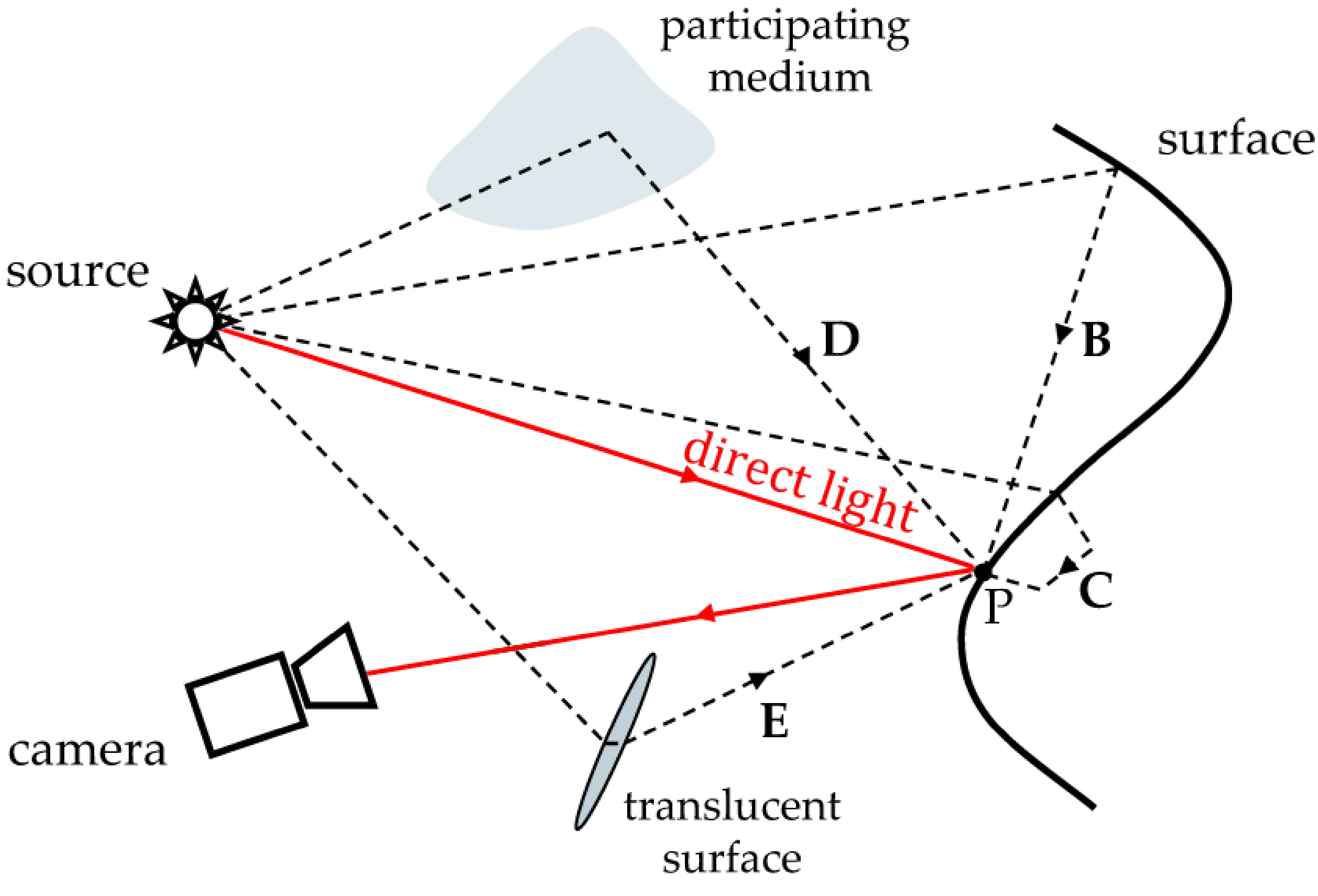
Fast Separation of Direct and Global Components of a Scene using High Frequency Illumination

S.K.Nayar, G.Krishnan, M.D.Grossberg, R.Raskar

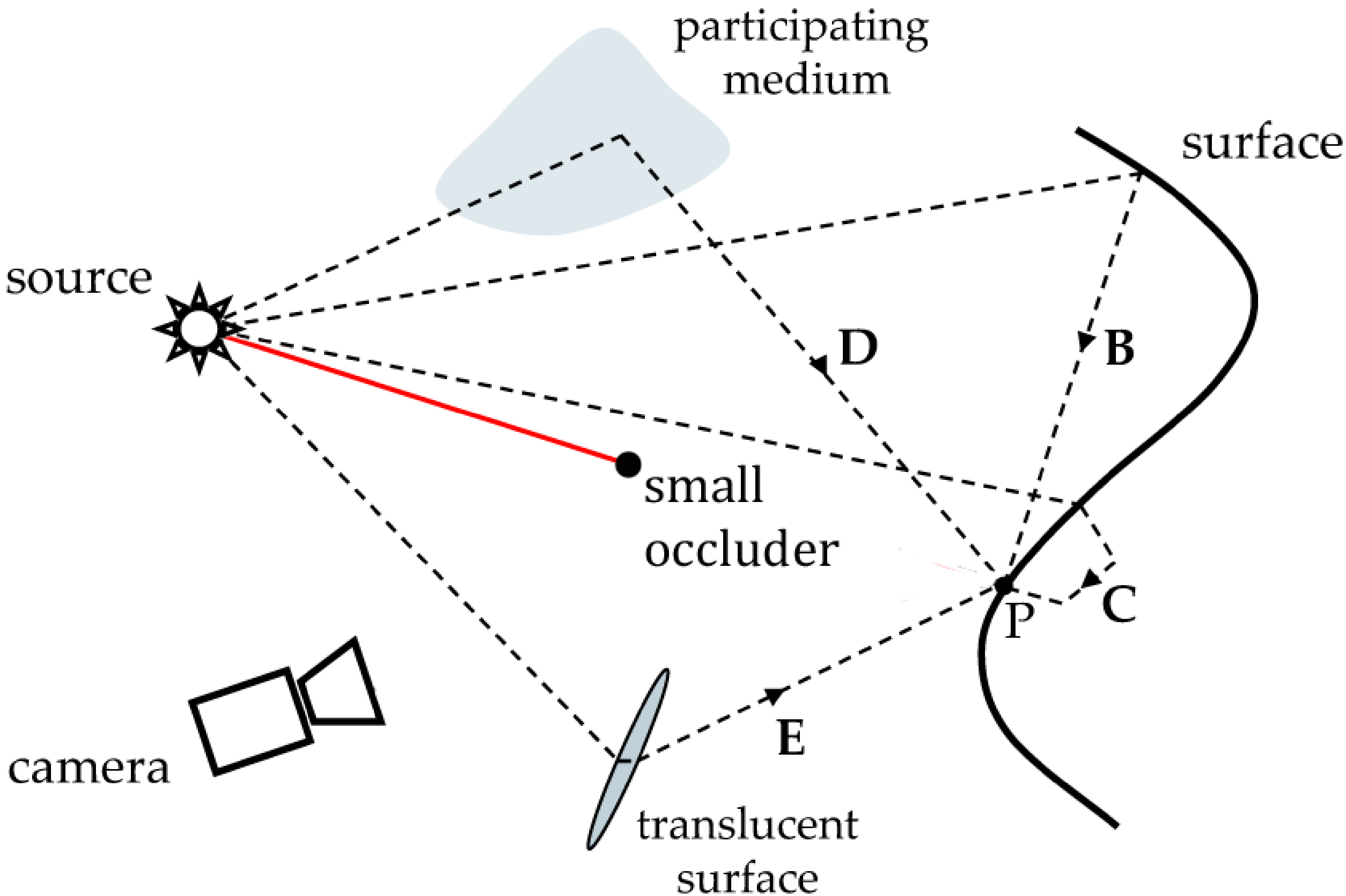
SIGGRAPH 2006

Presented by Vasily Volkov
UC Berkeley CS294-69

The problem: separate direct illumination

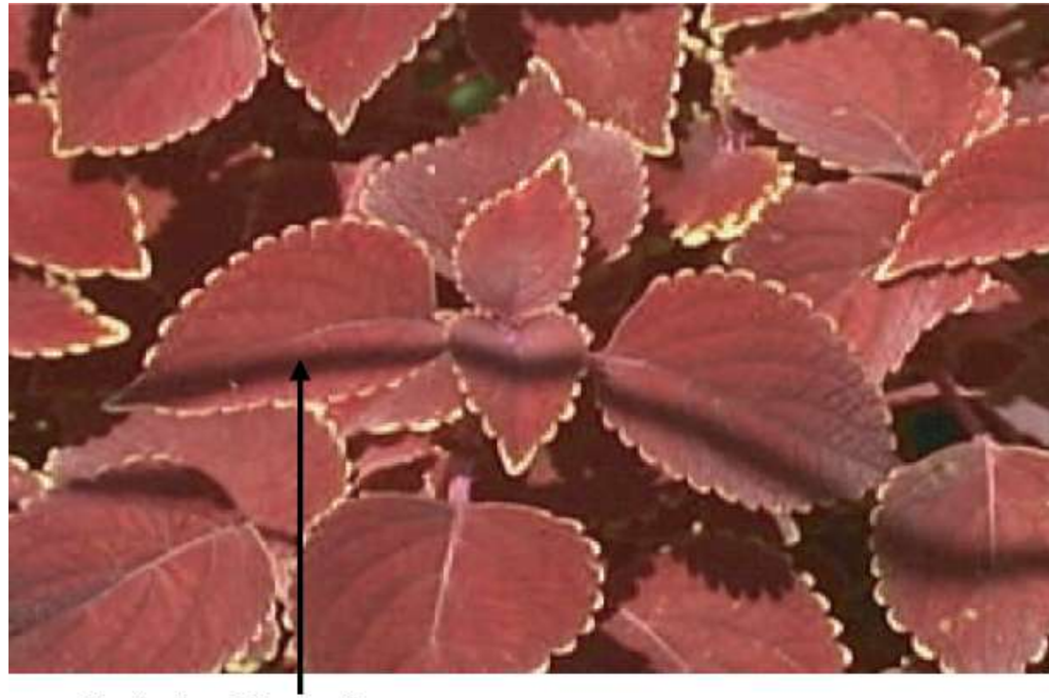


Idea: cancel it using a small occluder



Canceling direct illumination

Stick



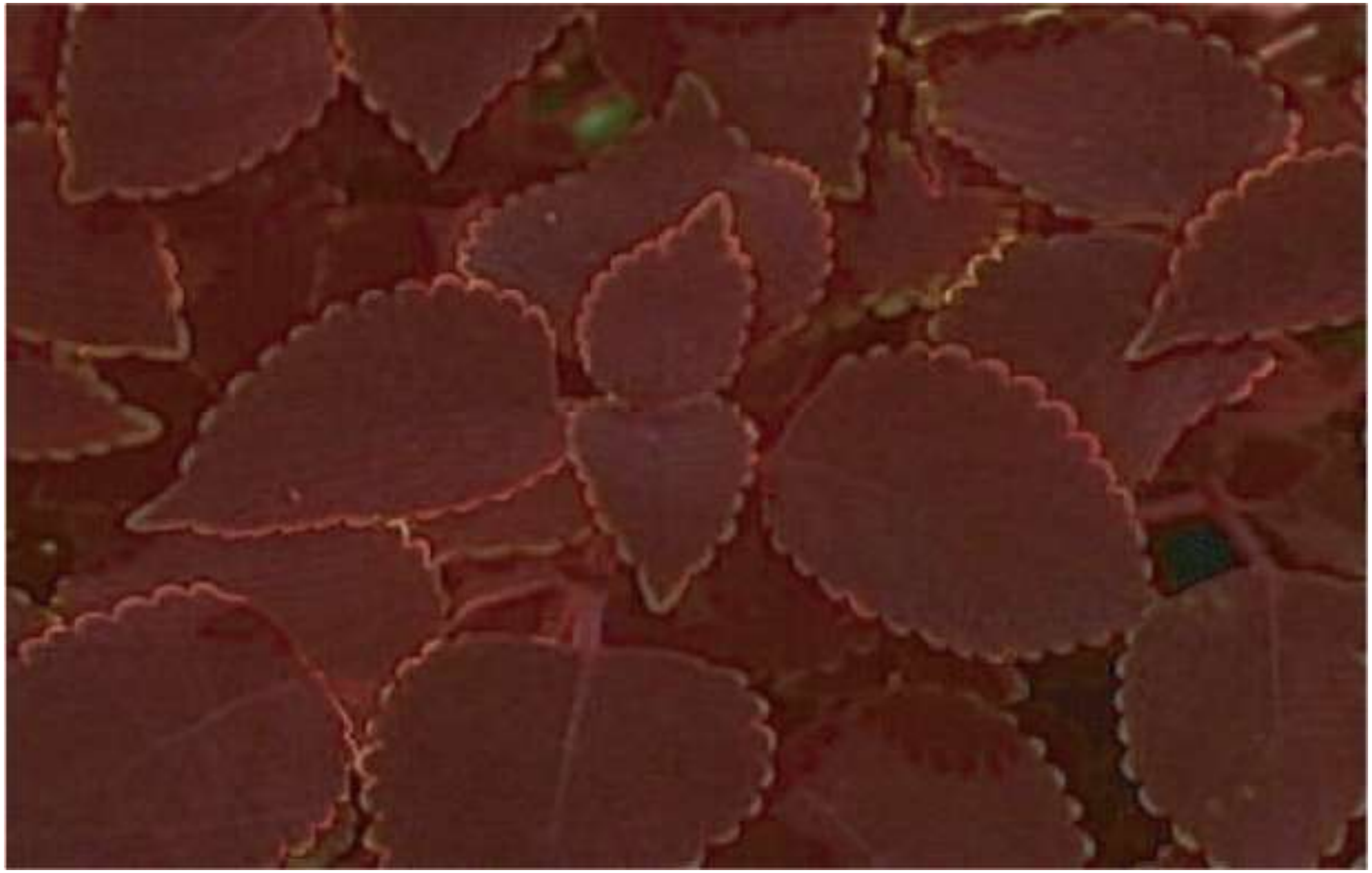
Stick Shadow

No direct illumination inside shadow

About same global illumination (but dimmer)

