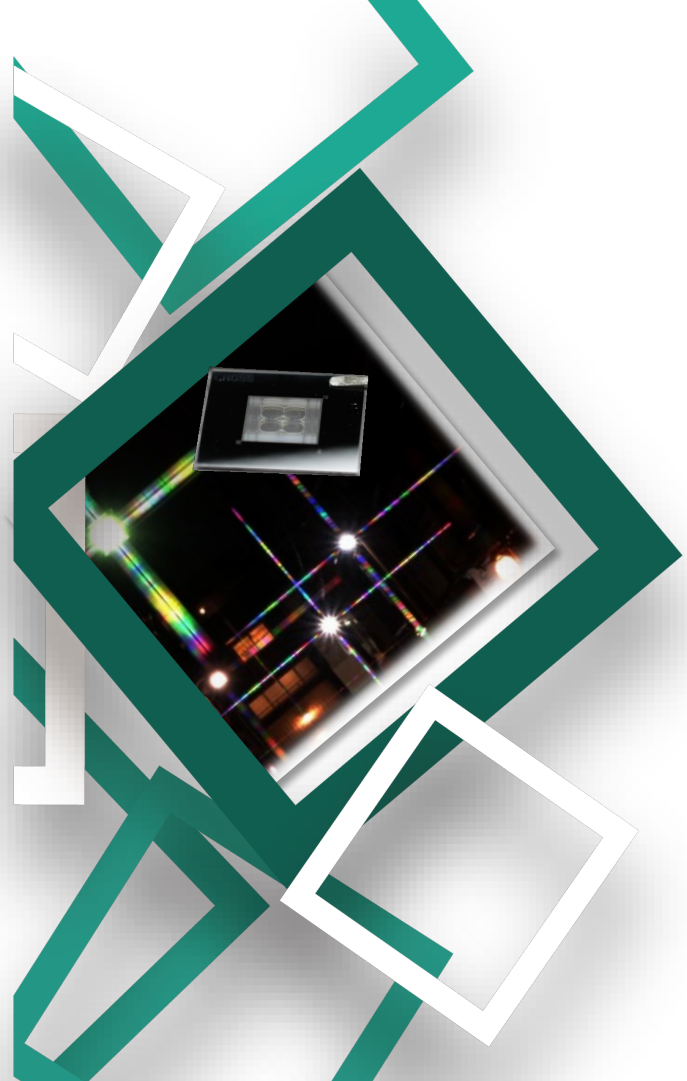
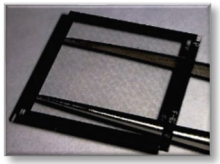
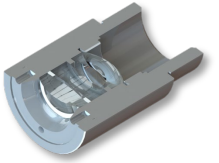
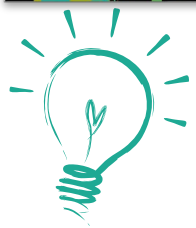








Computational Imaging

Lecture 22 Temporal Encoding V - Time Encoded Photography

Qilin Sun（孙启霖）

香港中文大学（深圳）

点昀技术（Point Spread Technology）

- Dealing with Motion Blur: Coded Exposure
- Optimized Codes for PSF Estimation and Invertibility
- Dealing with Motion Blur: Parabolic Sweep

Dealing with Motion Blur:

Coded Exposure

Motion Blur


Most scene is static
Can moving linearly from left to right
CHAMPAGNE
BRUT
Herbert Beaufort
Bouvy

Motion Blur

=

*

blurry image of moving object

motion blur kernel

sharp image of static object

What does the motion blur kernel depend on?

- Motion velocity determines direction of kernel.
- Shutter speed determines width of kernel.

Can we use deconvolution to remove motion blur?

