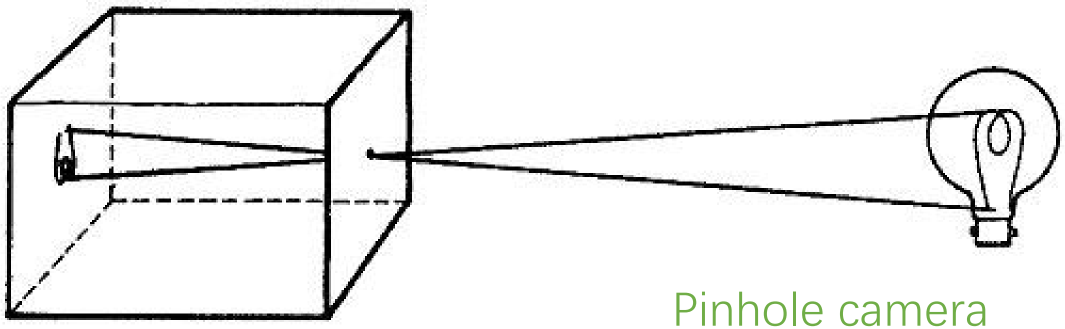
FlatCam: Thin, Lensless Cameras Using Coded Aperture and Computation

Author: M. Salman Asif,
Ali Ayremlou,
Aswin Sankaranarayanan,
Ashok Veeraraghavan,
and Richard G. Baraniuk

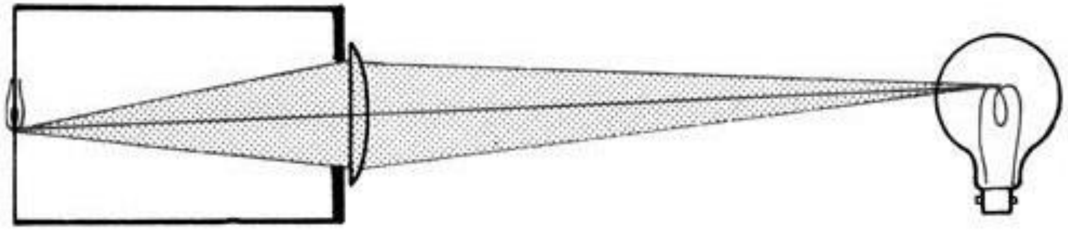
Presenter: Wang Zi

- Introduction: from pinhole to FlatCam
- Imaging Model
- FlatCam Design
- FlatCam Calibration
- Image Reconstruction
- Experimental Results and discussion

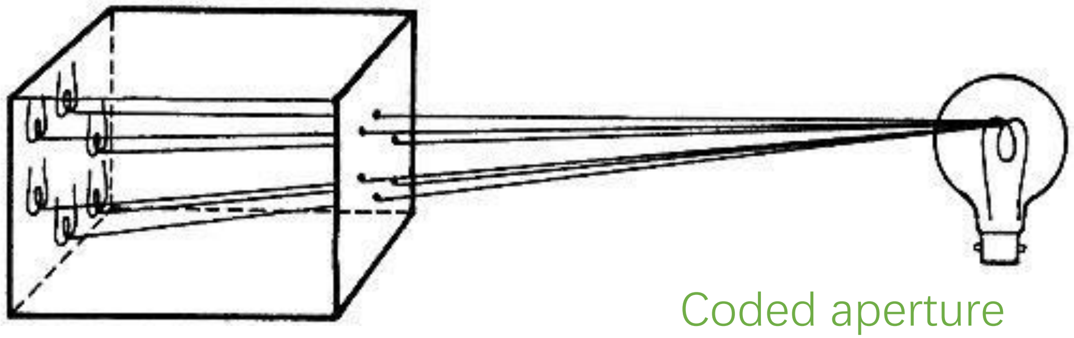
Introduction: from pinhole to FlatCam



Pinhole camera



Lens camera



Coded aperture

Limitation: low amount of light

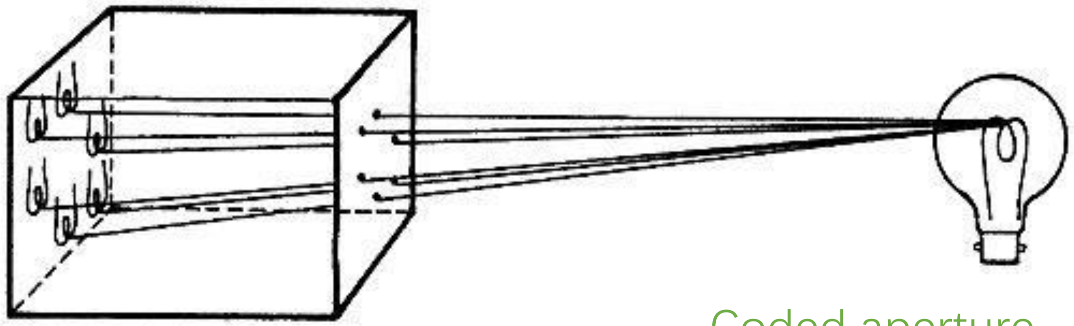
Improvement:

① Lens camera

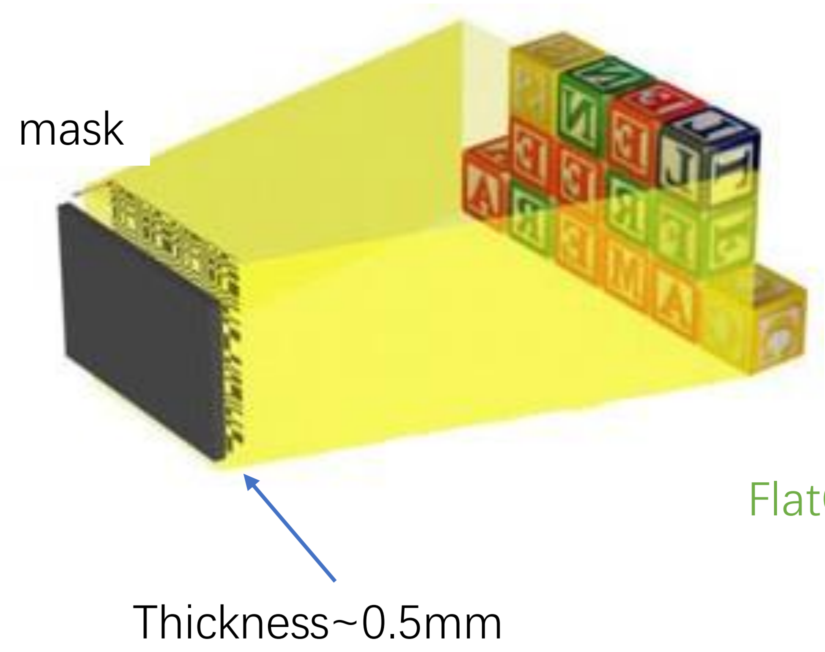
② Increase the number of pinhole

Reconstruction is needed

Introduction: from pinhole to FlatCam



Coded aperture



FlatCam

Flat is very thin!

↓

The size of FlatCam is small

↓

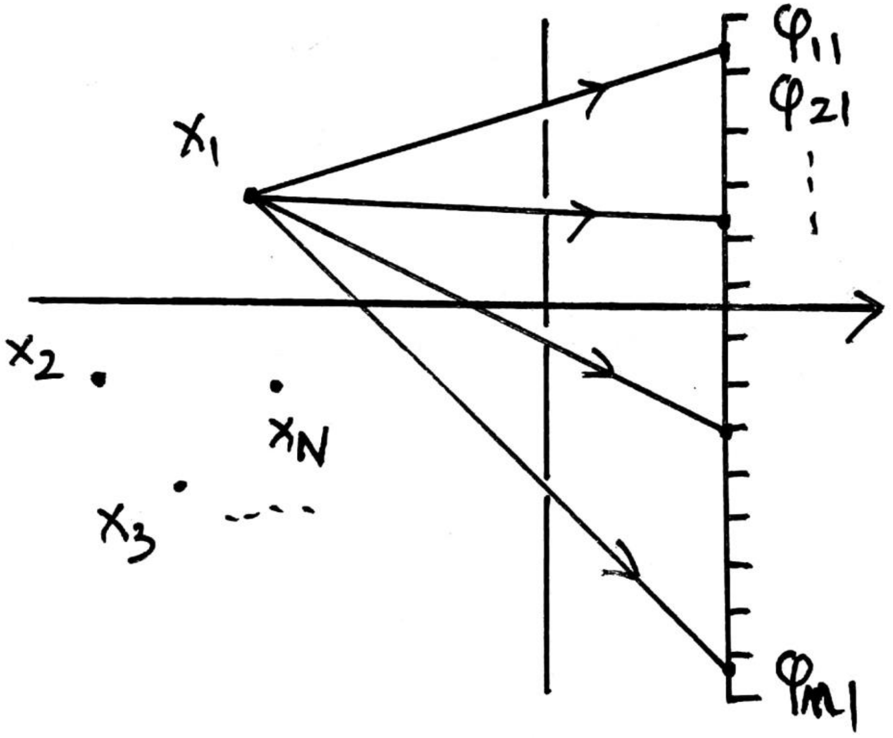
larger amount of light can be achieved

So how did they realize it?