Locally extracted global illumination

Move shadow around, collect images



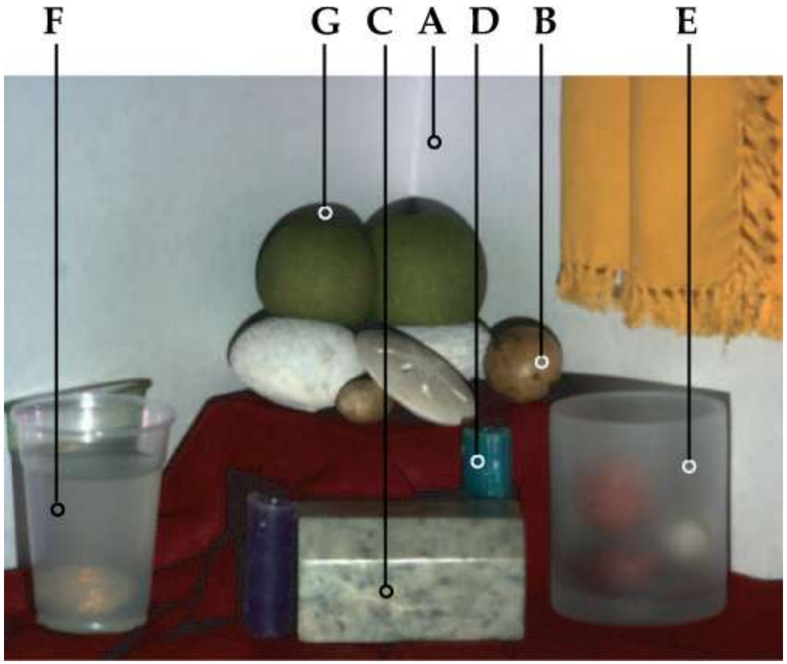
Global illumination for entire image

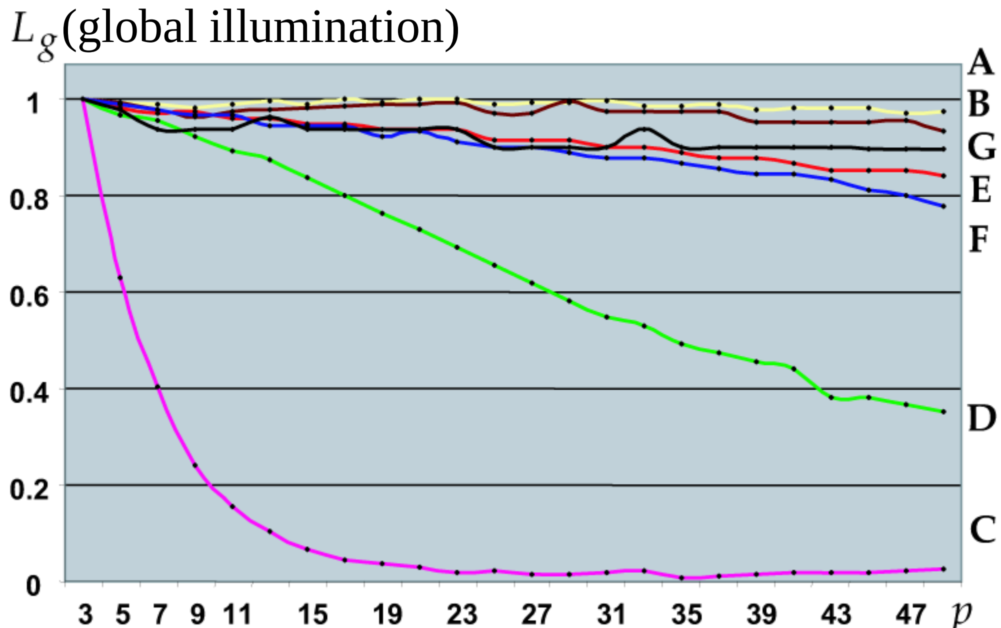
Composite of many shadow images

Get direct illumination by subtraction

Direct illumination = total – global illumination

The shadow must be small enough





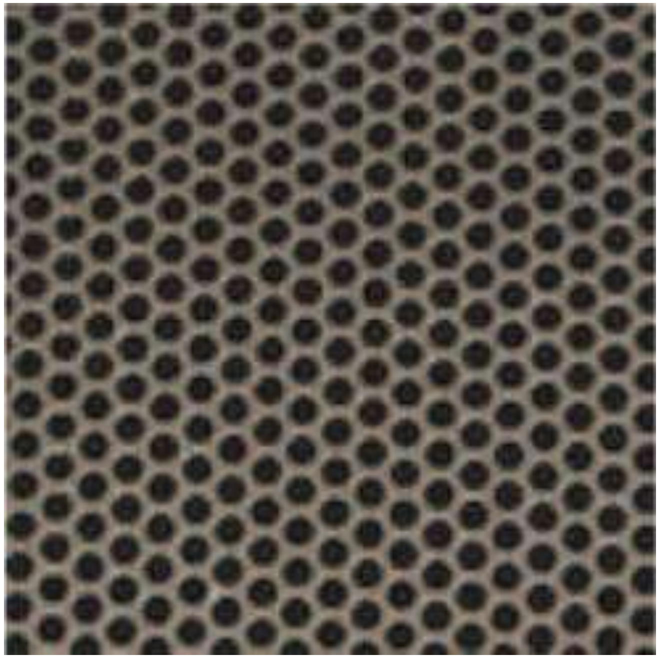
size of the shadow

C: subsurface scattering in marble

D: subsurface scattering in wax

F: diffuse scattering in diluted milk

Big but high frequency shadow is also OK



mesh

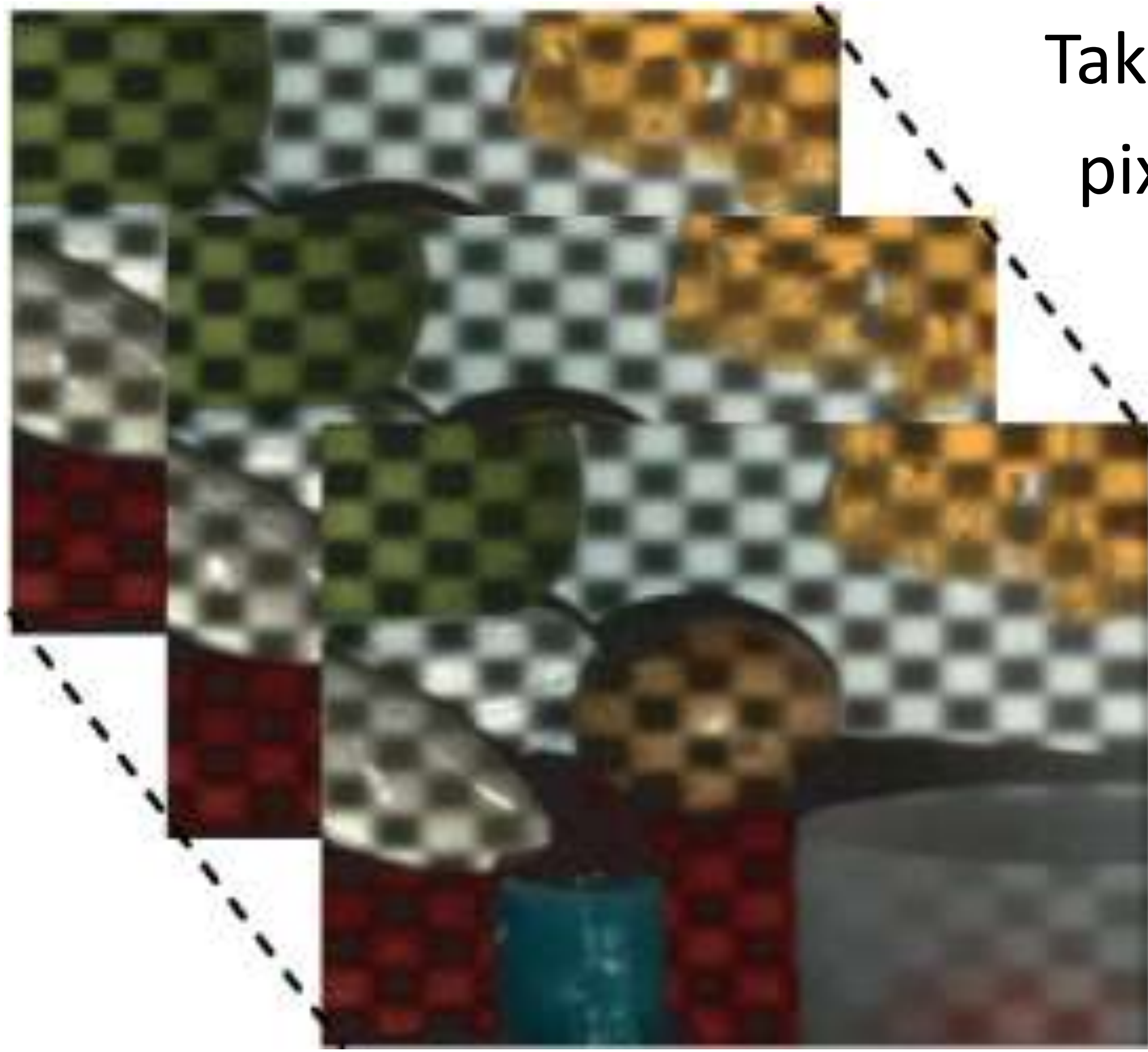
projector pattern

Key insight: global illumination is usually diffusive

Thus, any high-frequency shadow doesn't change it

Except it loses intensity by some factor

Only 25 images required for entire capture



Take min, max per pixel. Min=global illumination, max = total illumination

Example

direct illumination

global illumination

Separation using a single image

Instead of collecting images, use pixel windows

Take min or max in a window around pixel

Lose resolution, but OK in input is very high-res

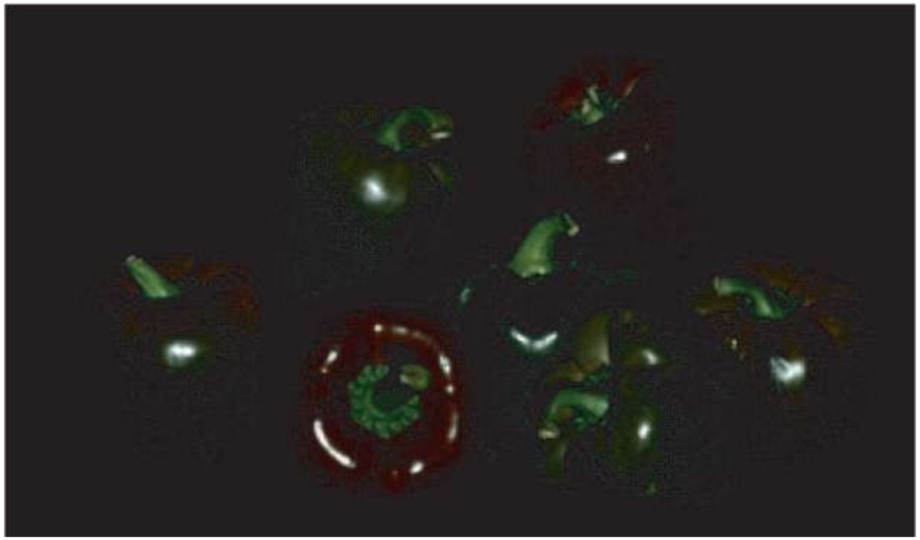
capturing:
2-pixel wide lines on
1024x768 projector