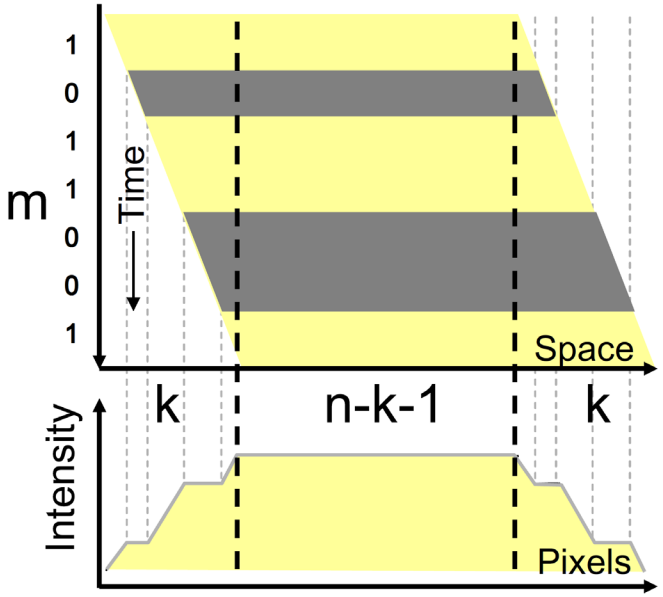
Challenges of Motion Deblurring

- Blur kernel is not invertible.

How would you deal with this?

- Blur kernel is unknown.
- Blur kernel is different for different objects.

The 1-D Motion Blur Process



A

Smear Matrix

X

Unknown Image

=

B

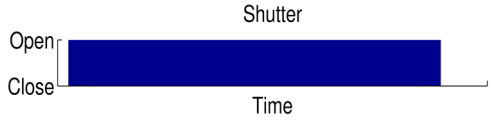
Blurred Image

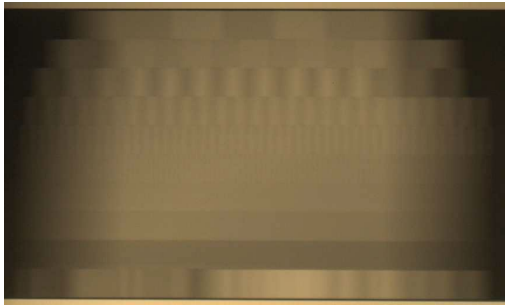
A time-to-space projection for a moving object of size n, with blur of k pixels

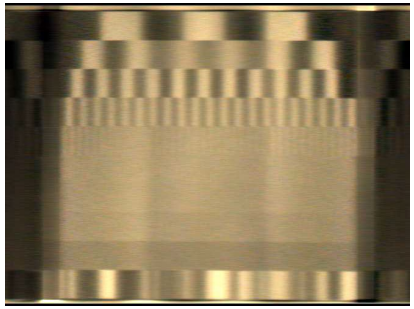
Sample The Motion with More Spatial Frequencies

Code exposure (i.e., shutter speed) to make motion blur kernel better conditioned.

Traditional Exposure







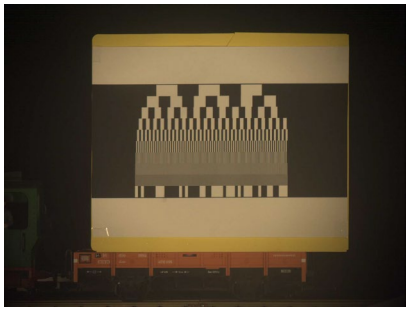
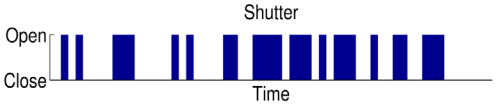
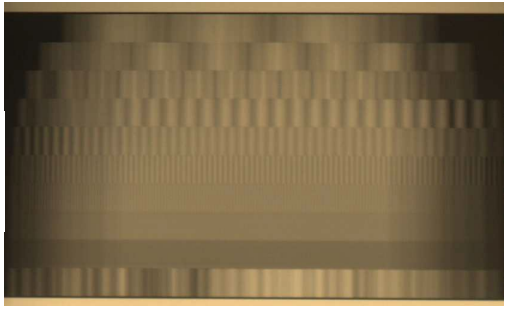