By design of the mask

Imaging Model

1-D linear model:



The response is

$$y = \varphi_1 x_1 + \varphi_2 x_2 + \cdots + \varphi_N x_N = \Phi x$$

With noise it becomes

$$y = \Phi x + e$$

Imaging Model

2-D linear model:

Notify $x \in R^{N^2 \times 1}$, $y = \in R^{M^2 \times 1}$, $\Phi \in R^{M^2 \times N^2}$, we still have

$$y = \Phi x + e$$

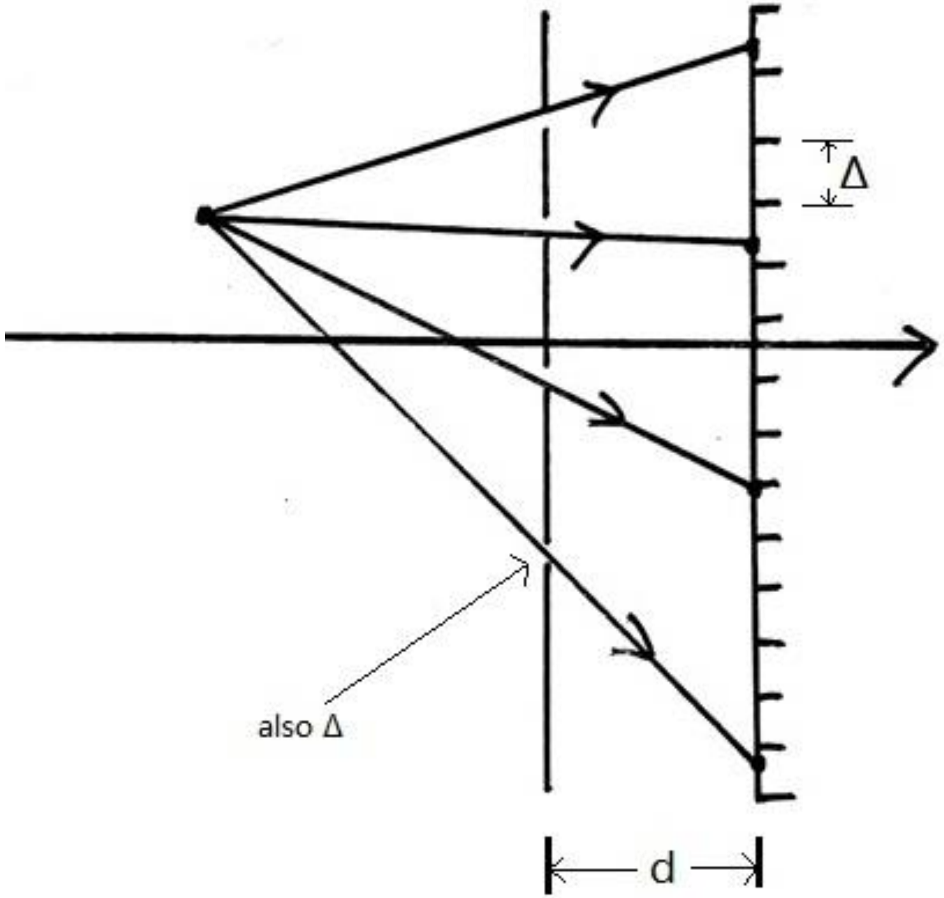
When reconstructing image, we can easily do $\min_x ||y - \Phi x||$ and obtain x

Now the question:

How to design a proper Φ and small thickness? –FlatCam design

How to obtain Φ ? – FlatCam calibration

FlatCam Design



Parameters that affects Φ:

Mask pattern: the shape of the mask

d: the distance from sensor plane to coded mask

Δ: feature size, usually also sensor size

Main performance what we concern about:

Light amount

Computational cost

Calibrate and characterize

Singular value of Φ → Reconstruction: $\min_{x} ||y - \Phi x||$

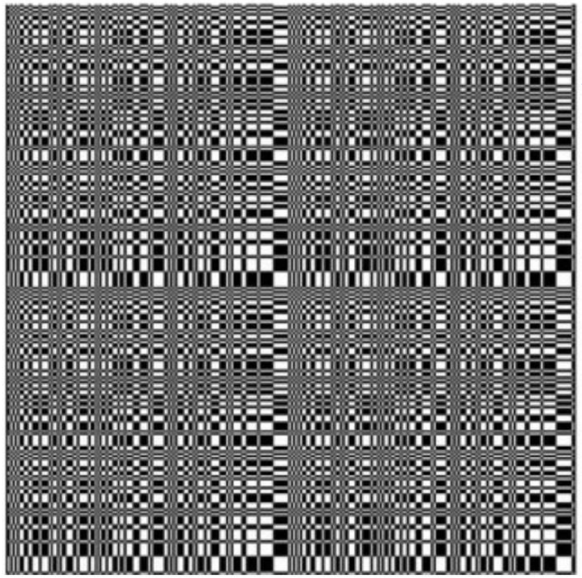
FlatCam Design

Mask Pattern:

1. Light throughput :

the more transparent features,

the more light throughput

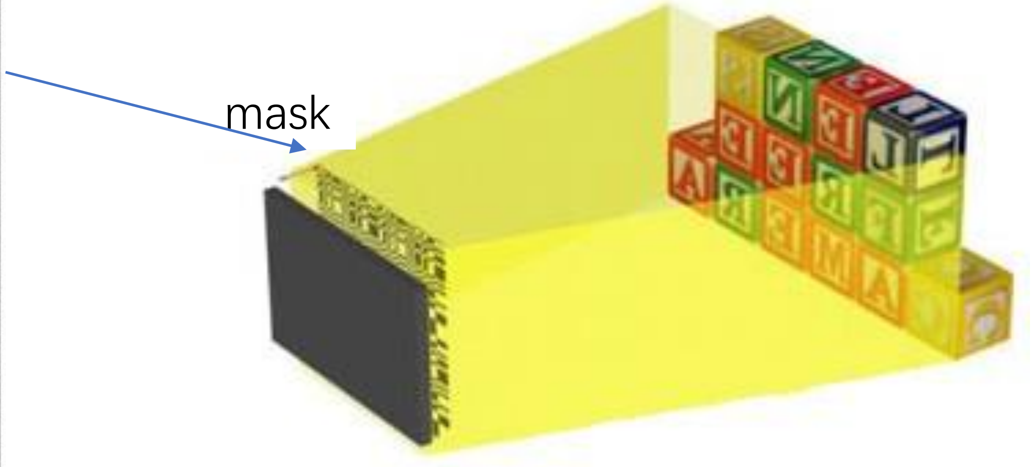


White: transparent feature,

light can go through

Black: opaque feature,

light can not go through



FlatCam Design

Mask Pattern:

2. Computational complexity:

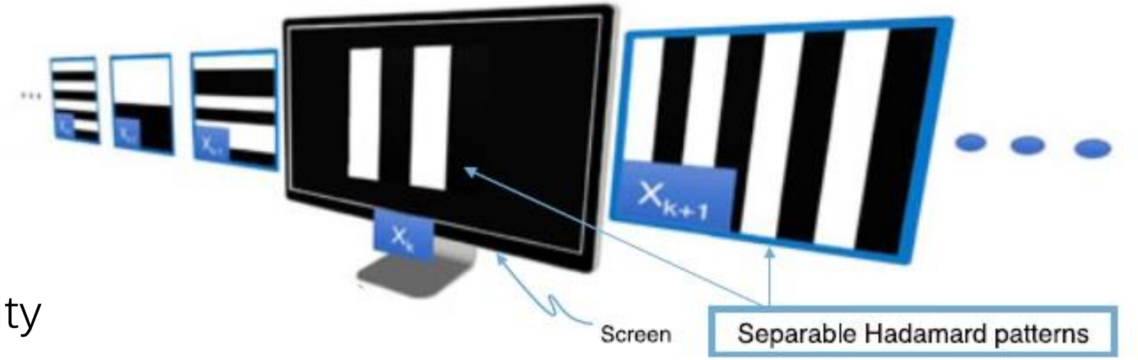
$$y = \Phi x + e$$

The complexity of $\Phi \in R^{M^2 \times N^2}$ is $O(M^2N^2) \approx O(N^4)$

According to the paper, if we design a separable mask, (mask matrix is rank-1), the model can be rewritten as

$$Y = \Phi_L X \Phi_R^T + E$$

Where $\Phi_L , \Phi_R \in R^{M \times N}$Then the computational complexity is $O(MN) \approx O(N^2)$



3. Calibration

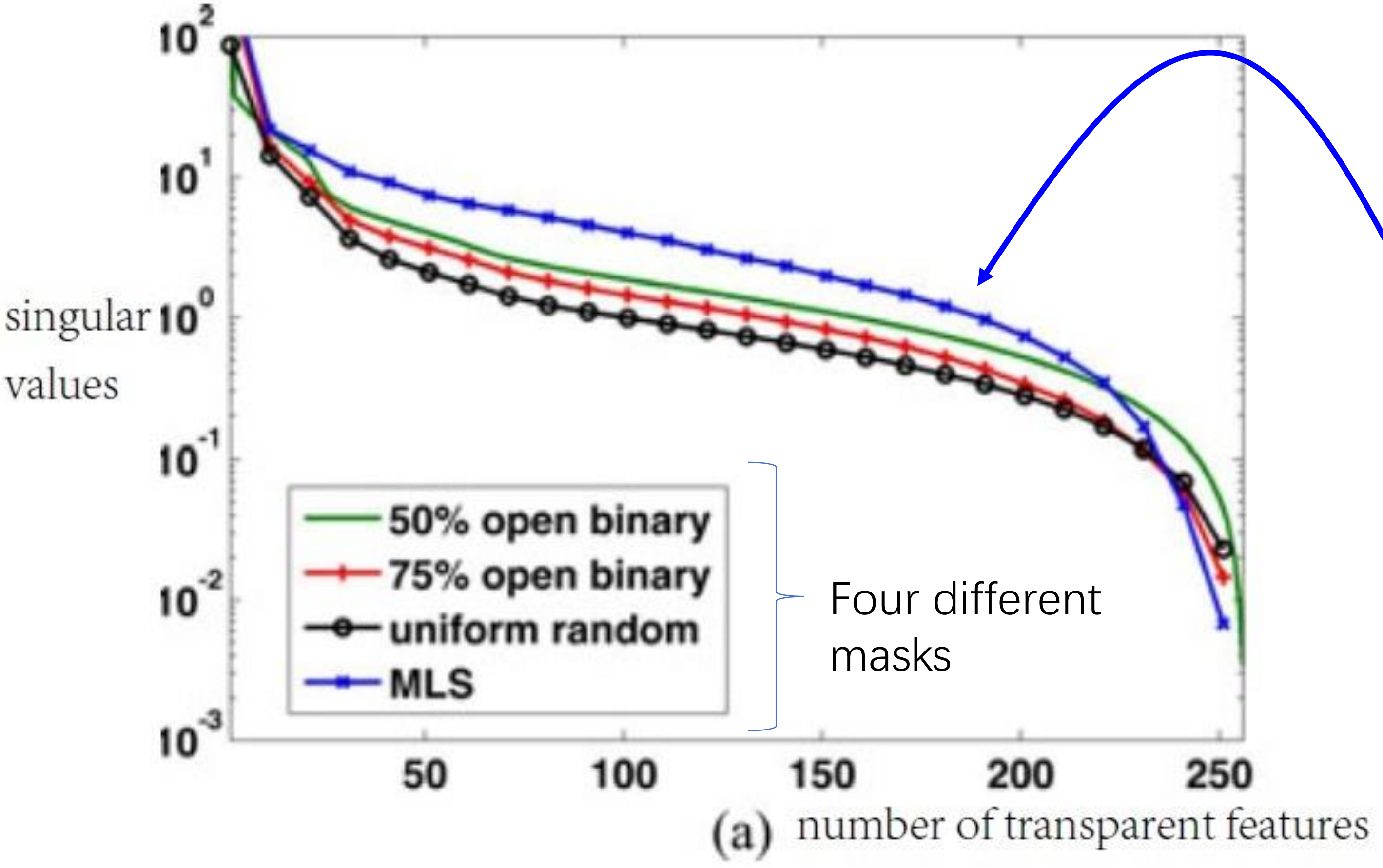
separable mask is very easy to calibrate and characterize

(will be shown in calibration part)

FlatCam Design

Mask Pattern:

4. Singular value of Φ



Four different masks

We want:

High singular values

High number of transparent features

Separable mask

So we choose MLS

Q

FlatCam Design

Mask Placement $\boldsymbol{d}$ and Feature Size $\boldsymbol{\Delta}$

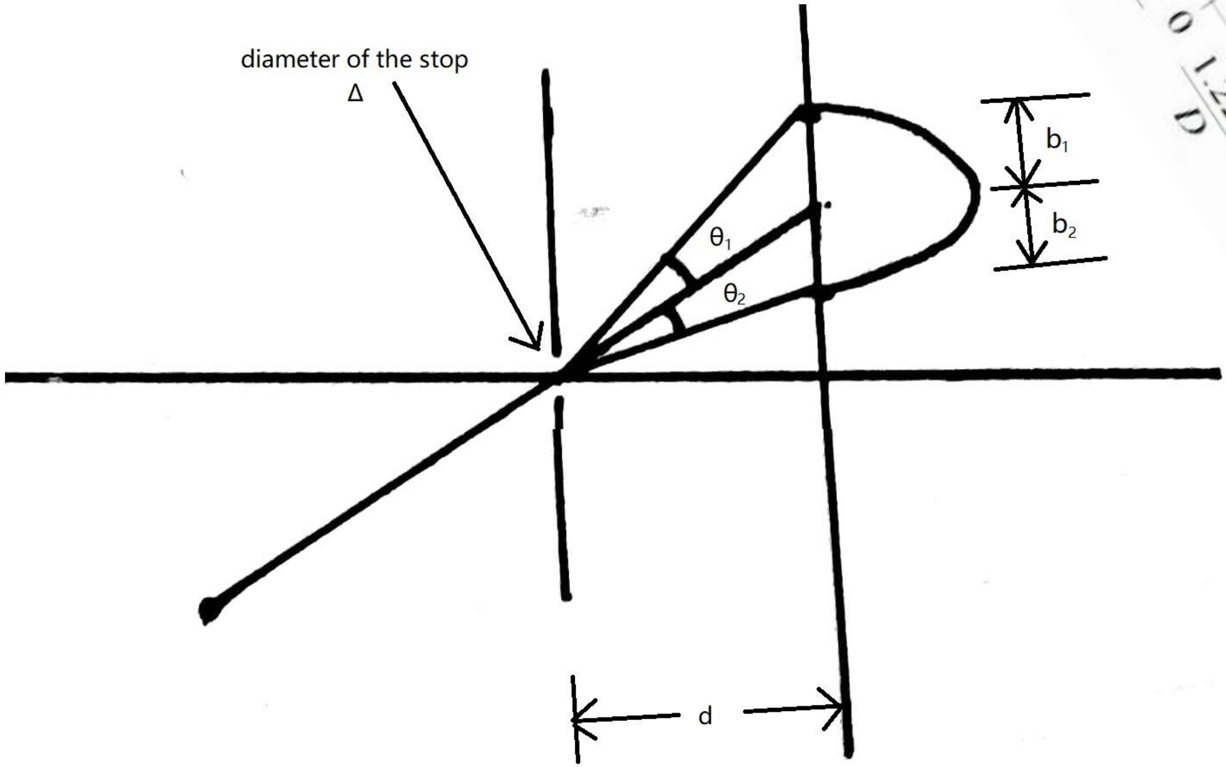
1. Minimize the total blur

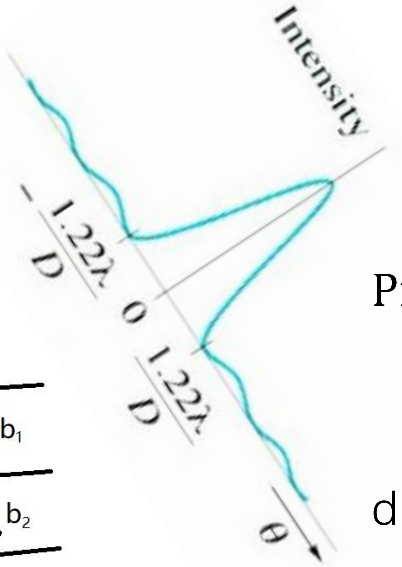
   Size of diffraction blur: $2.44\lambda d/\Delta$