direct, 4x4 window

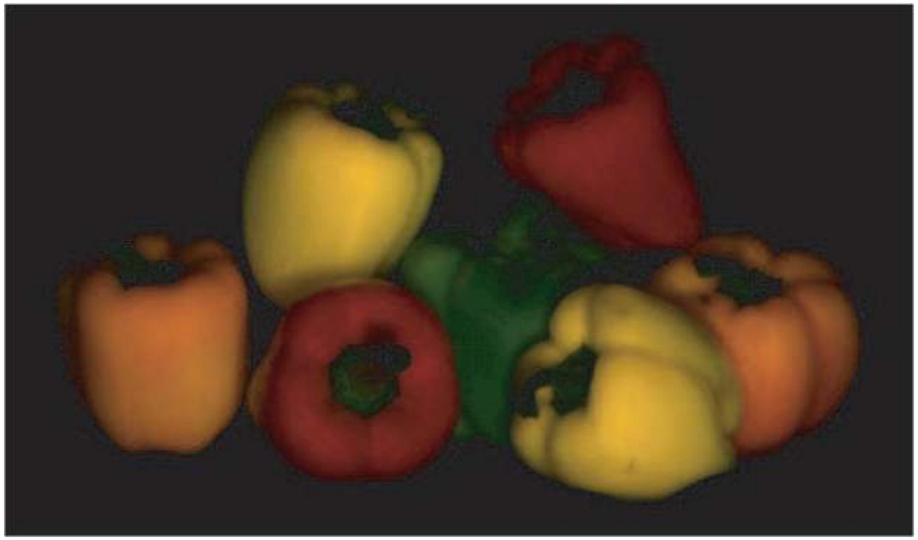
global illumination

More examples

original

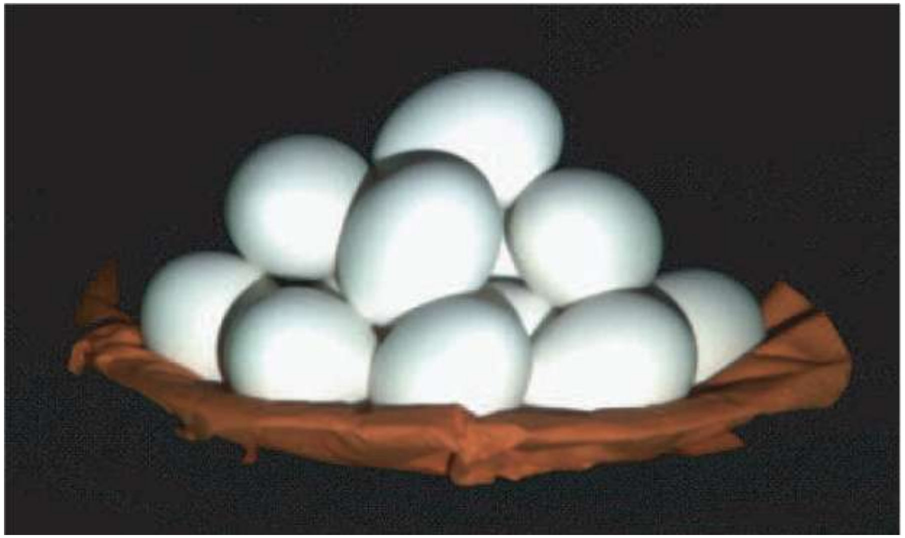


direct illumination

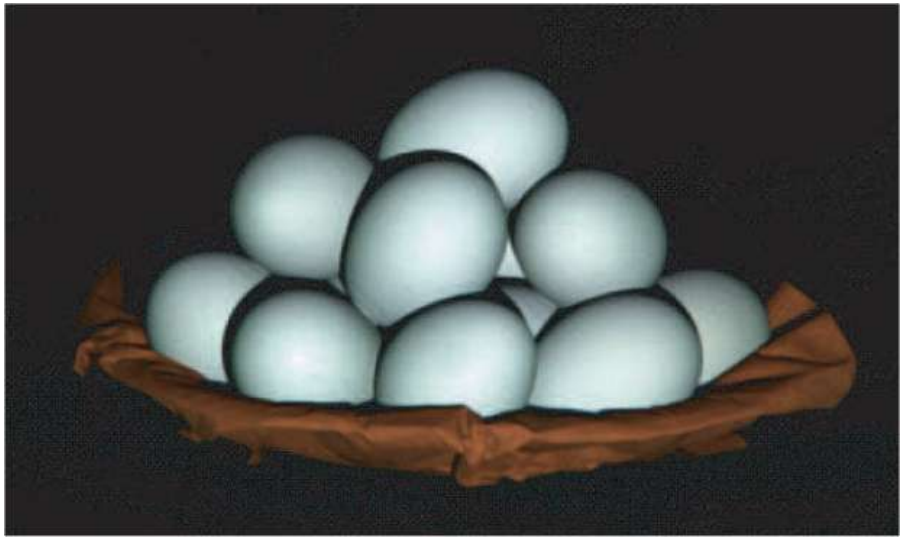


global illumination

More examples



original



direct illumination

global illumination

Application: new images

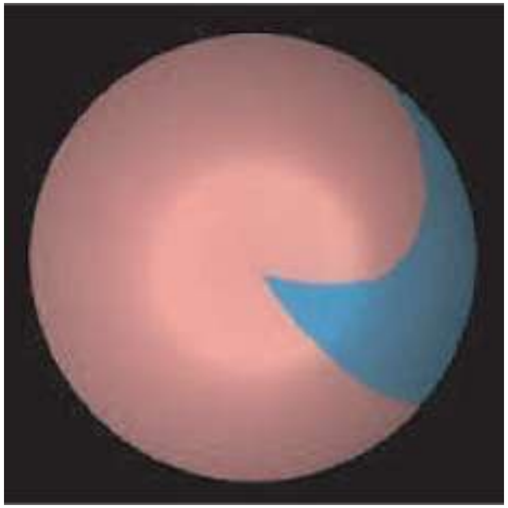
original

original

amplified global component

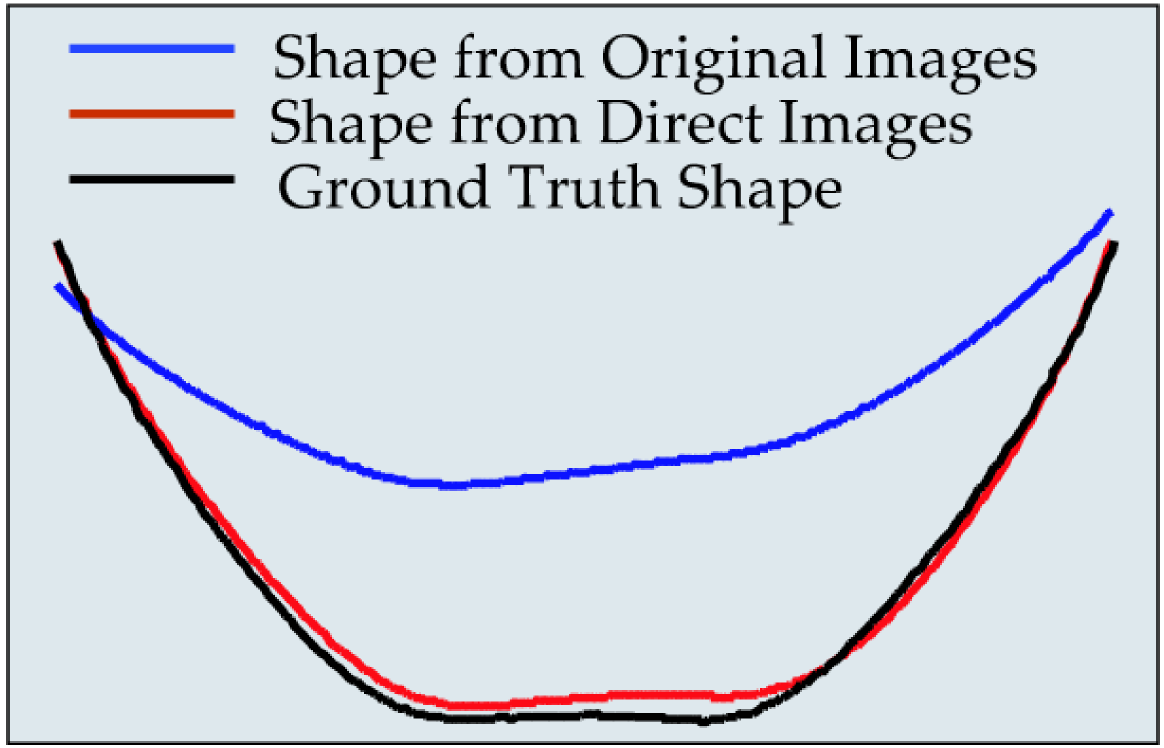
altered global component hue

Application: shape from image



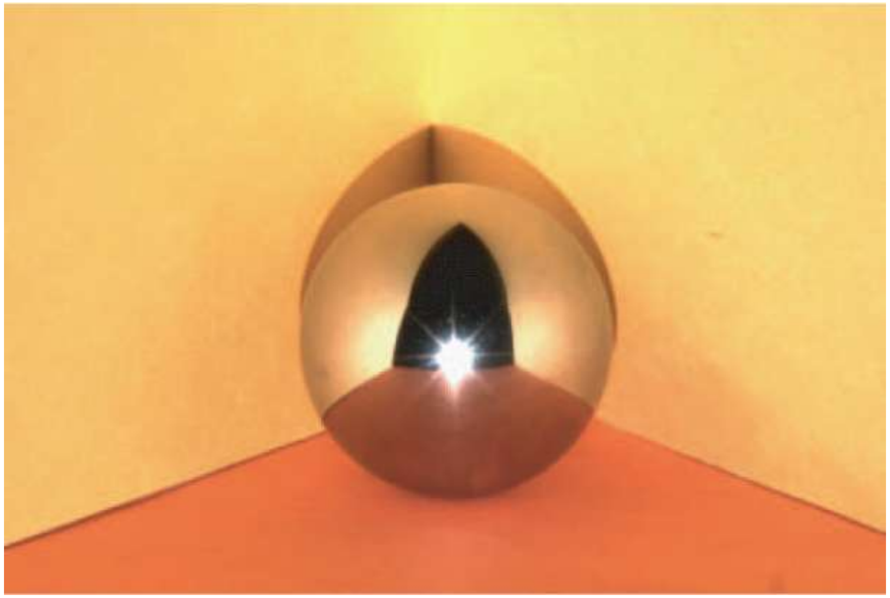
original image

direct illumination

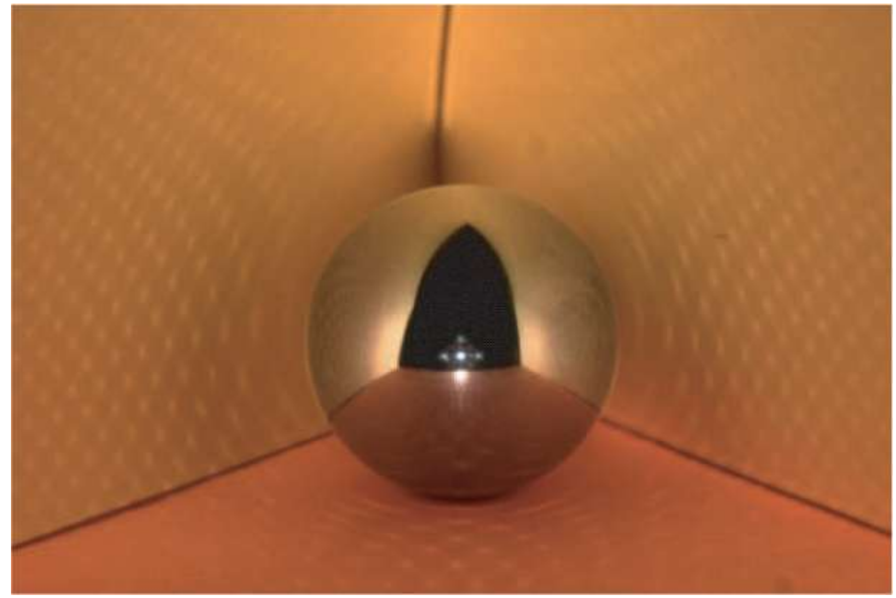


Improved accuracy

Limitation: non-diffusive surfaces



scene



global illumination

If surfaces are non-diffusive, even high-frequency shadows compromise global illumination

Do finer splitting?

- Direct illumination = 1-bounce
- Global illumination = n-bounce, n >= 2. (Split?)