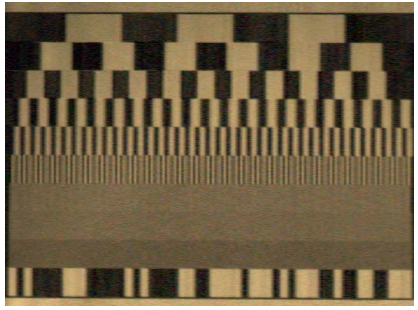
Photo of Test Setup

Coded Exposure





Blurred Photo



Deblurred Image

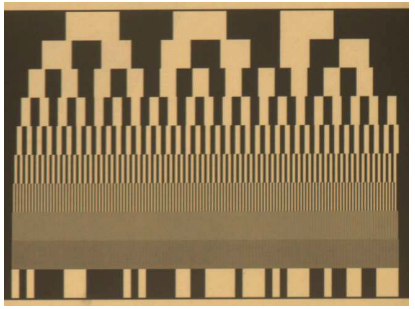


Photo of Test Target

Coded Exposure a.k.a. Flutter Shutter

Code exposure (i.e., shutter speed) to make motion blur kernel better conditioned.

traditional camera

blurry image of moving object

=

motion blur kernel

*

sharp image of static object

flutter-shutter camera

blurry image of moving object

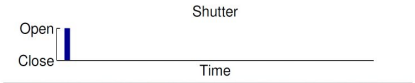
=

motion blur kernel

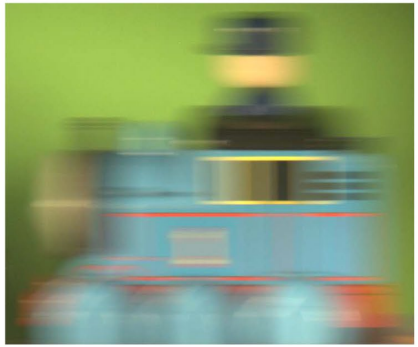
*

sharp image of static object

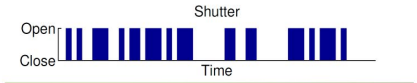
Goal: Select a More Invertible Temporal Code

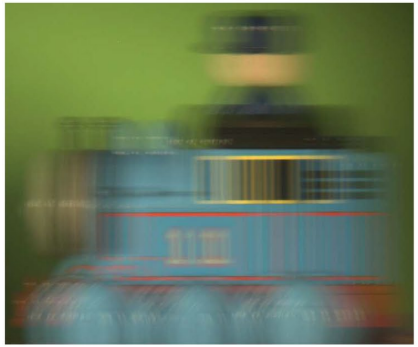


(a) Short Exposure Photo

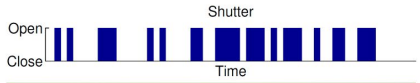


(b) Traditional, 200ms

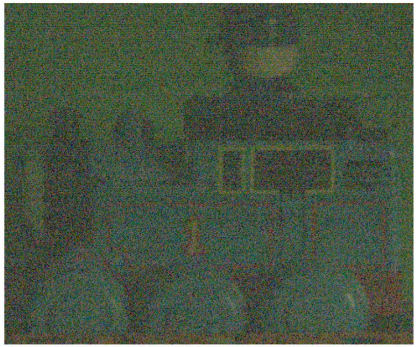




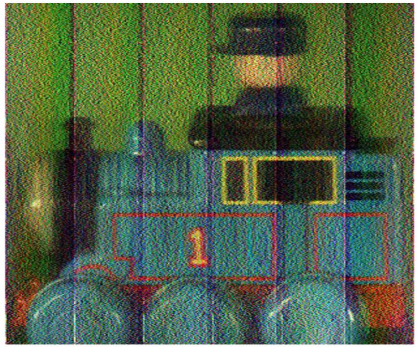
(c) MURA Code, 200ms



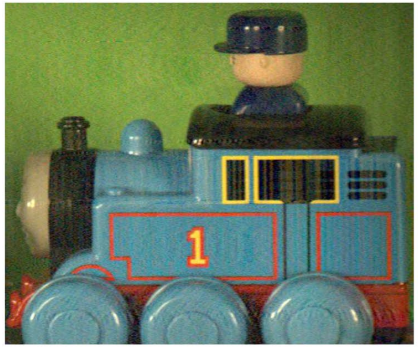
(d) Our Code, 200ms



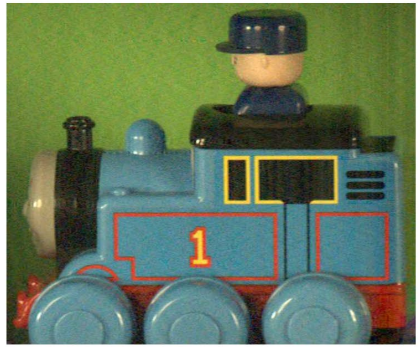
(e) Log intensity of (a)