   Size of geometric blur: Δ

   When $\Delta = \sqrt{2.44\lambda d}$, the blur is minimal for a fixed d

diffraction blur





Principle of diffraction:

$$\theta_1 \approx \theta_2 \approx \frac{1.22\lambda}{\Delta}$$

diffraction blur:

$$= b_1 + b_2$$

$$\approx \mathrm{dsin}(\theta_1 + \theta_2)$$

θ very small so view triangle as a arc, and line segment as d

$$= \mathrm{dsin}\left(\frac{2.44\lambda}{\Delta}\right)$$

$$\approx d\frac{2.44\lambda}{\Delta}$$

θ very small so take first-order approximation

FlatCam Design

Mask Placement d and Feature Size Δ

1. $\Delta = \sqrt{2.44\lambda d}$

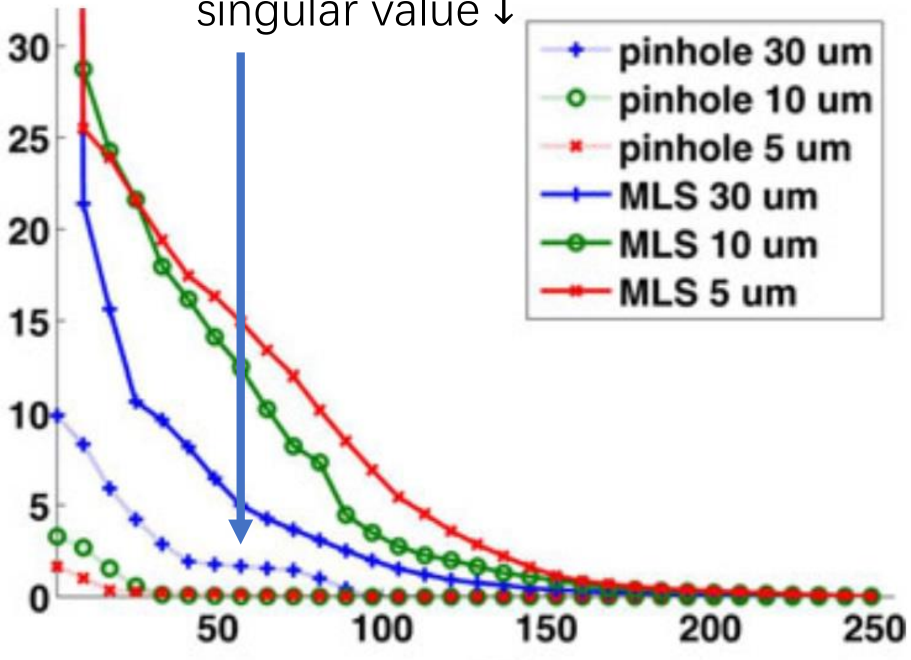
2. enabling sufficient multiplexing to obtain a well-conditioned linear system

   $\Delta \uparrow \longrightarrow$ extent of multiplexing $\downarrow \longrightarrow$ singular value $\downarrow$

There is a trade off between total blur and condition number!

$\Delta \uparrow$

singular value $\downarrow$



(c)

Q

FlatCam Calibration

Known many pairs of X and Y in

$$Y = \Phi_L X \Phi_R^T + E$$

How to estimate $\boldsymbol{\Phi_L}$ and $\boldsymbol{\Phi_R}$?

Let's use separable scene $X = ab^T$

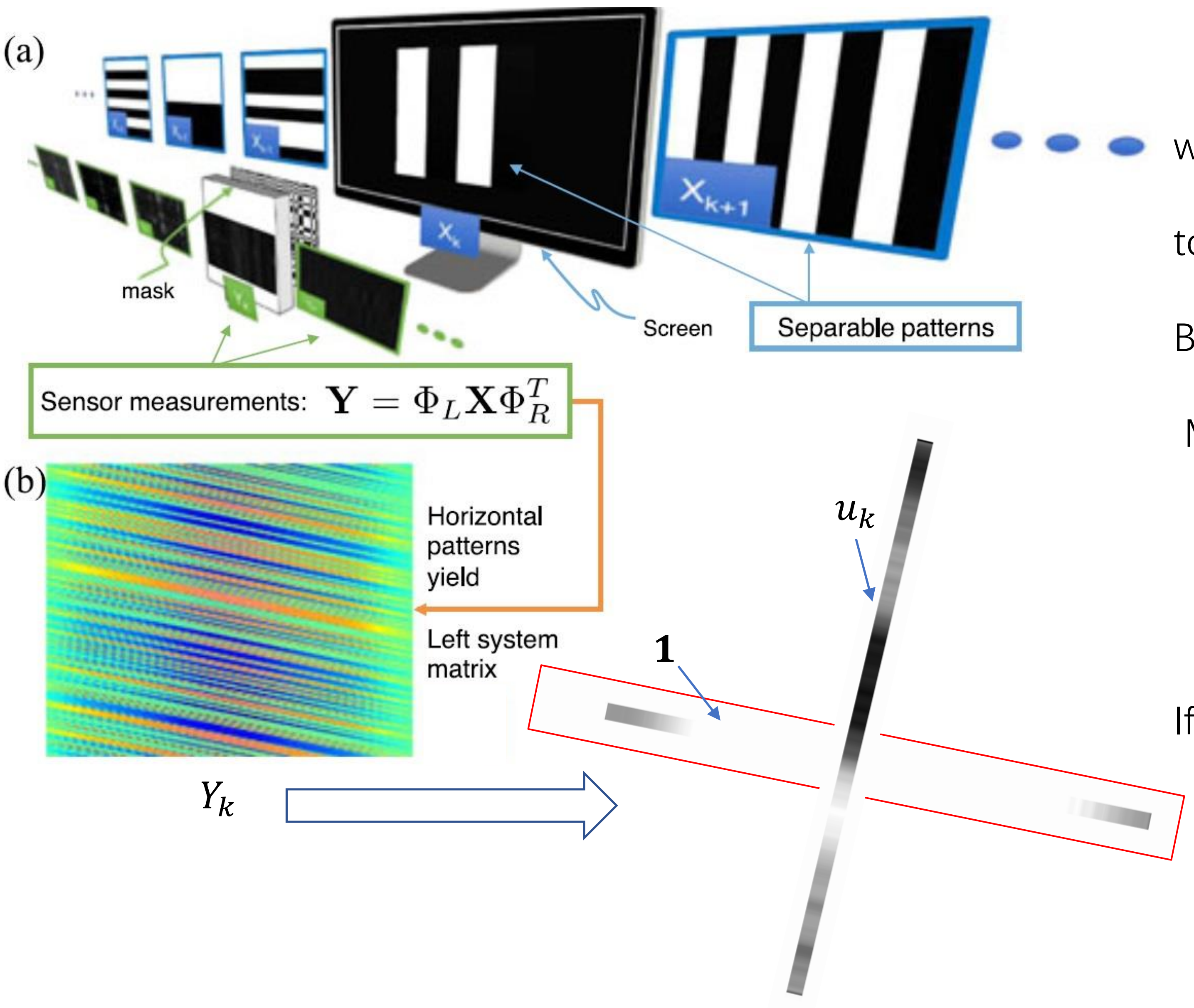
$$Y = (\Phi_L a)\,(\Phi_R b)^T + E$$

By apply SVD to Y, we can obtain $(\Phi_L a)$ and $(\Phi_R b)$ very accurately

FlatCam Calibration

$$Y = \Phi_L X \Phi_R^T + E$$

$$Y = (\Phi_L a)\,(\Phi_R b)^T + E$$



we use $X_k = a_k \mathbf{1}^T$

to generate $Y_k = (\Phi_L a_k)\,(\Phi_R \mathbf{1})^T + E$, $k = 1,2,\dots,N$

By SVD we obtain $u_k = (\Phi_L a_k)$

Matrix form is

$$[u_1, u_2, \dots, u_N] = [\Phi_L a_1, \Phi_L a_2, \dots, \Phi_L a_N)]$$

$$U = \Phi_L A$$

If we design H invertible, then

$$\Phi_L = U A^{-1}$$

Q

Image Reconstruction

Now we know the measurements Y,

transfer matrix Φ_L and Φ_R in equation

$$Y = \Phi_L X \Phi_R^T + E$$

How to recover X?