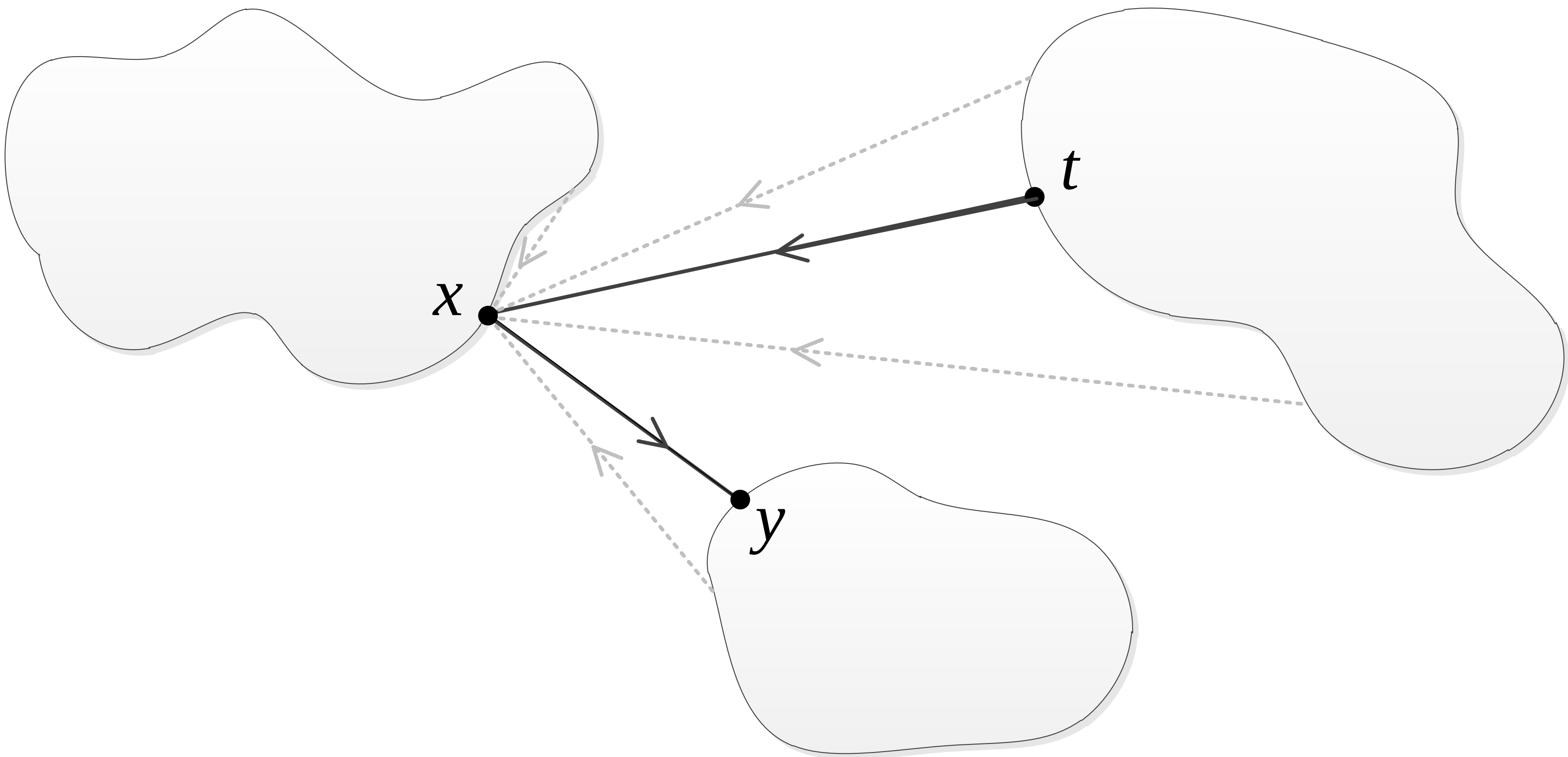
A Theory of Inverse Light Transport

S.M. Seitz, Y. Matsushita, and K. Kutulakos

CVPR 2005

Presented by Vasily Volkov
UC Berkeley CS294-69

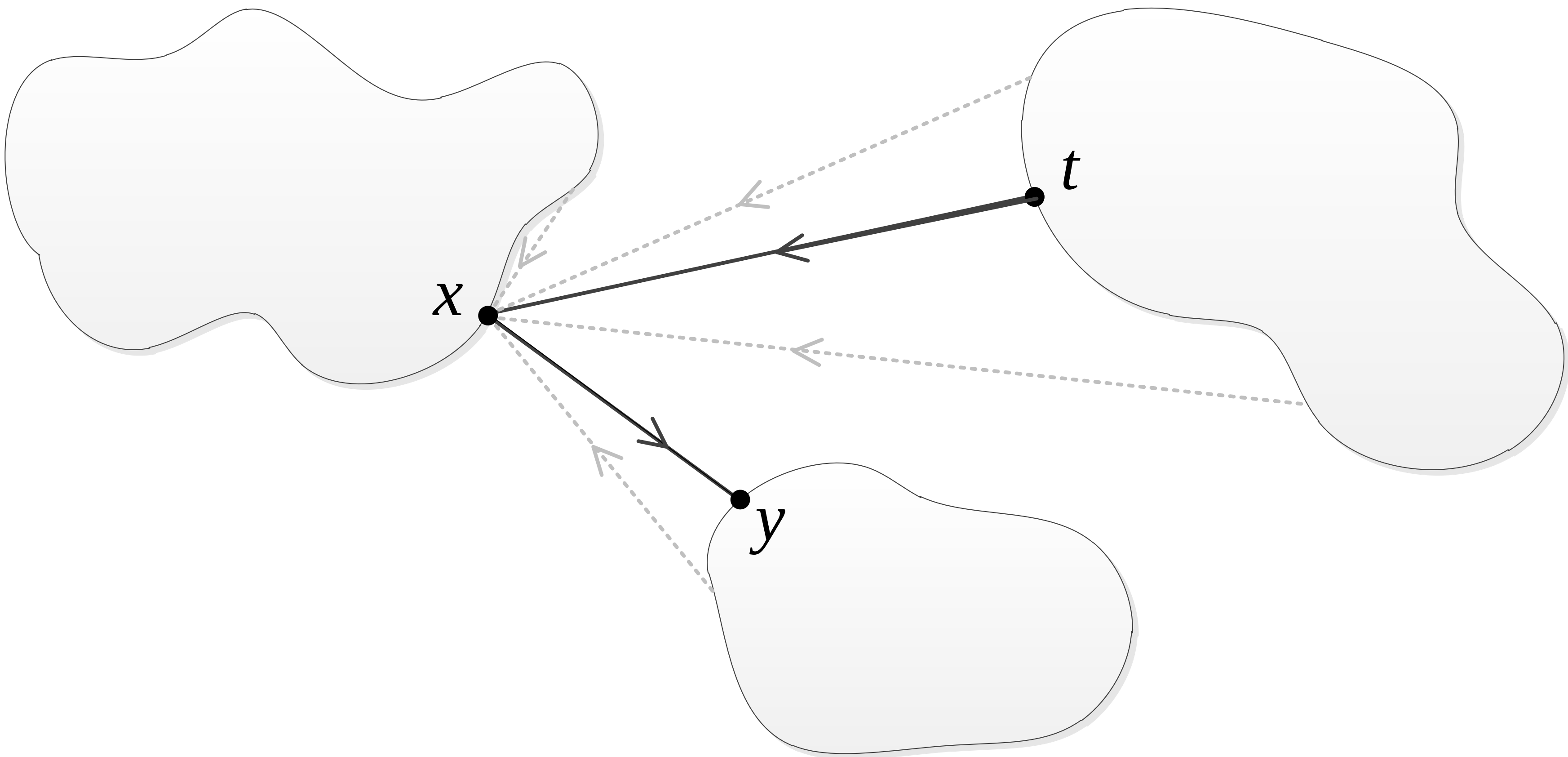
Rendering equation



$$L(\boldsymbol{x},\boldsymbol{y}) = \epsilon(\boldsymbol{x},\boldsymbol{y}) + \int \rho(\boldsymbol{x},\boldsymbol{y},\boldsymbol{t}) L(\boldsymbol{t},\boldsymbol{x})\, d\boldsymbol{t}$$

radiance from $\boldsymbol{x}$ to $\boldsymbol{y}$ — emission — reflection

Use direct illumination, not emission



$$L(\boldsymbol{x},\boldsymbol{y}) = \underbrace{L_1(\boldsymbol{x},\boldsymbol{y})}_{\text{direct light from } \boldsymbol{x} \text{ to } \boldsymbol{y}} + \int \rho(\boldsymbol{x},\boldsymbol{y},\boldsymbol{t}) L(\boldsymbol{t},\boldsymbol{x})\, d\boldsymbol{t}$$

Discrete rendering equation

$$\boldsymbol{l} = \boldsymbol{l}_1 + A\boldsymbol{l}$$

$\boldsymbol{l}$ – given discrete 4D light field (vector)

A – given discrete interreflection operator (matrix)

Solve for direct light $\boldsymbol{l}_1$:

$$\boldsymbol{l}_1 = \boldsymbol{l} - A\boldsymbol{l} = \underbrace{(I - A)}\boldsymbol{l}$$

interreflection cancellation operator C_1

N-bounce light fields

Solving $\boldsymbol{l} = \boldsymbol{l}_1 + A\boldsymbol{l}$ for $\boldsymbol{l}$ we have

$$\boldsymbol{l} = (I - A)^{-1}\boldsymbol{l}_1$$

$$= \boldsymbol{l}_1 + \underbrace{A\boldsymbol{l}_1}_{\boldsymbol{l}_2} + \underbrace{A^2\boldsymbol{l}_1}_{\boldsymbol{l}_3} + \underbrace{A^3\boldsymbol{l}_1}_{\boldsymbol{l}_4} + \cdots$$

Where $\boldsymbol{l}_n \equiv A^{n-1}\boldsymbol{l}_1$ is n-bounce light field, i.e. light that bounced n times off surfaces.

Since $\boldsymbol{l}_1 = (I - A)\boldsymbol{l}$, we have:

$$\boxed{\boldsymbol{l}_n = A^{n-1}(I - A)\boldsymbol{l} \equiv C_n\boldsymbol{l}}$$

From 4D light field to 2D illumination field

Using 4D light field l (≈lumigraph) is costly

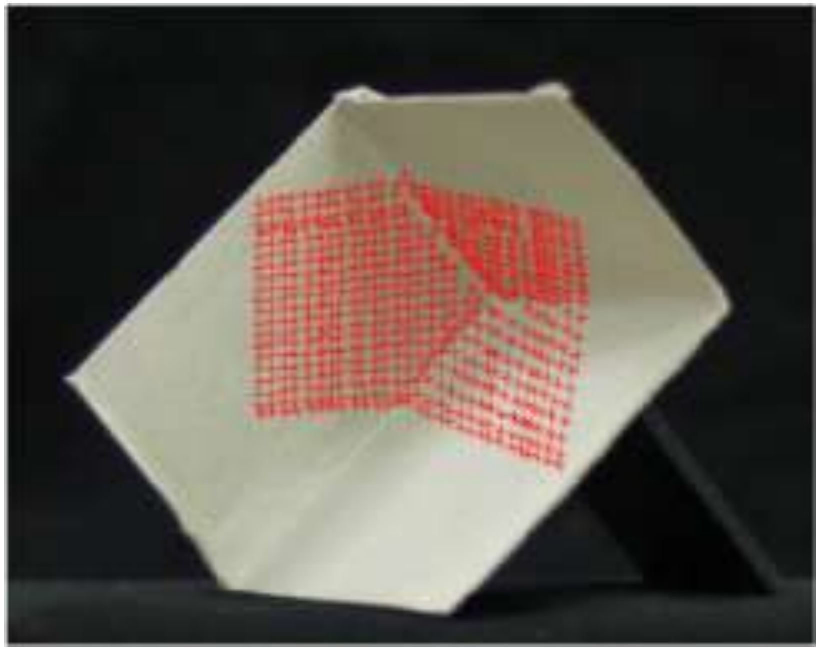
Assume Lambertian reflection!

- Same radiance in all directions – reduces problem to 2D
- Much less expensive, but approximate

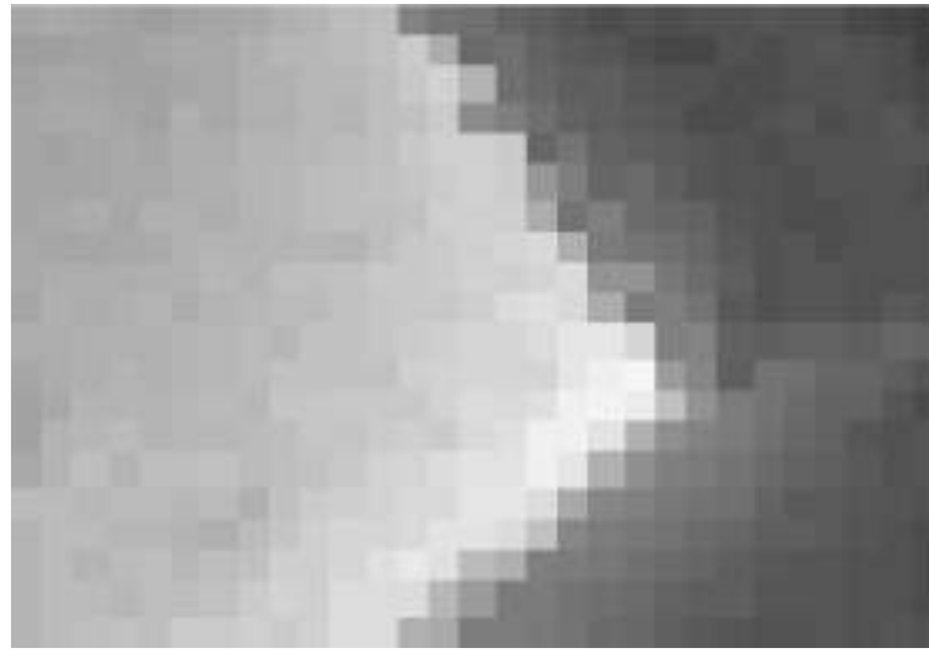
2D radiance – can capture using a single image

- Must ensure all scene points are visible
- Otherwise, bounces off occluded points are not counted

Capturing 2D illumination field



scene and sample points



captured image I

Capturing matrix A

Where to get A? Get it from $\boldsymbol{l}_1 = (I - A)\boldsymbol{l}$

Capture many independent light fields $\boldsymbol{t}_1$, $\boldsymbol{t}_2$, …

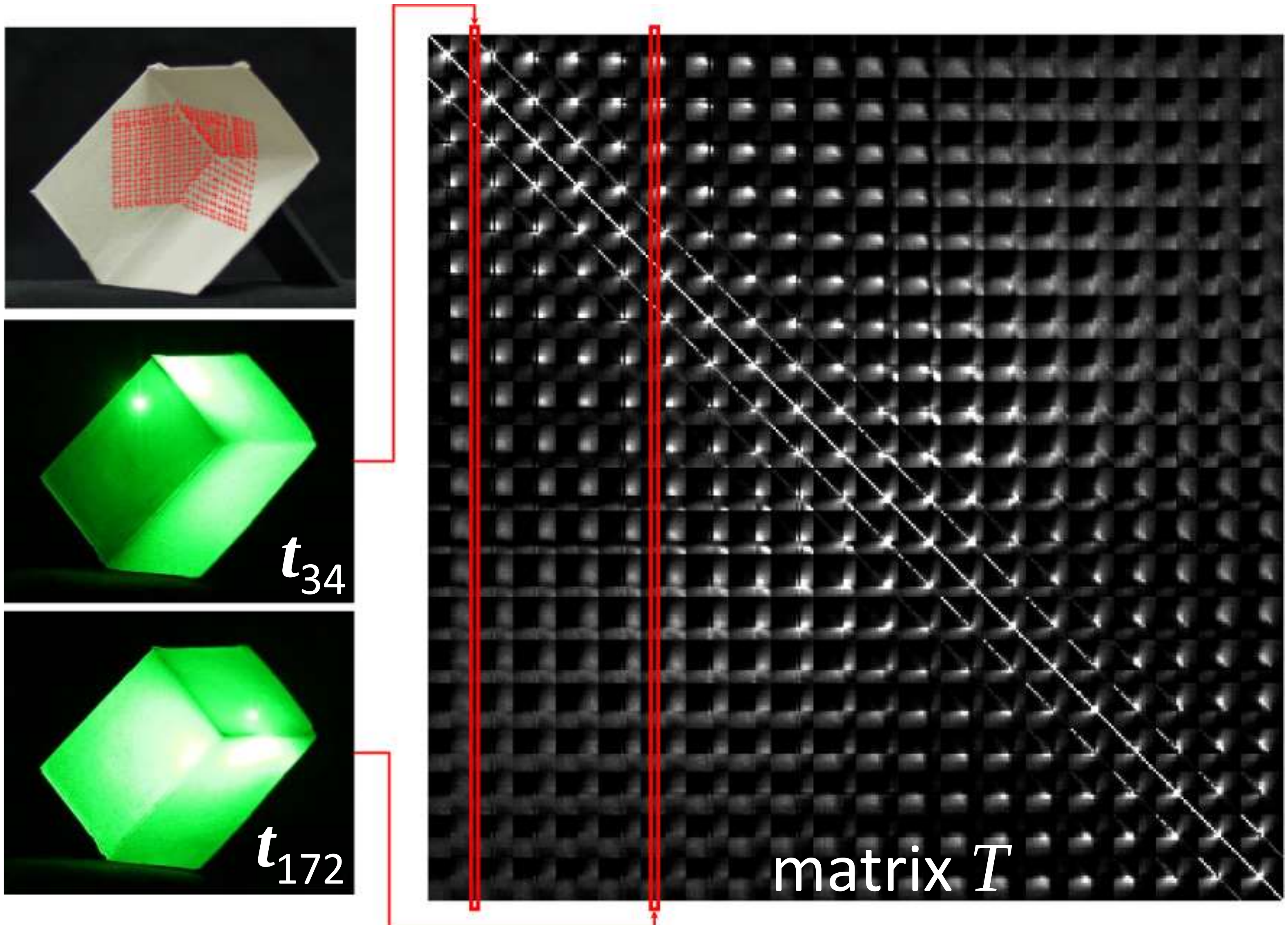
Build square matrix $T = [\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_m]$