(f) Deblurred Image



(g) Deblurred Image

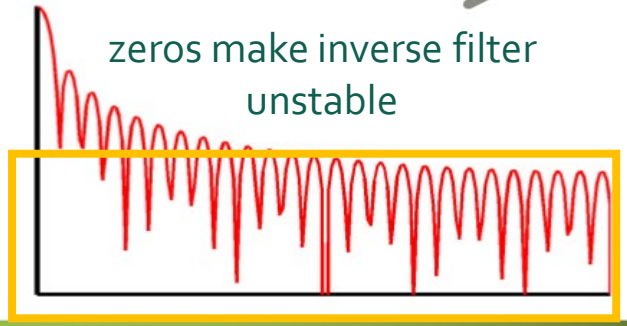


(h) Deblurred Image

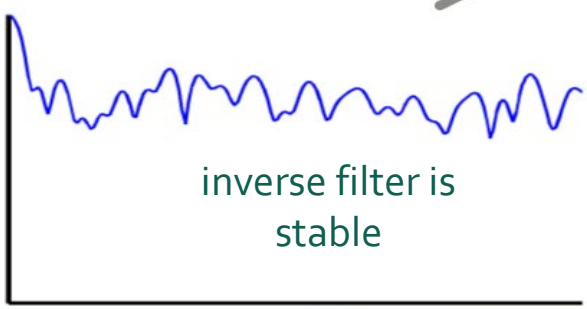
Coded Exposure a.k.a. Flutter Shutter

motion blur kernel in time domain

motion blur kernel in Fourier domain

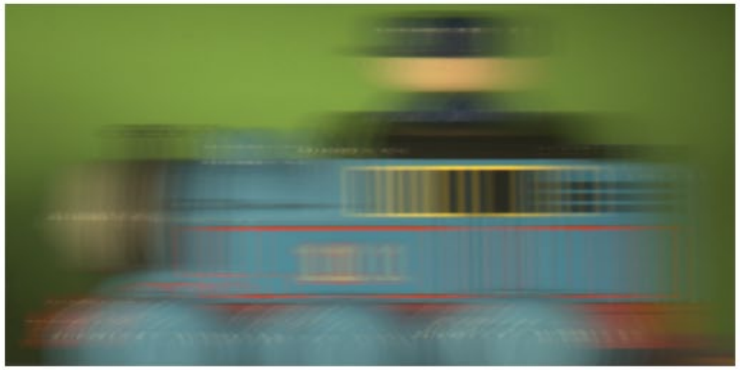


zeros make inverse filter unstable

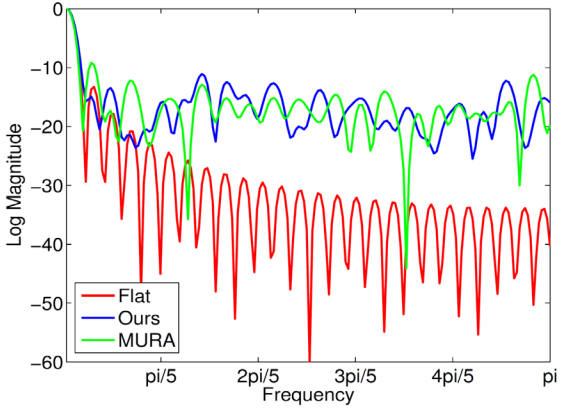


inverse filter is stable

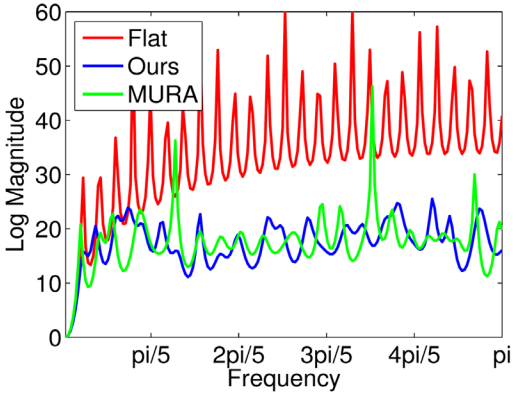
Why is flutter shutter better?