① Noise minimization:

$$\hat{X} = \underset{X}{\operatorname{argmin}} \left\| Y - \Phi_L X \Phi_R^T \right\|_2^2$$

This is a typical least square problem

and the solution is

$$\hat{X} = {\Phi^L}^\dagger Y (\Phi_R^T)^\dagger$$

② Regularized noise minimization:

$$\hat{X} = \underset{X}{\operatorname{argmin}} \left\| Y - \Phi_L X \Phi_R^T \right\|_2^2 + \tau \, \|X\|_2^2$$

The solution can be expressed explicitly with SVD

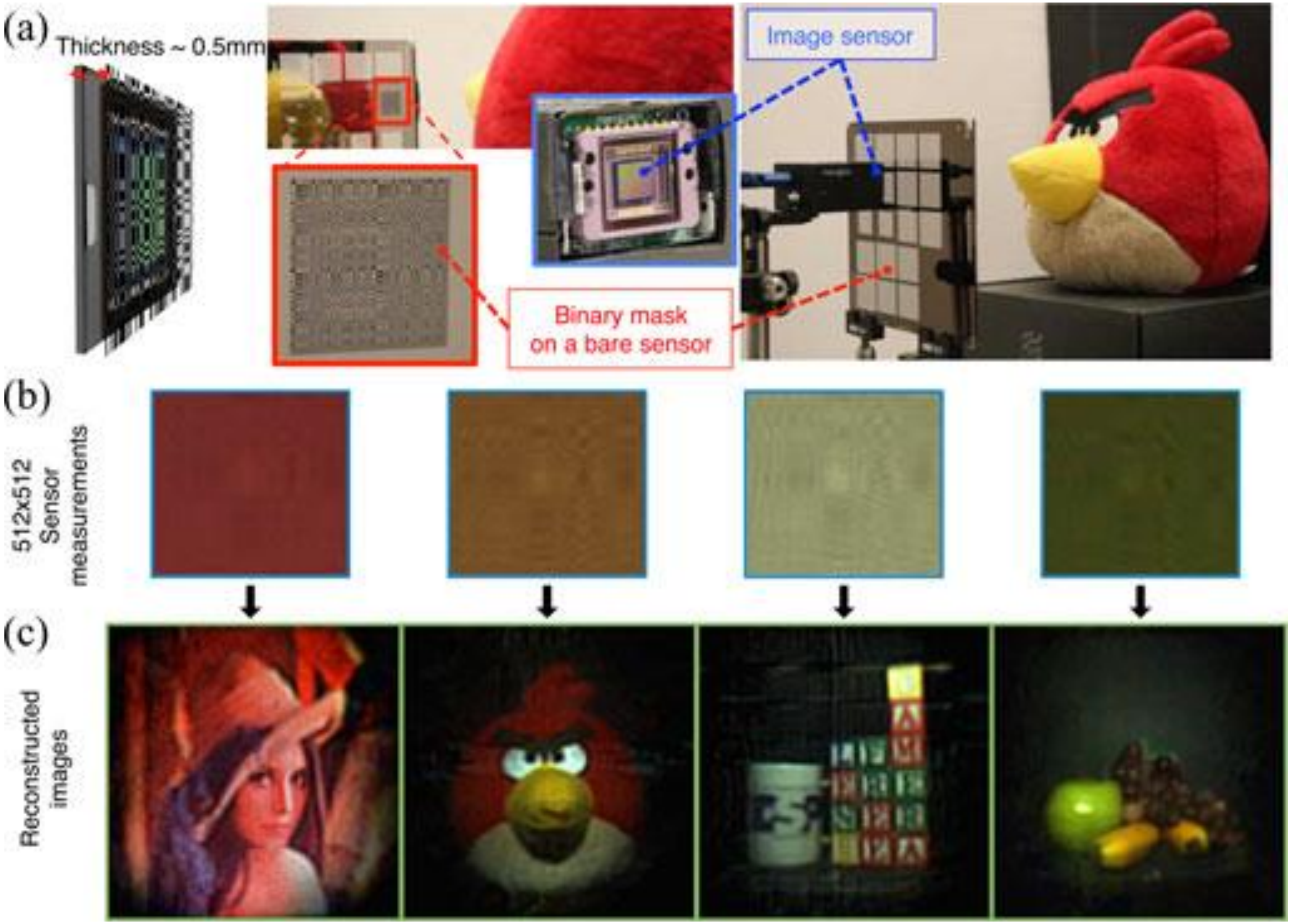
$$\widehat{X}_{\text{Tik}} = V_L [(\Sigma_L U_L^T Y U_R \Sigma_R)./(\sigma_L \sigma_R^T + \tau \mathbf{1}\mathbf{1}^T)] V_R^T$$

③ Total variation (TV) regularized noise minimization

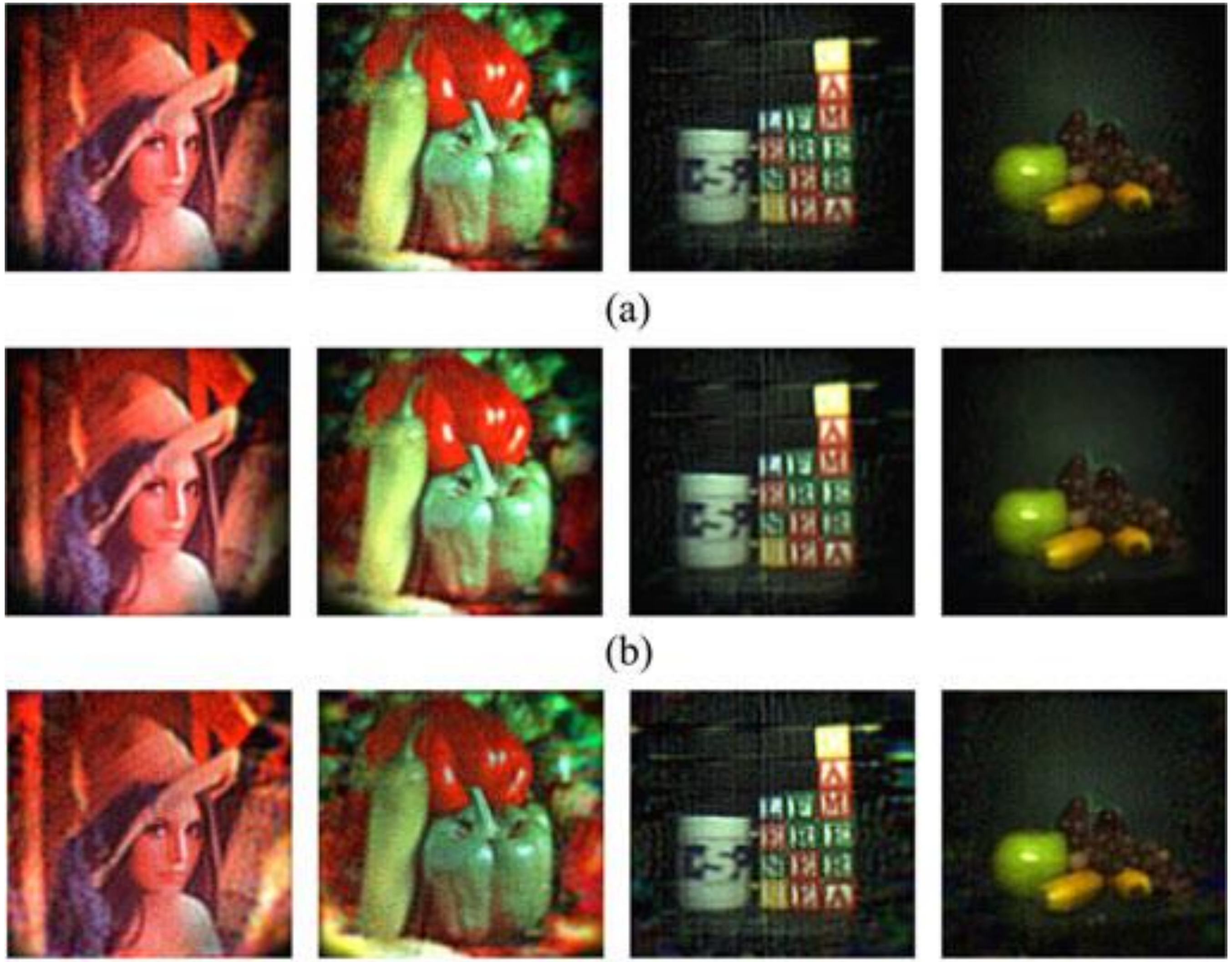
$$\hat{X} = \underset{X}{\operatorname{argmin}} \left\| Y - \Phi_L X \Phi_R^T \right\|_2^2 + \lambda \|X\|_{TV}$$