Now, $T_1 = (I - A)T$ and $I - A = T_1 T^{-1}$

How to get T_1? (Capture?) Here is a workaround:

- Note that $I - A$ has 1s on the diagonal
- Use impulse illumination to get diagonal T_1

Impulse illumination using laser



Highlight single pixel – direct light is zero elsewhere

Dealing with T_1

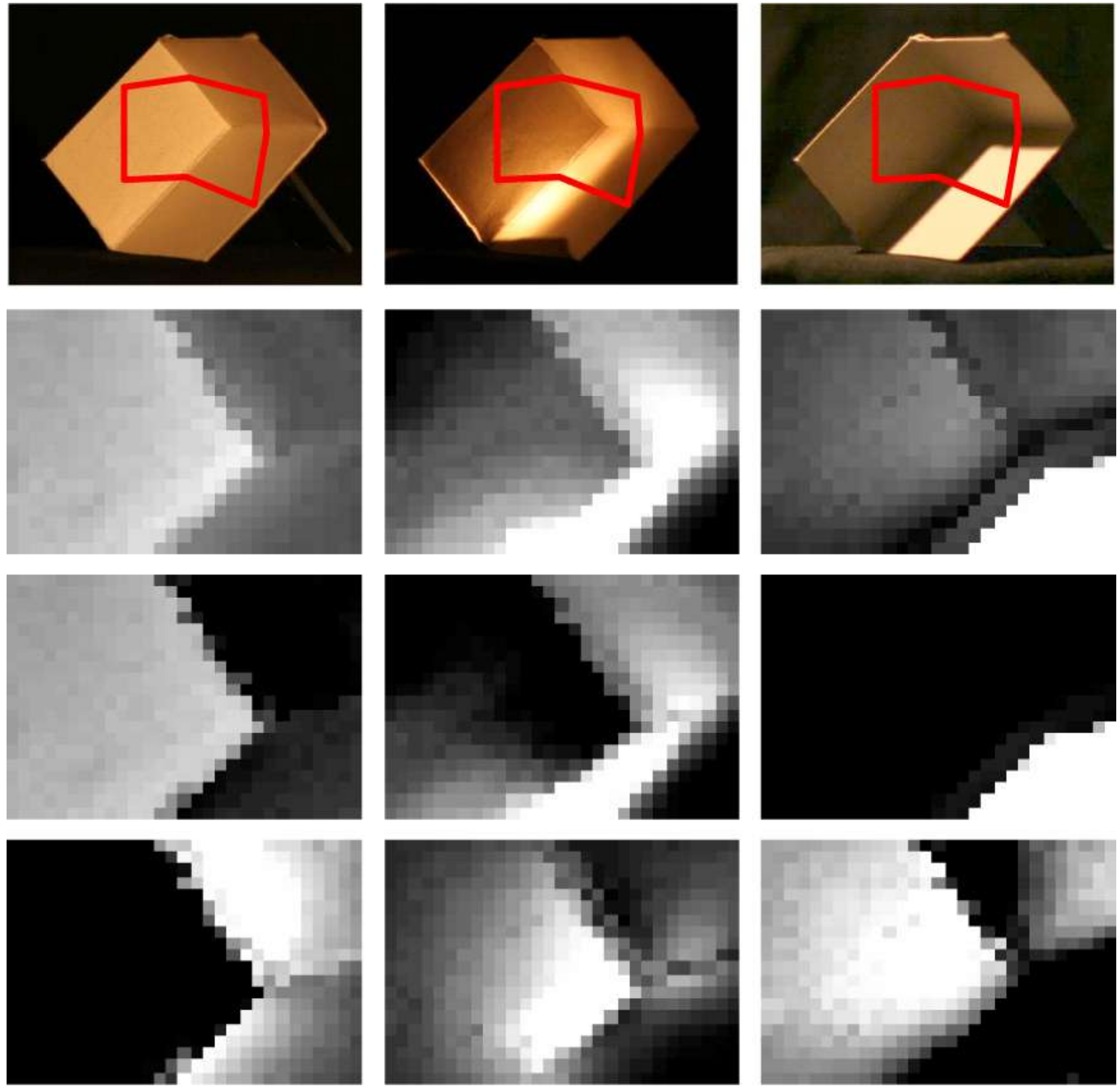
Why $I - A$ has 1s on the diagonal?

- A is interreflection matrix
- Point $\boldsymbol{x}$ doesn't reflect light from $\boldsymbol{x}$
- So, A has 0 on diagonal and $I - A = T_1 T^{-1}$ has 1s

Since T_1 is diagonal, $T_1 T^{-1}$ is row-scaling of T^{-1}

- Since $T_1 T^{-1}$ has 1s on diagonal, this defines $I - A$

Results



Input image $\boldsymbol{I}$

Direct light $\boldsymbol{I}_1$

2-bounce $\boldsymbol{I}_2$
(amplified 3x)

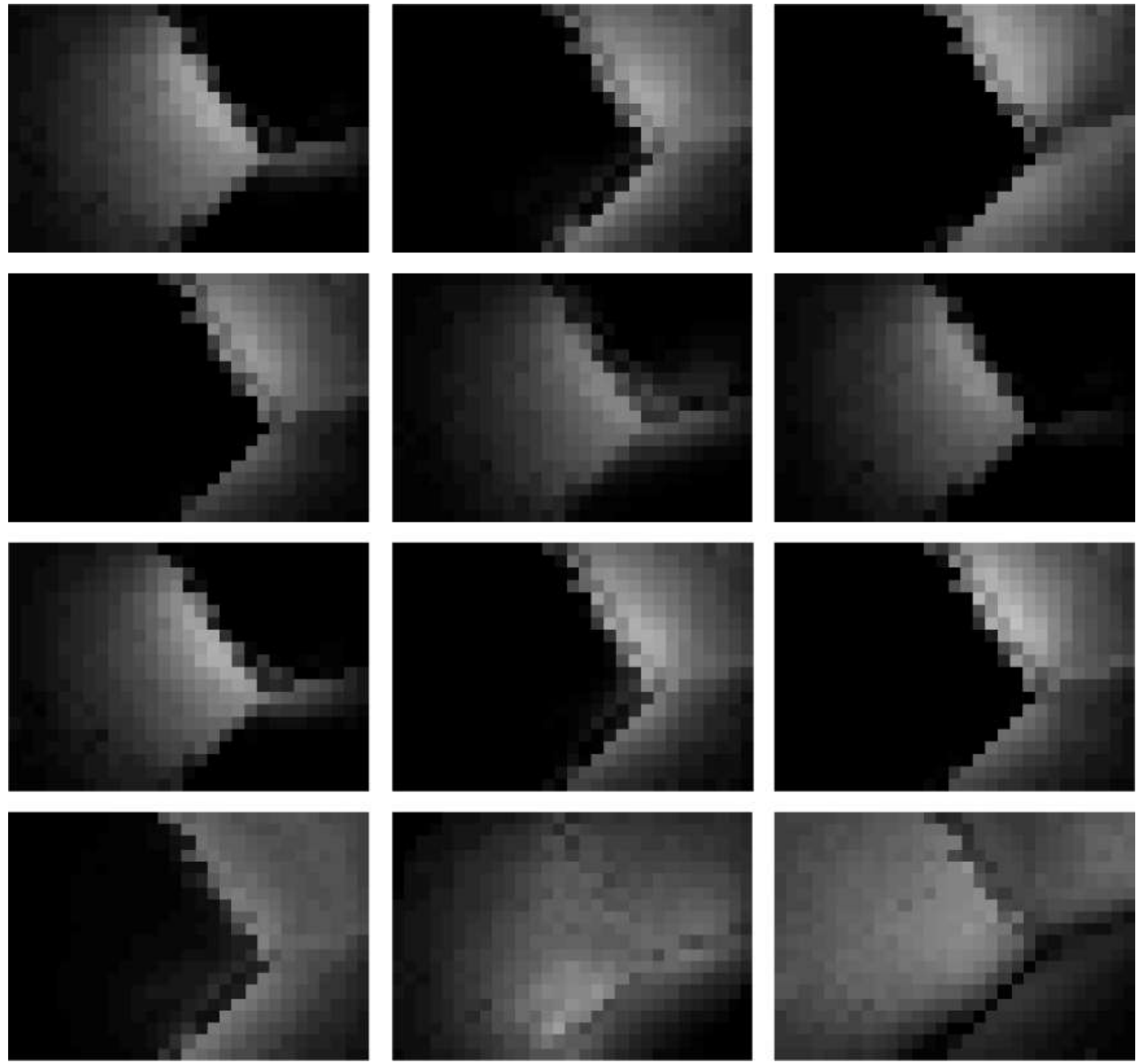
Results

3-bounce $\boldsymbol{l}_3$ (amplified 3x)

4-bounce $\boldsymbol{l}_4$ (6x)

5-bounce $\boldsymbol{l}_5$ (15x)

indirect illumination $\boldsymbol{l} - \boldsymbol{l}_1$



Summary

1. If we know both the lightfield and the interreflection operator, we can compute all N-bounce light fields

2. If scene is Lambertian, the lightfield can be described with a single image – it is sufficient to compute all N-bounce light fields

3. They use explicit interreflection operator – prohibitive cost unless using low resolution

Optical Computing for Fast Light Transport Analysis

M. O'Tool and K. Kutulakos

SIGGRAPH Asia 2010

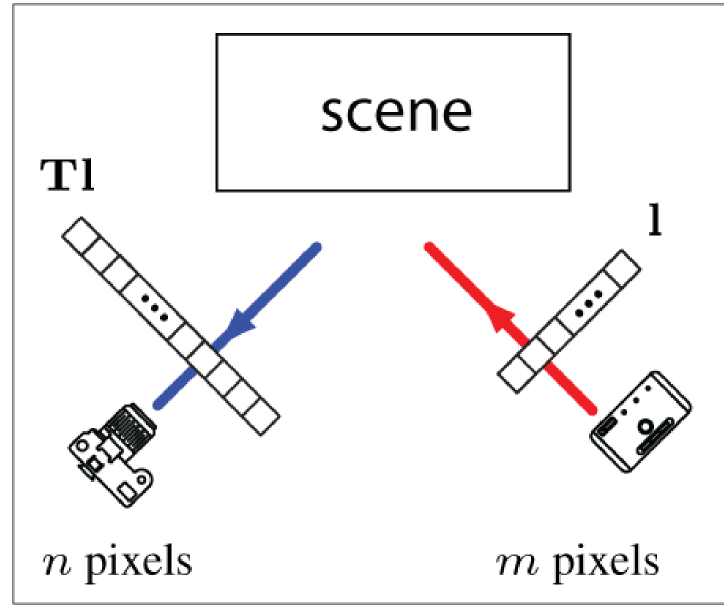
Presented by Vasily Volkov
UC Berkeley CS294-69

Problem statement

Scene is lit using light $\boldsymbol{l}$

Scene's photo is read as $\boldsymbol{p}$

Linear connection: $\boldsymbol{p} = T\boldsymbol{l}$



Relighting:

- Given projector light $\boldsymbol{l}$, estimate resulting image $\boldsymbol{p}$

Inverse light transport:

- Given image $\boldsymbol{p}$, find what light $\boldsymbol{l}$ produces it

The approach

The *transport matrix* T can be very large

- Capturing it explicitly has prohibiting cost

Solution: capture a low-rank approximation

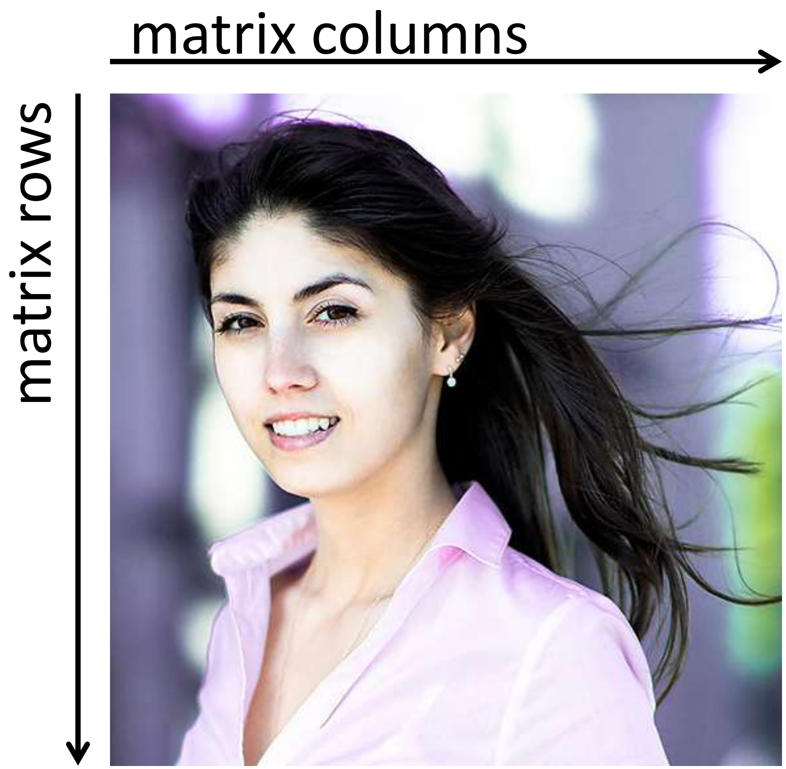
- E.g. rank-10: $T \approx \boldsymbol{p}_1 \boldsymbol{l}_1^T + \boldsymbol{p}_2 \boldsymbol{l}_2^T + \cdots + \boldsymbol{p}_{10} \boldsymbol{l}_{10}^T$
- Inexpensive and might be accurate enough