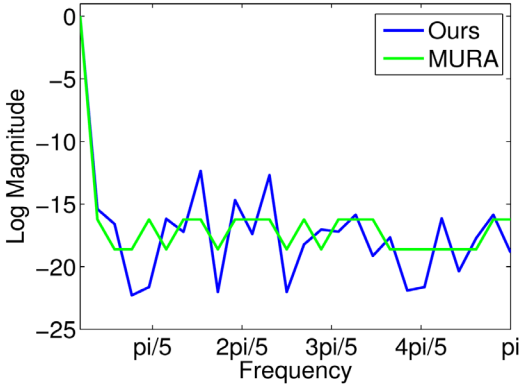
Frequency, Covariance and Matrix Conditionality Analysis



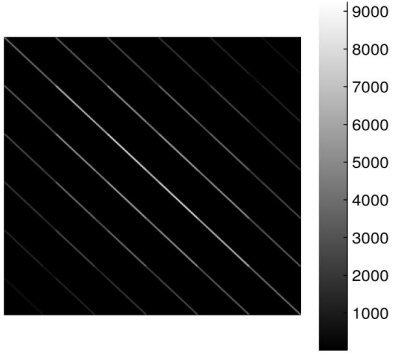
Linear Convolution Filter DFT



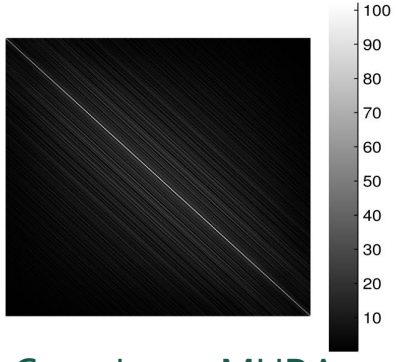
Decoding Filter DFT



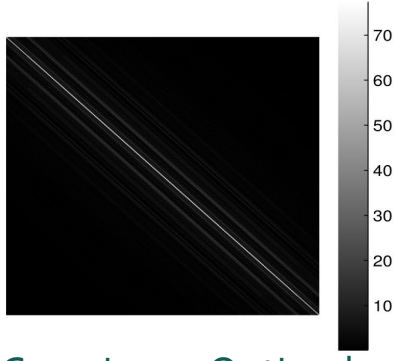
Circular Convolution Filter DFT



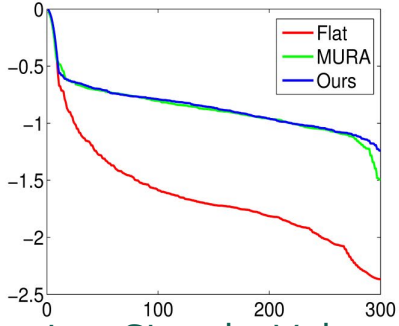
Covariance:Box



Covariance:MURA



Covariance:Optimal



Log Singular Values

The Near Optimal Temporal Code

- Implementing **a randomized linear search** and considered approximately **3×10^6** candidate codes.

1010000111000001010000110011110111010111001001100111

Motion Decoding

- Motion Model

$$AX = B + \eta$$

- Use a least-square estimation to solve for the deblurred image $\hat{X}$

$$\hat{X} = A^{+}B$$

Background Estimation

An opaque object moving in front of a stationary (non-blurred) but non-zero-valued background.

Not sufficient to know the moving object's PSF to deblur the image.

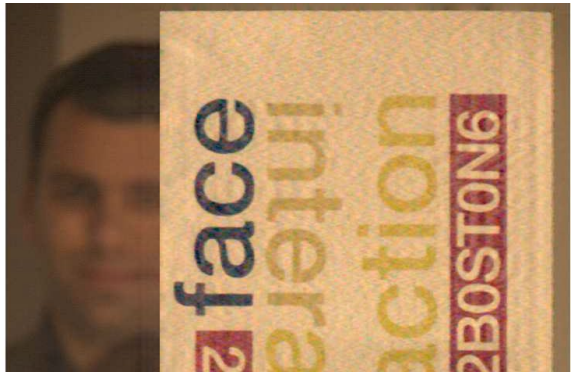
- In the case of a non-zero background, the blurred image is given by

