forms of coherence
- Think about modern extensions with deep learning

76