Products $\boldsymbol{p}_1 \boldsymbol{l}_1^T$, … are never computed explicitly

- Instead, use $T\boldsymbol{l} \approx \boldsymbol{p}_1(\boldsymbol{l}_1^T \boldsymbol{l}) + \cdots + \boldsymbol{p}_{10}(\boldsymbol{l}_{10}^T \boldsymbol{l})$

Closer look into low-rank approximations

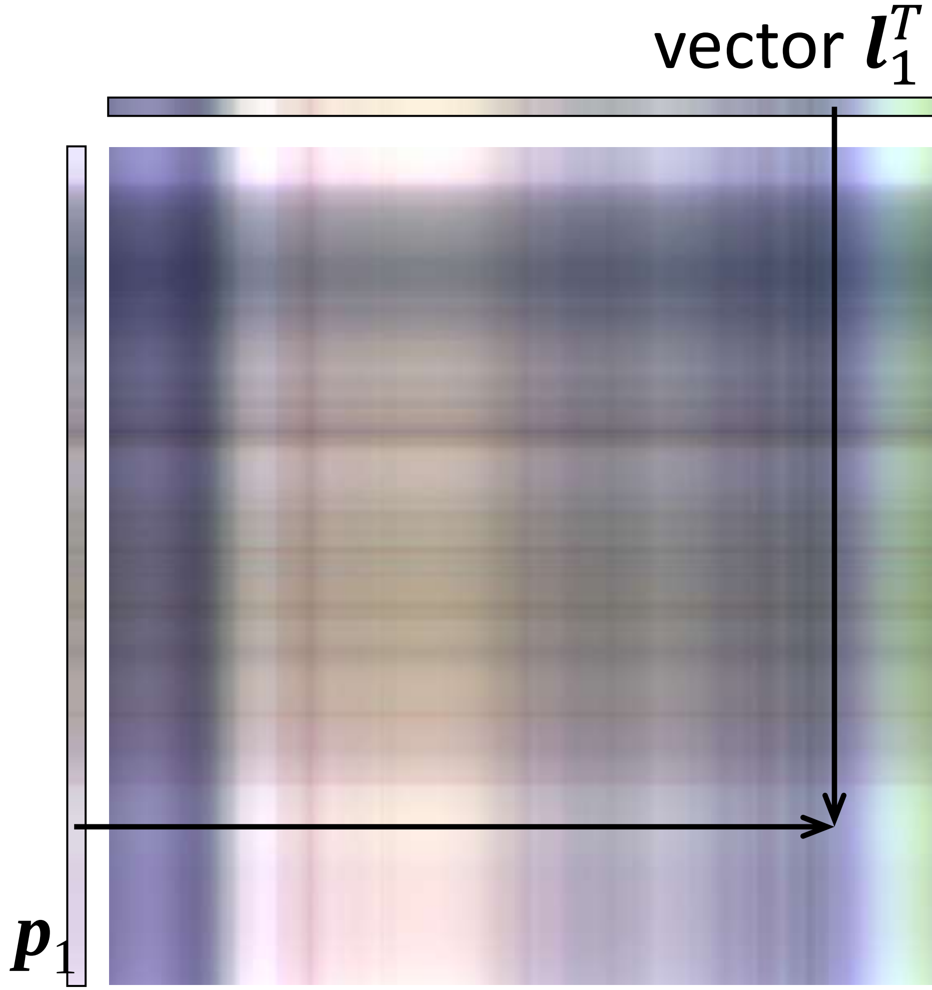
Take an arbitrary matrix A, e.g. this one:



Matrix entries a_{ij} are shown as color intensities
(These are 3 independent matrices for R,G,B)

Rank-1 approximation

matrix A



rank-1 approximation

$$A \approx \boldsymbol{l}_1 \boldsymbol{r}_1{}^T$$

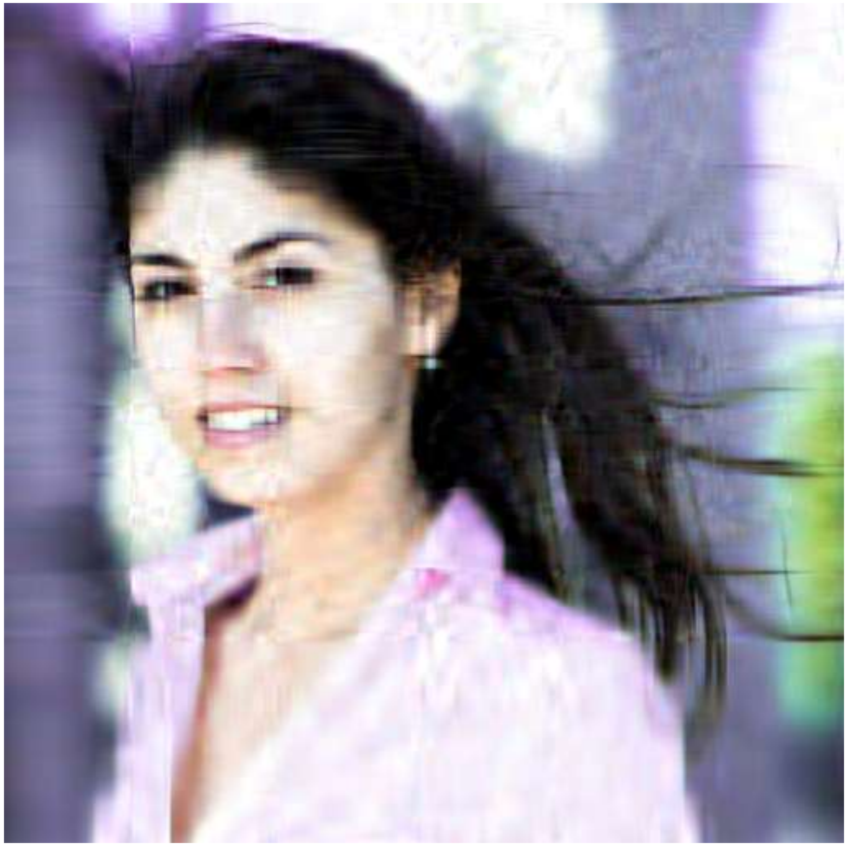
Higher rank = better approximation

rank-5

rank-10

Rank-5 approximation: $A \approx \boldsymbol{l}_1\boldsymbol{r}_1^T + \boldsymbol{l}_2\boldsymbol{r}_2^T \ldots + \boldsymbol{l}_5\boldsymbol{r}_5^T$

Stop when accuracy is sufficient



rank-25

rank-50

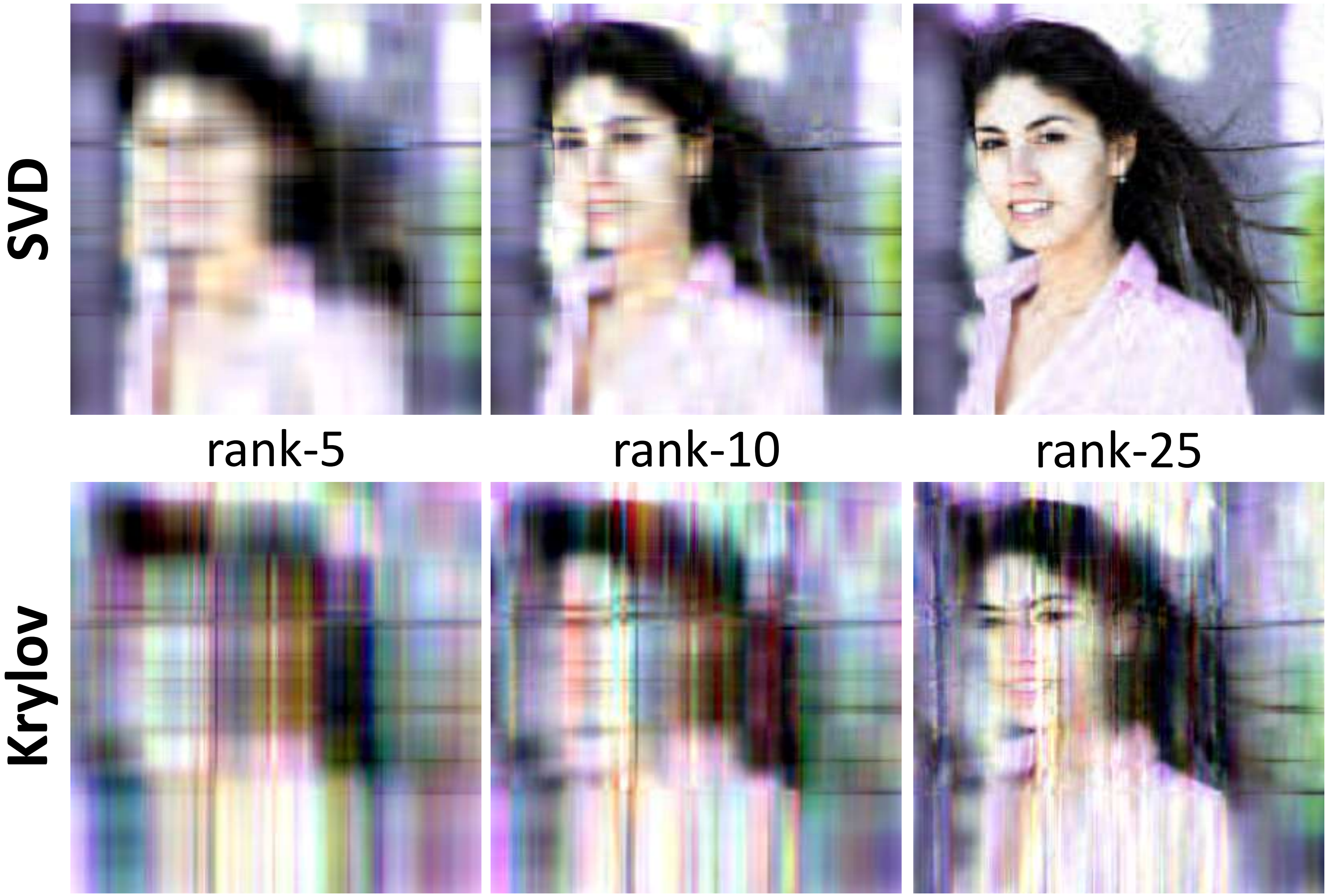
How to compute it? These were done using SVD:

- Most accurate, but requires explicit matrix

Efficient solution: Krylov methods

- Take random vector $\boldsymbol{r}_1$
- Compute $[\boldsymbol{r}_1, A\boldsymbol{r}_1, A^2\boldsymbol{r}_1, \ldots, A^{k-1}\boldsymbol{r}_1]$
- Orthonormalize them – these are $[\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_k]$
  - Interleave these two steps if in finite precision
- Take $[A\boldsymbol{r}_1, A\boldsymbol{r}_2, \ldots, A\boldsymbol{r}_k]$ for $[\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_k]$

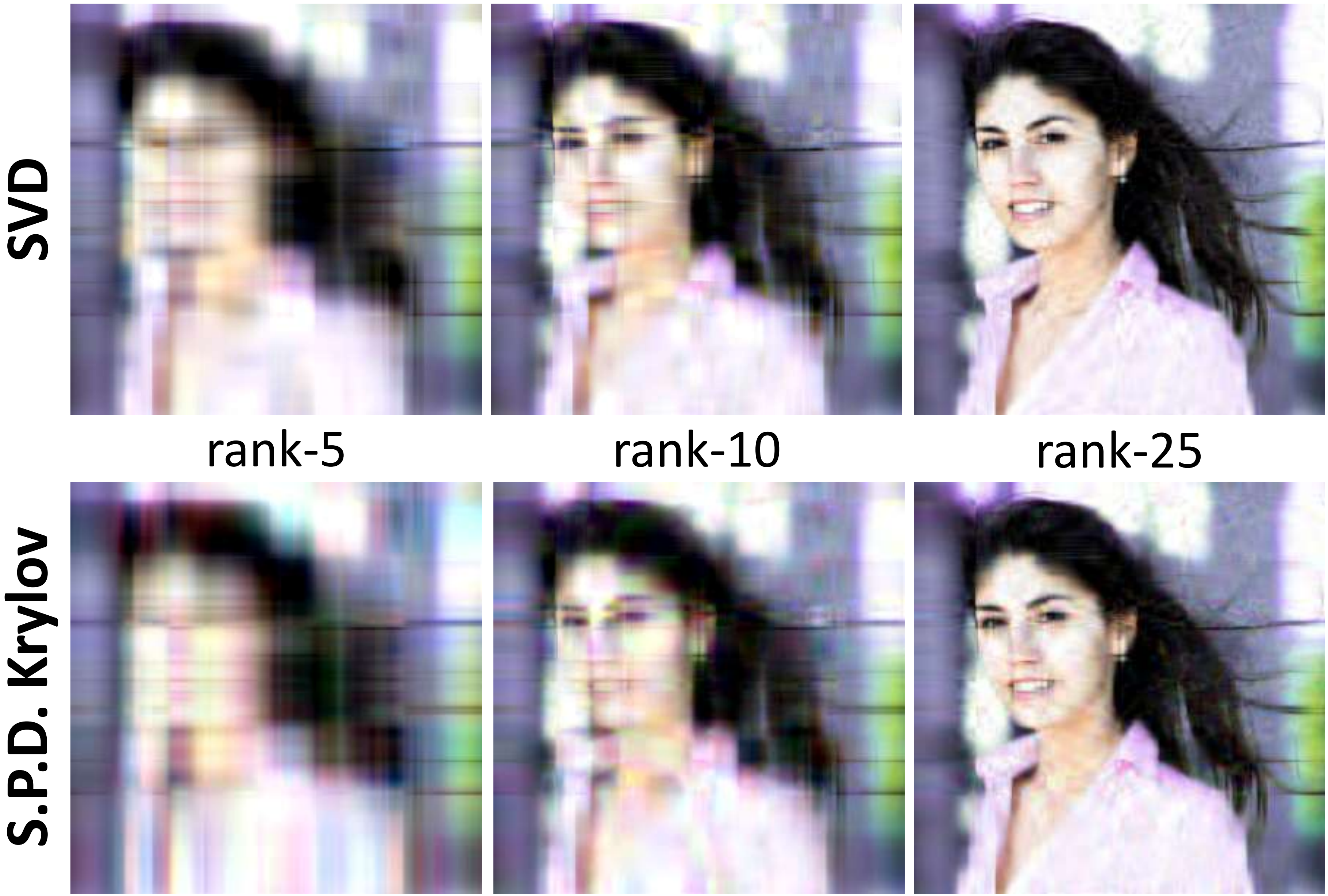
- Now $A \approx \boldsymbol{l}_1\boldsymbol{r}_1^T + \boldsymbol{l}_2\boldsymbol{r}_2^T + \cdots + \boldsymbol{l}_k\boldsymbol{r}_k^T$
- No explicit A needed, only function $A \cdot \boldsymbol{x}$