This is a convex optimization problem

Experimental Results



Experimental Results



(a)

(b)

(c)

① $\hat{X} = \underset{X}{\operatorname{argmin}} \left\| Y - \Phi_L X \Phi_R^T \right\|_2^2$

75ms per image

② $\hat{X} = \underset{X}{\operatorname{argmin}} \left\| Y - \Phi_L X \Phi_R^T \right\|_2^2 + \tau \|X\|_2^2$

10s per image

③ $\hat{X} = \underset{X}{\operatorname{argmin}} \left\| Y - \Phi_L X \Phi_R^T \right\|_2^2 + \lambda \|X\|_{TV}$

75s per image

Experimental Results

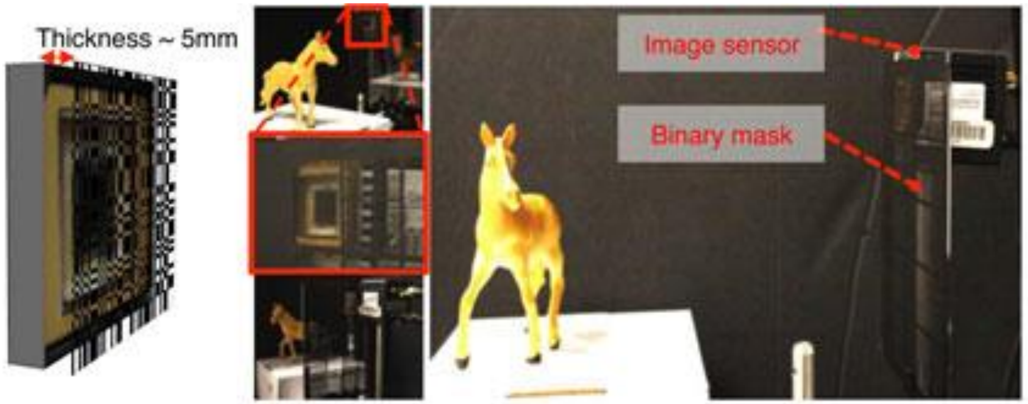
(a)

(b)

30 frames per second

10 frames per second

Experimental Results



(a)

64x64 Reconstructed images
(b)

SWIR FlatCam prototype and results

Experimental Results

Reconstruction in "thin" configuration			
Test image	**02 mask**	**04 mask**	**MLS mask**
Barbara	PSNR = 22.22 dB	PSNR = 22.22 dB	PSNR = 24.54 dB
USAF target	PSNR = 19.27 dB	PSNR = 20.47 dB	PSNR = 24.66 dB
Toys	PSNR = 25.43 dB	PSNR = 25.39 dB	PSNR = 29.1 dB

($d = 500\ \mu$m)

Reconstruction in "thick" configuration			
Test image	**02 mask**	**04 mask**	**MLS mask**
Barbara	PSNR = 27.91 dB	PSNR = 27.29 dB	PSNR = 39.02 dB
USAF target	PSNR = 31.12 dB	PSNR = 28.22 dB	PSNR = 44.52 dB
Toys	PSNR = 34.99 dB	PSNR = 34.61 dB	PSNR = 44.98 dB

($d = 6500\ \mu$m)

Q

Discussion: Advantages

Coded mask very close to the sensor plane	→	Thin, flat camera
MLS: 50% transparent feature mask	→	Large amount of light
Separable mask	→	Simple calibration and reconstruction

Discussion: Advantages

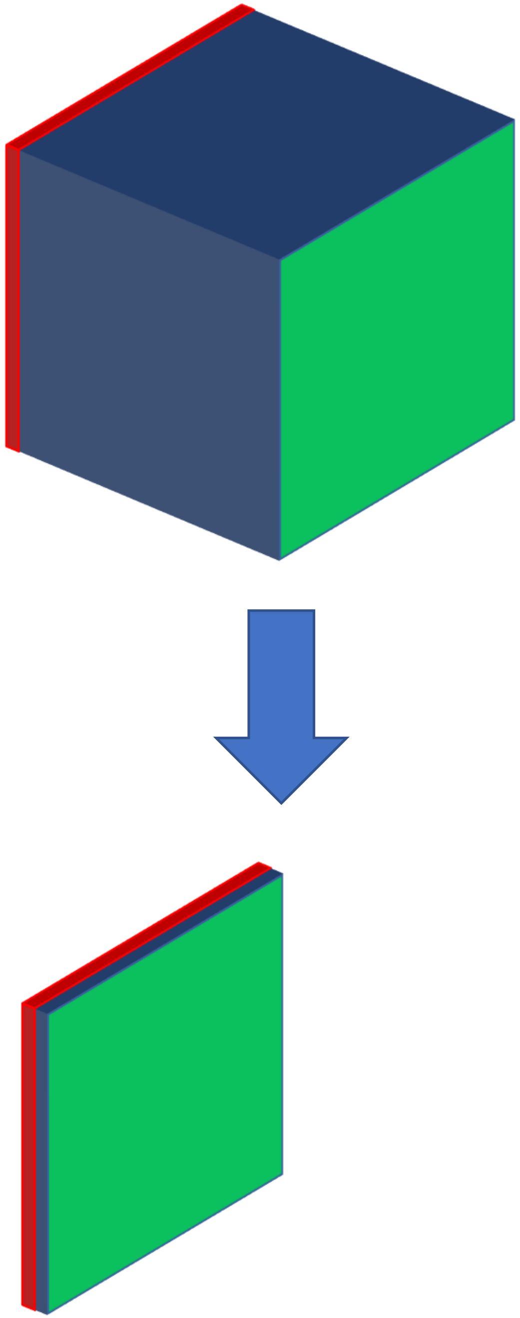
Thin, flat camera

Traditional coded aperture camera:

$$thickness\ width\ ratio(TWR) = \frac{T_{traditional}}{W} \approx 1$$

FlatCam: $T_{Flat} \ll T_{traditional}$

$$(TWR) = \frac{T_{Flat}}{W} \approx 0.075$$



Light amount

θ_{CRA} is determined by the sensor, so

$$L \propto W^2 {N_A}^2$$

N_A ↑: more light for each sensor

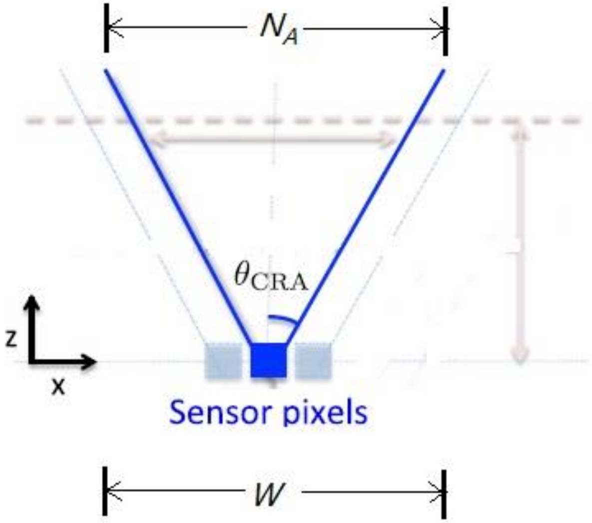
W ↑: more sensors

Since

$$TWR = \frac{T}{W}$$

We have

$$L \propto \frac{T^2 {N_A}^2}{(TWR)^2}$$



When thickness and aperture is the same, Light amount in FlatCam is almost $(\frac{1}{0.075})^2 \approx 178$ times more than traditional coded aperture camera!