SVD
rank-5
rank-10
rank-25
Krylov

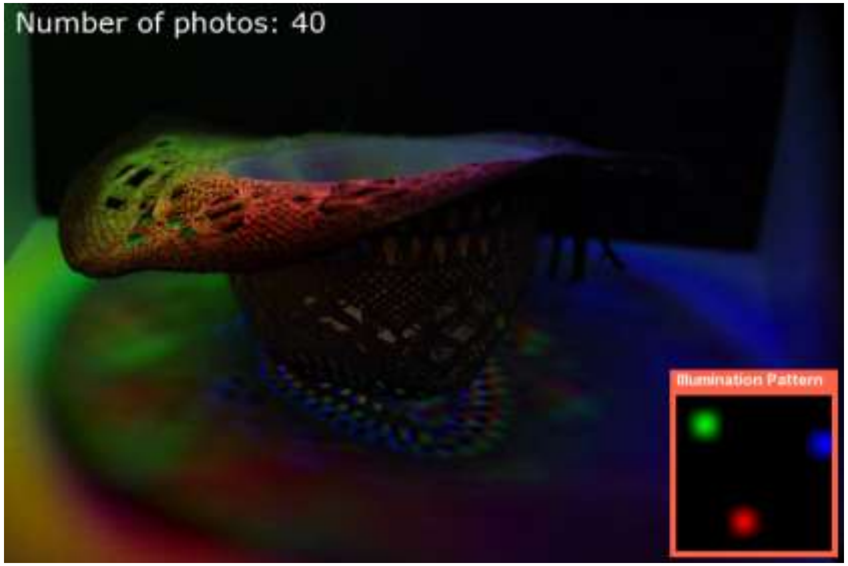
Faster convergence: use A^TA

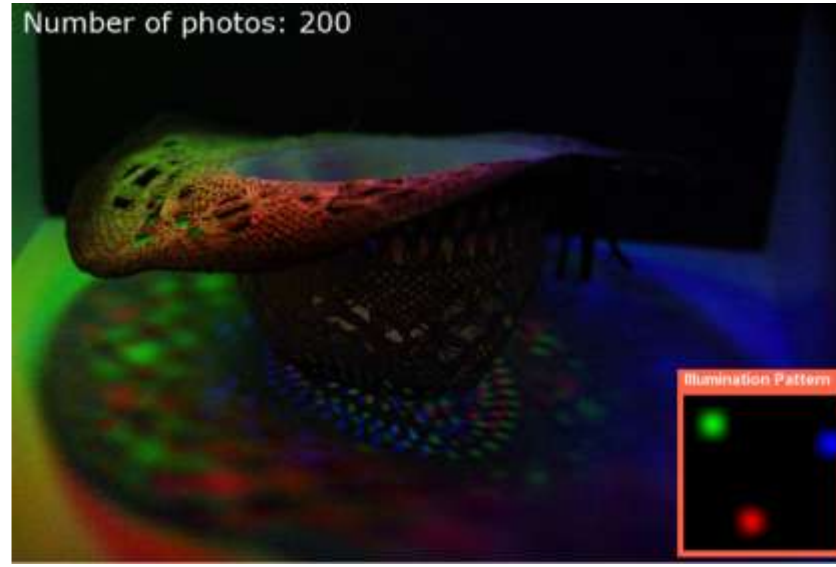
- Use A^TA when building Krylov subspace
  - I.e. compute $\boldsymbol{r}_1$, $A^TA\boldsymbol{r}_1$, $(A^TA)^2\boldsymbol{r}_1$, …
  - A^TA is s.p.d. – much better numerical properties

- This requires additional function $A^T \cdot \boldsymbol{x}$


SVD
rank-5
rank-10
rank-25
S.P.D. Krylov

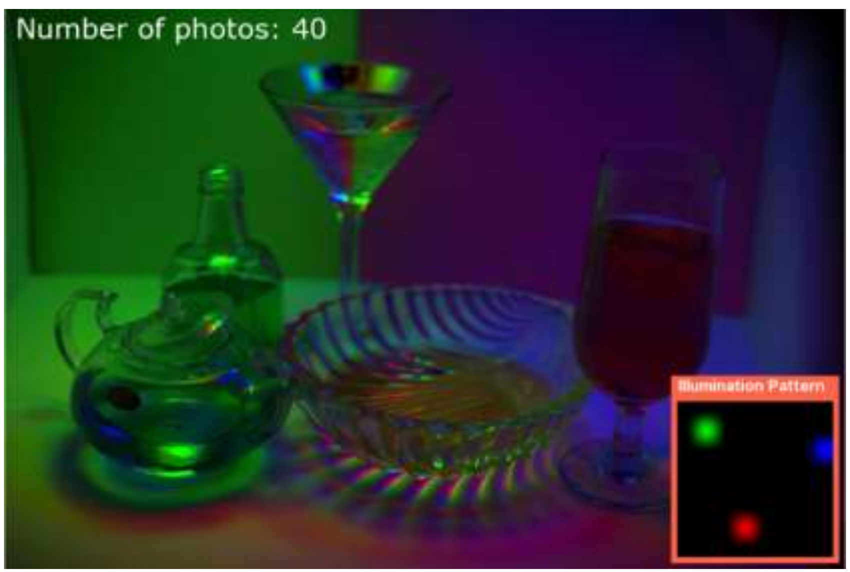
Use same idea for transport matrix

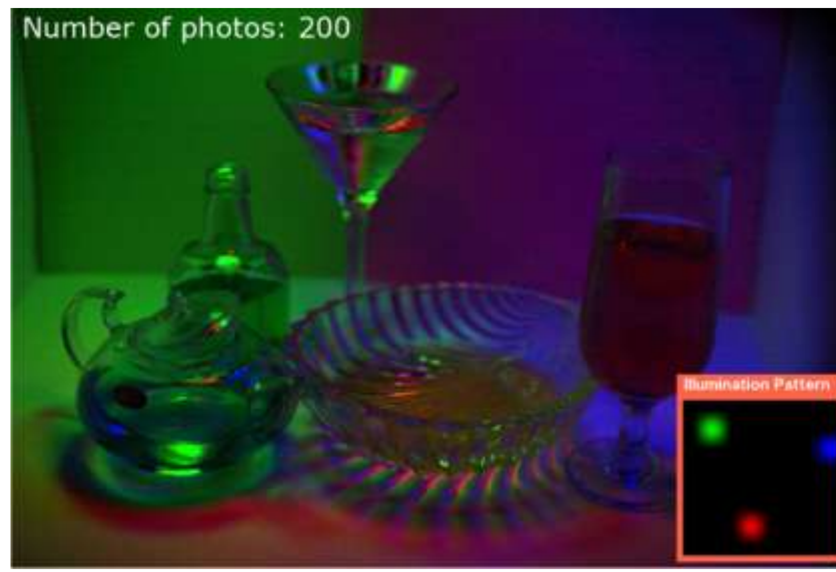




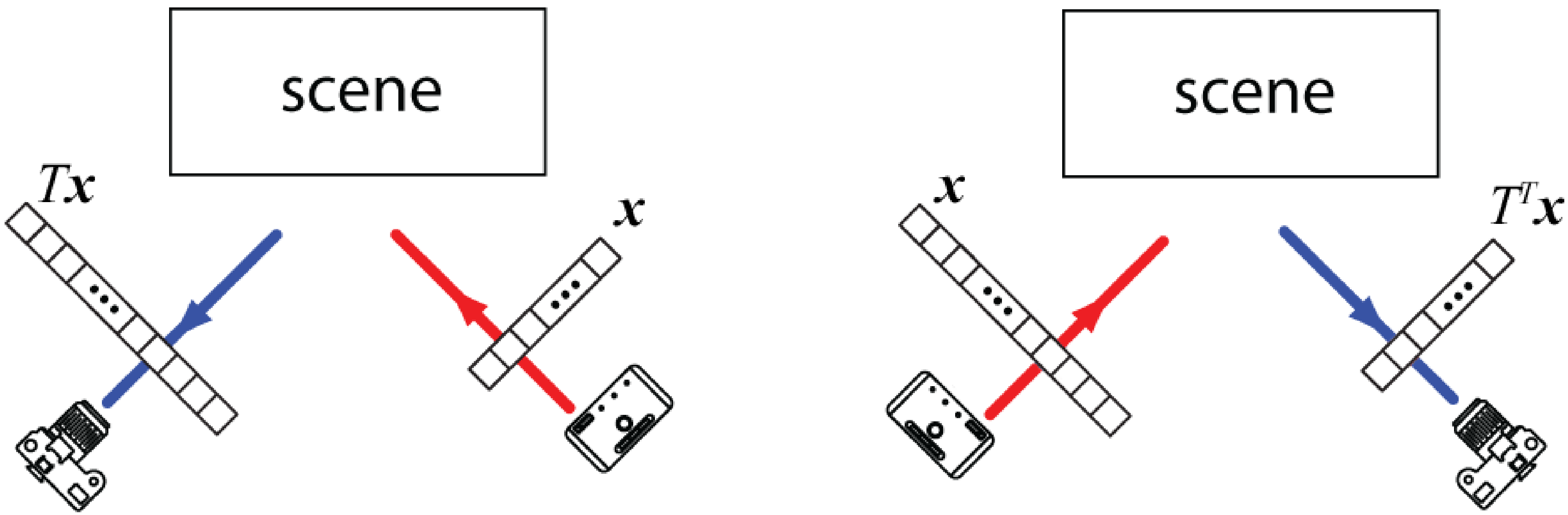
rank-10 approximation

rank-50 approximation





Optical matrix-vector multiply

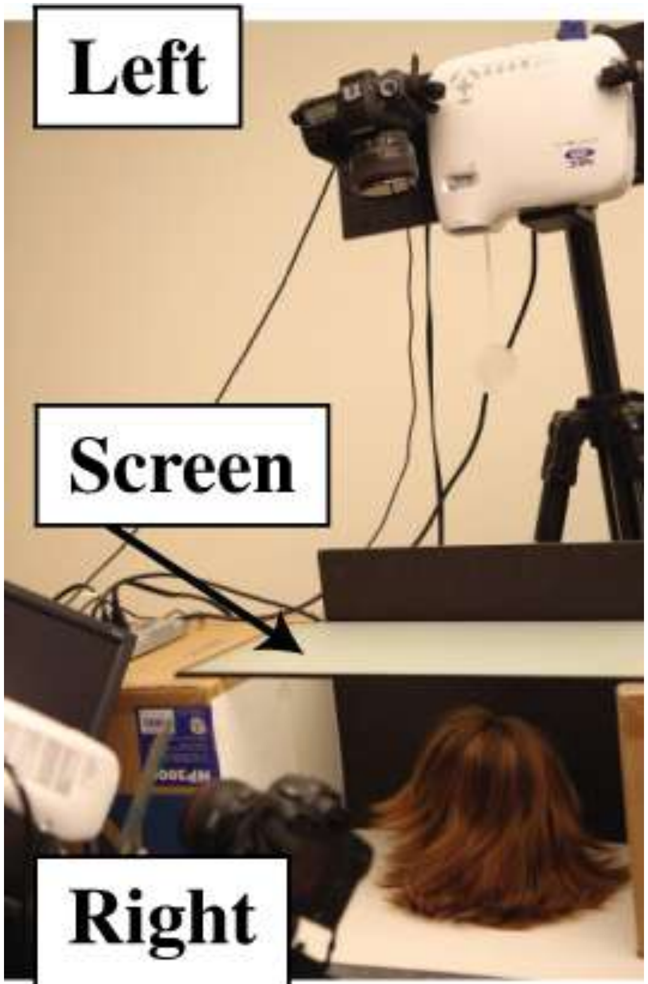


Capturing $T\mathbf{x}$ Capturing $T^T\mathbf{x}$

$T^T\mathbf{x}$: same arrangement as $T\mathbf{x}$, but swap camera and projector

- Or use two of each with beam splitters

Optical matrix-vector multiply



Project x using left projector, read Tx in right camera
Project x using right projector, read $T^T x$ in left camera

Intricacies

Might not work with high-rank T

- Use diffusive light: shoot it through a screen

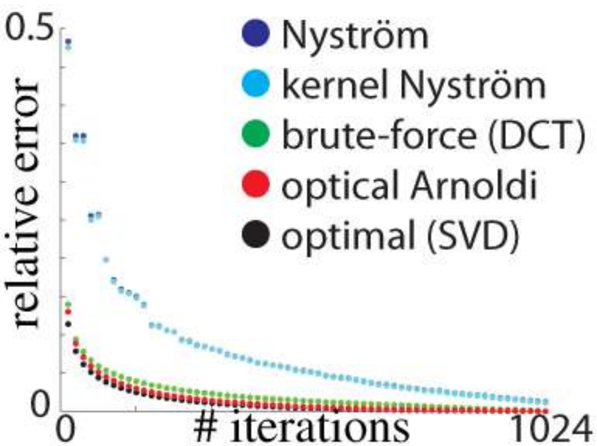
Some vectors have negative pixel values

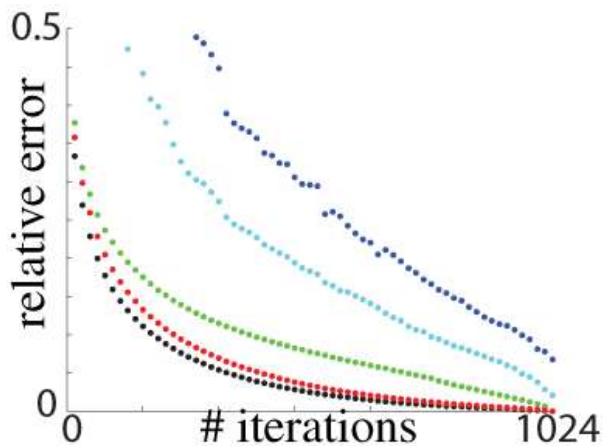
- Process positive and negative separately
- Doubles number of photos

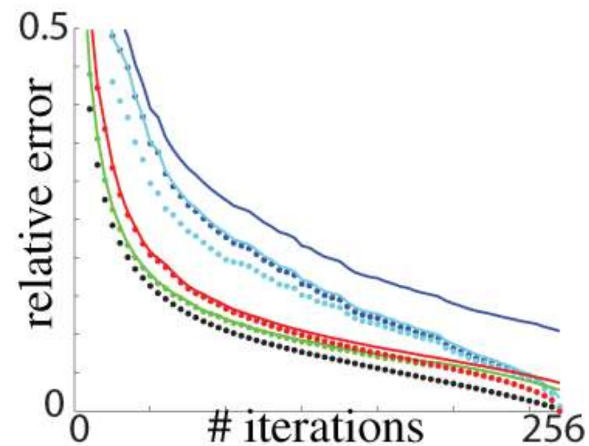
Convergence of matrix approximation

Acquire low-resolution transport matrices explicitly

Compare Krylov with SVD and brute-force







≈ as brute-force

better than brute-force

brute-force is better

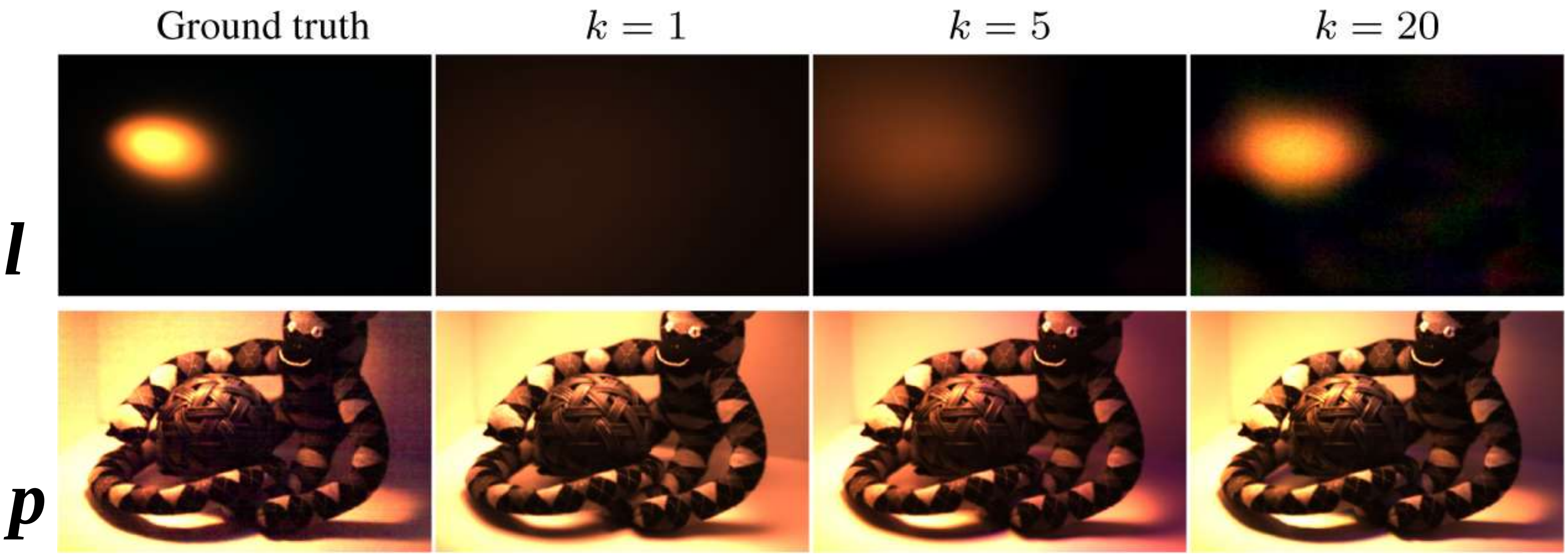
Relighting results

Build rank-10 approximation of T

- Requires 40 photos, 3 seconds per photo

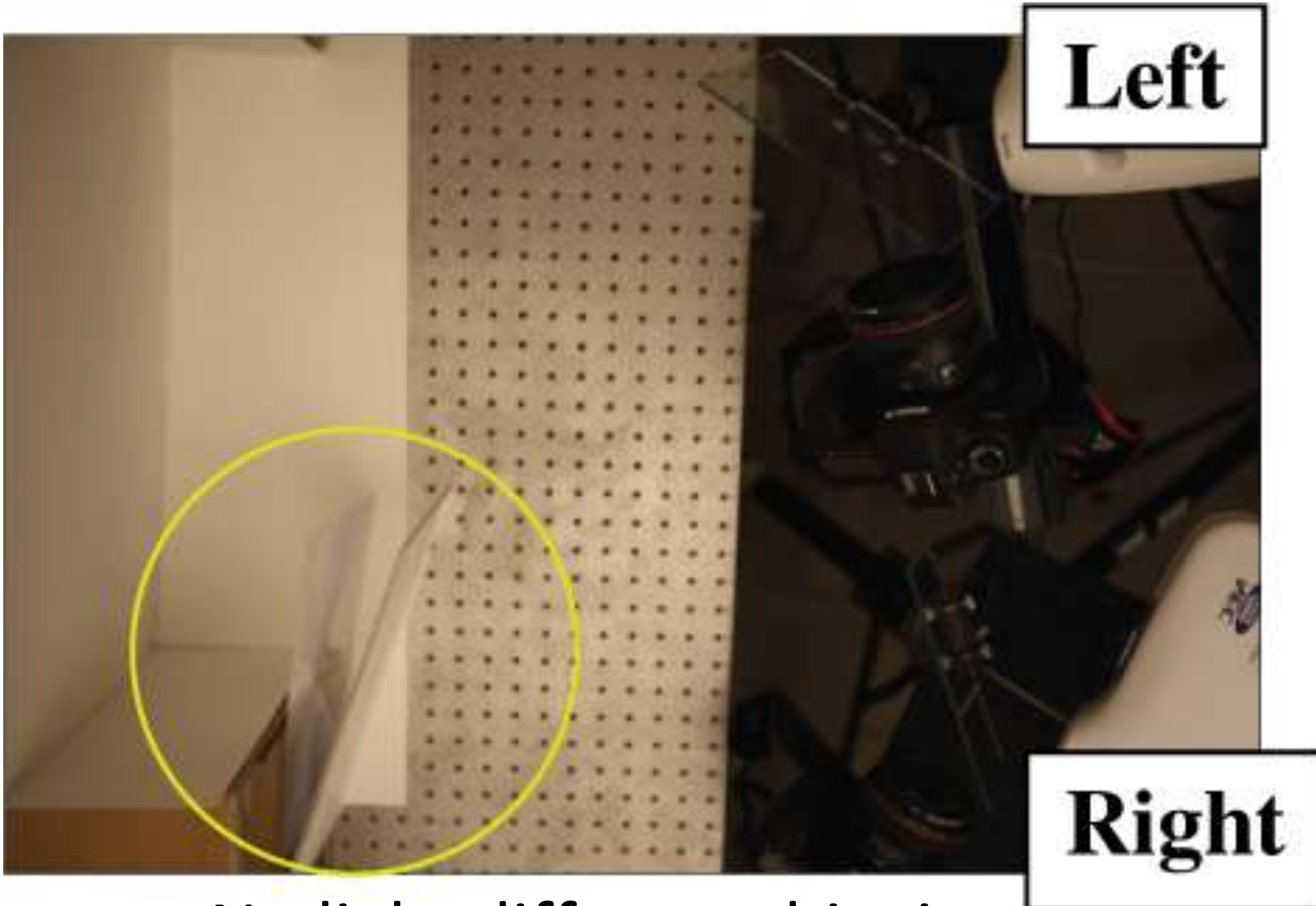
Use it to *compute* $T\mathbf{l}$ for any given $\mathbf{l}$, in 3 seconds

Inverse light transport results



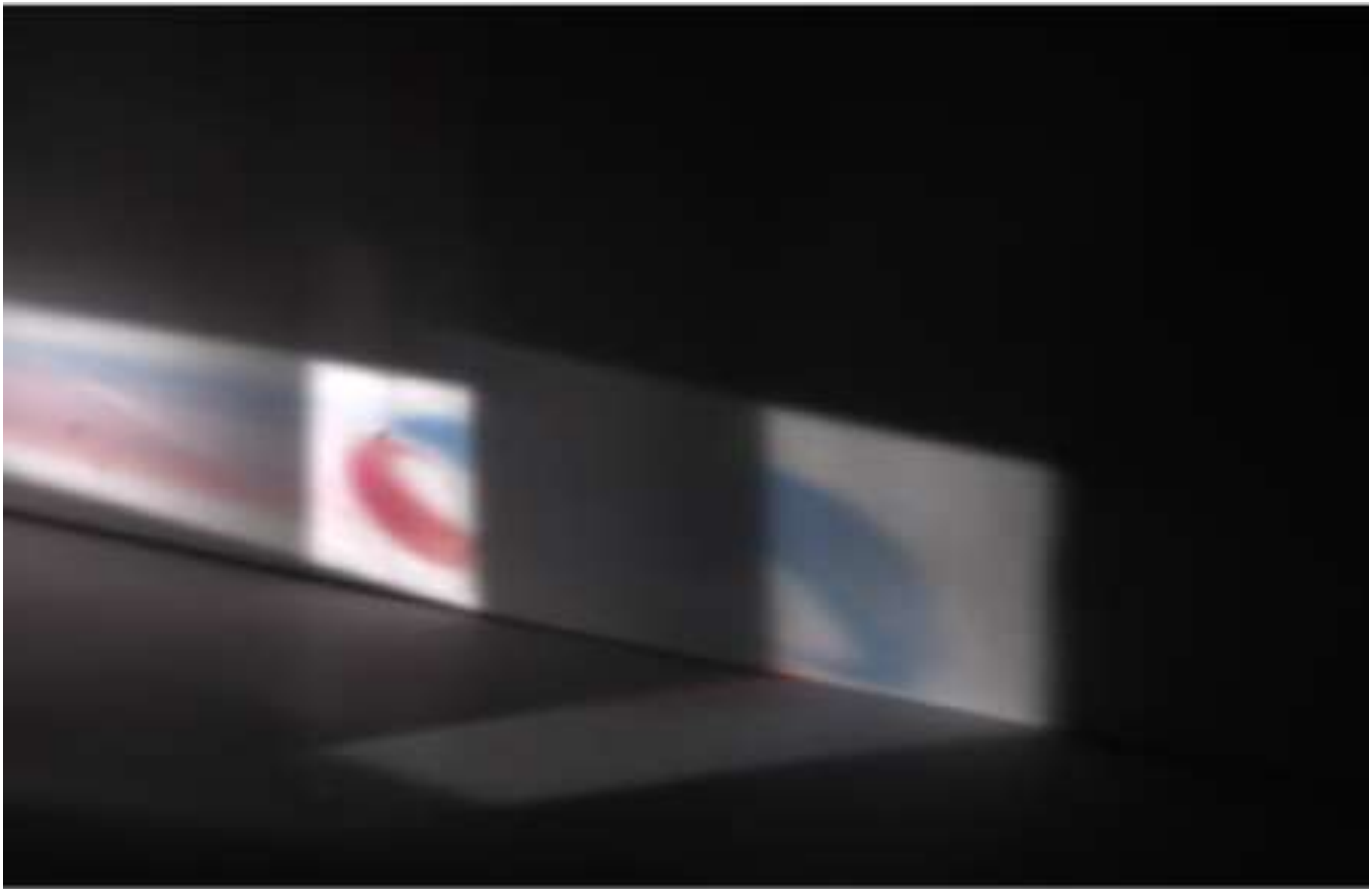
- Use given image $\boldsymbol{p}$ to initiate the Krylov method
  - The resulting approximation is tuned to $\boldsymbol{p}$
  - Might even work well with a high-rank matrix
- Use rank-k approximation to solve $\boldsymbol{p} = T\boldsymbol{l}$ for $\boldsymbol{l}$

Inverse problem for high-rank matrix



No light diffusers this time

The input image



Ground truth

The solution found

(20 iterations)

Video

http://www.youtube.com/watch?v=fVBICVBEGVU#t=2m12s