Discussion: Limitations

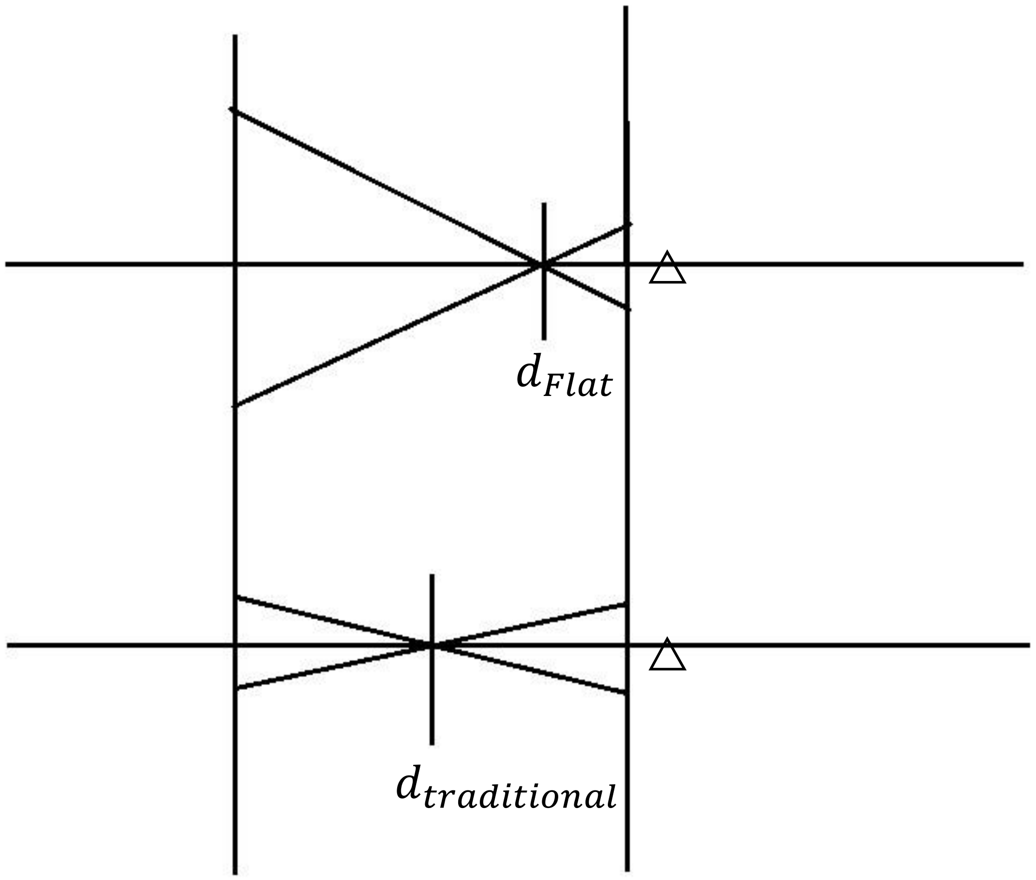
Coded mask very close to the sensor plane → angular resolution decreases

Singular value from mask pattern and thickness → spatial resolution decreases

Time of reconstruction by SVD: 75ms → Not enough real-time

Discussion: Limitations

angular resolution decreases



Flat: small d
Low resolution

Coded aperture: large d
High resolution

Discussion: Limitations

spatial resolution decreases

Mask pattern d $\xrightarrow{\text{affect}}$ Φ_L and Φ_R $\xrightarrow{\text{affect}}$ Singular value of transfer matrix $\xrightarrow{\text{affect}}$ Reconstruction error $\xrightarrow{\text{affect}}$ resolution

Deeper discussion about spatial resolution of computational photography will be in DiffsuerCam

Thank you!

Thank you!

Imaging Model

In fact, if we have

$$y = \Phi x + e$$

Where Φ is a Toeplitz matrix due to **symmetry**, then we can rewrite it into

$$Y = \Phi_L X {\Phi_R}^T + E$$

We will see this model is more efficient later.

When reconstructing image, we can easily do $\min_{x} ||Y - \Phi_L X {\Phi_R}^T||$ and obtain x

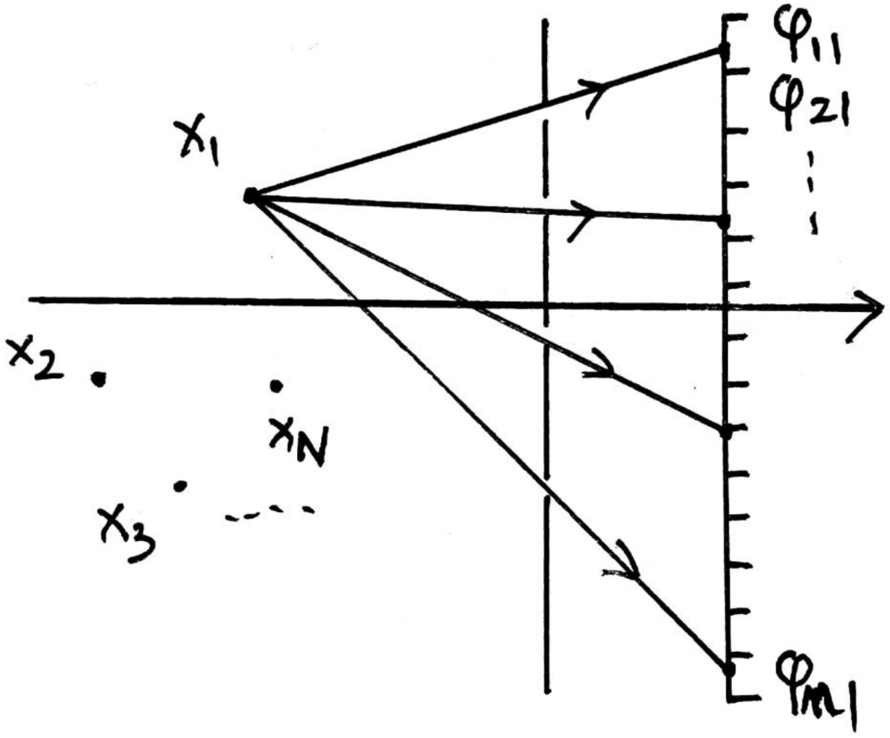
Now the question:

How to design a proper Φ ? –FlatCam design

How to obtain Φ ? – FlatCam calibration

Imaging Model

1-D linear model:



Consider a single point source x_1 and φ_1 is the unit sample response, or PSF

$$\varphi_1 = [\varphi_{11}, \varphi_{21}, \dots, \varphi_{M1}]^T \in R^M$$

The response is

$$y = \varphi_1 x_1$$

Consider many single point source

$$x = [x_1, x_2, \dots, x_N]^T$$

The response is

$$y = \varphi_1 x_1 + \varphi_2 x_2 + \cdots + \varphi_N x_N = \Phi x$$

Where

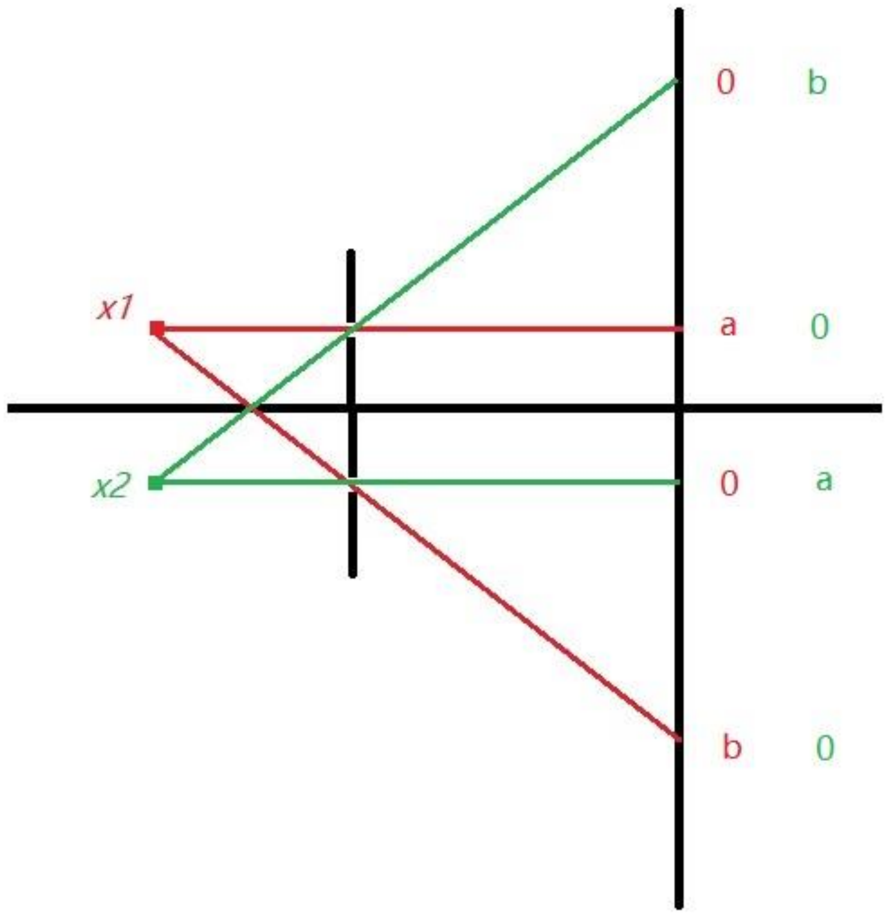
$$\Phi = [\varphi_1, \varphi_2, \dots, \varphi_N] = \begin{bmatrix} \varphi_{11} & \cdots & \varphi_{1N} \\ \vdots & \ddots & \vdots \\ \varphi_{M1} & \cdots & \varphi_{MN} \end{bmatrix}$$

With noise it becomes

$$y = \Phi x + e$$

Imaging Model

Separable mask based 2-D linear model:



Due to the **symmetry**, We have

$$\varphi_1 = [0, a, 0, b]^T$$

$$\varphi_2 = [b, 0, a, 0]^T$$

then

$$\Phi = \begin{bmatrix} 0 & b \\ a & 0 \\ 0 & a \\ b & 0 \end{bmatrix}$$

There is some prior with Φ !

It is a Toeplitz matrix

FlatCam Calibration

First we use $X_k = h_k \mathbf{1}^T$ to generate Y_k, $k = 1,2,\dots,N$

Since $Y_k = (\Phi_L h_k)\,(\Phi_R \mathbf{1})^T + E$, we notify $u_k = (\Phi_L h_k)$,

which is known

Matrix form is

$$[u_1, u_2, \dots, u_N] = [\Phi_L h_1, \Phi_L h_2, \dots, \Phi_L h_N)]$$

$$U = \Phi_L H$$

If we design H invertible, then

$$\Phi_L = UH^{-1}$$

Hadamard matrix is a good idea. For example if the size of object is 4x4 so that $N = 4$

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

Problem: how can we use a image with negative intensity?

$$X_2 = h_2 \mathbf{1}^T = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 \end{bmatrix}$$

FlatCam Calibration

$$h_2 = [1, -1, 1, -1]^T$$

$$X_2 = h_2 \mathbf{1}^T = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 \end{bmatrix}$$

Make use of superposition!

$$h_2 = h_{2(1)} - h_{2(2)}$$

$$h_{2(1)} = [1,0,1,0]^T$$

$$h_{2(2)} = [0,1,0,1]^T$$

$$X_{2(1)} = h_{2(1)} \mathbf{1}^T = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$X_{2(2)} = h_{2(2)} \mathbf{1}^T = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Then we have

$$X_2 = X_{2(1)} - X_{2(2)}$$

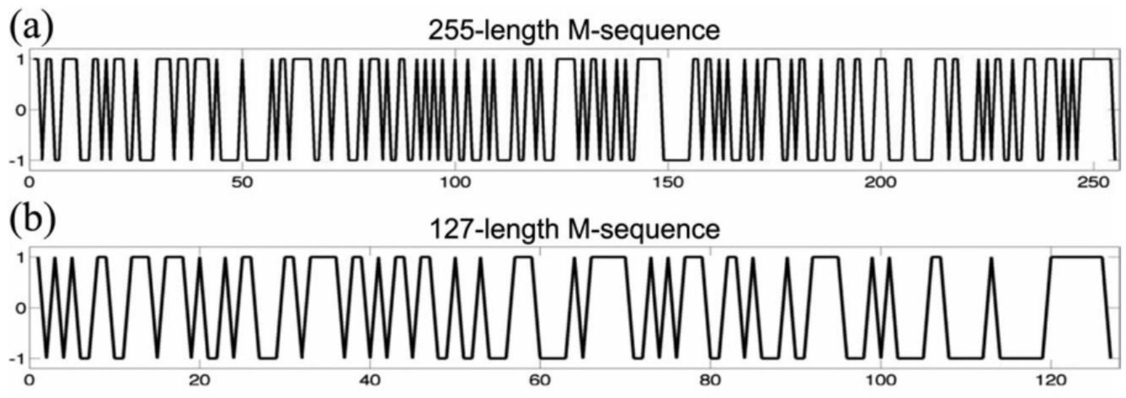
So the problem is solved by:

$$\left.\begin{array}{l} X_{k(1)} \longrightarrow Y_{k(1)} \longrightarrow u_{k(1)} = \Phi_L h_{k(1)} \\ X_{k(2)} \longrightarrow Y_{k(2)} \longrightarrow u_{k(2)} = \Phi_L h_{k(2)} \end{array}\right. \Longrightarrow u_k = u_{k(1)} - u_{k(2)}$$

Hence U and H are both known in the equation

$$\Phi_L = U H^{-1}$$

FlatCam Calibration



We generate mask with the outer product of M-sequence. For example, when $n = 2$

$$m = [1, -1, -1]^T$$

The mask we use will be

$$m \otimes m = \begin{bmatrix} 1 & -1 & -1 \\ -1 & 1 & 1 \\ -1 & 1 & 1 \end{bmatrix}$$

But the entry -1 is optically infeasible!

How about replace -1 with 0 in M-sequence?

$$[1,0,0]^T \otimes [1,0,0]^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

No! the number of transparent feature will be about 25%, not 50% we want

How about replace -1 with 0?

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

The rank is 2, not separable anymore

How to solve it?

FlatCam Calibration

$$\Psi_{\pm 1} = \begin{bmatrix} 1 & -1 & -1 \\ -1 & 1 & 1 \\ -1 & 1 & 1 \end{bmatrix}$$

$$X \xrightarrow{\Psi_{\pm 1}} Y$$ Optically infeasible

$$\Psi_{0/1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

$$X \xrightarrow{\Psi_{0/1}} Y_{0/1}$$ Optically feasible

$$\mathbf{1}\mathbf{1}^T = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$X \xrightarrow{\Psi_{1/1}} Y_{1/1}$$

Consider a single point source X

where $X_{ij} = 1$

$$Y = \Phi_L X \Phi_R^T = \varphi_i {\varphi_j}^T$$

Y is a rank-1 matrix

Since

$$\Psi_{\pm 1} = 2\Psi_{0/1} - \Psi_{11}$$
$$Y = 2Y_{0/1} - Y_{1/1}$$

We want to find a linear operation

$$f(\Psi_{0/1}) = k\Psi_{\pm 1}$$

Then we have

$$f(Y_{0/1}) = Y$$

Such a operation is

subtracting the row and column means of the sensor image