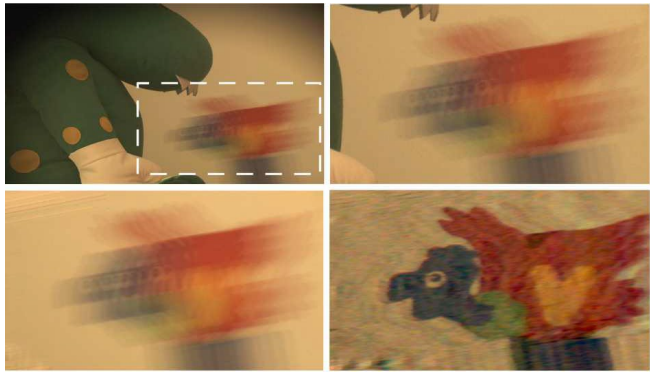
$$B = AX + A_g X_g$$

where X is the moving foreground object, X_g is the static background.
A_g is a diagonal matrix whose elements attenuate the static background.

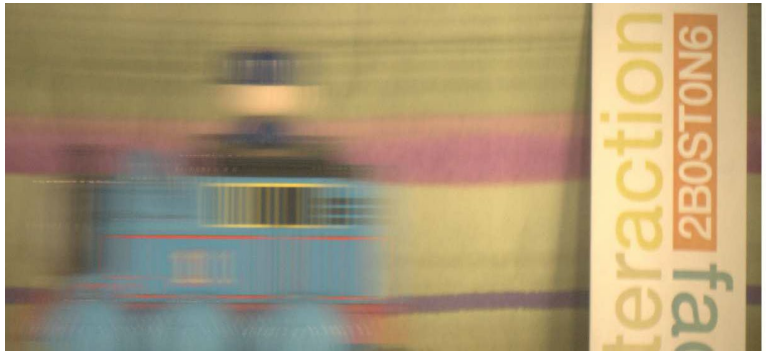
$$A_g = I - \mathtt{diag}(A * I_{(n+k-1)\times 1})$$

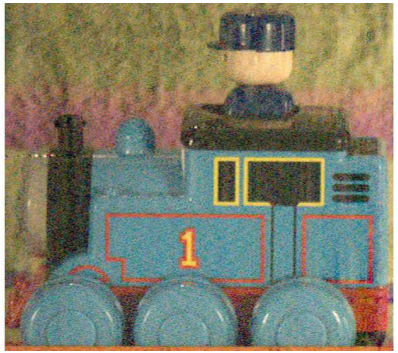
Different Types of Backgrounds





Continuity with the unblurred part of the face





With a striped background aligned with motion lines

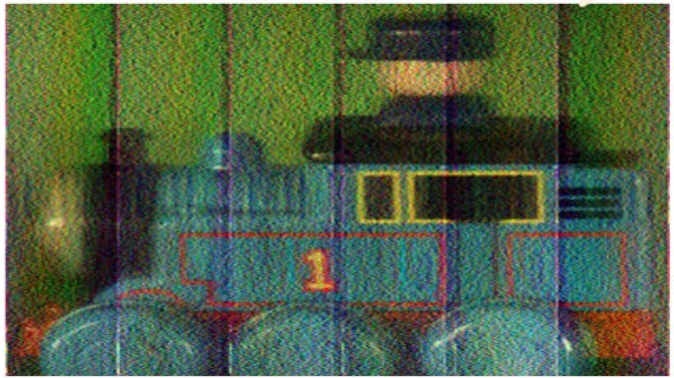
Paper smears across the face

Motion Deblurring Comparison

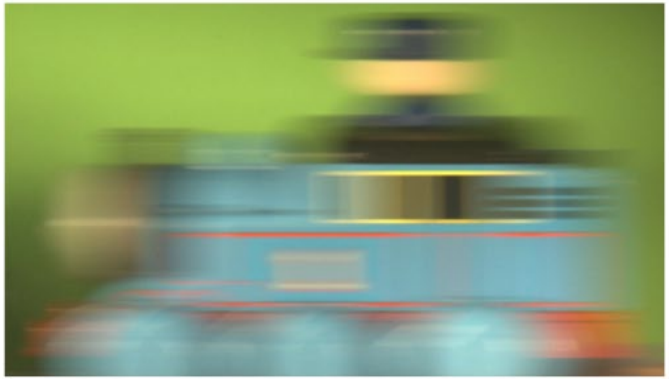
conventional photography

flutter-shutter photography

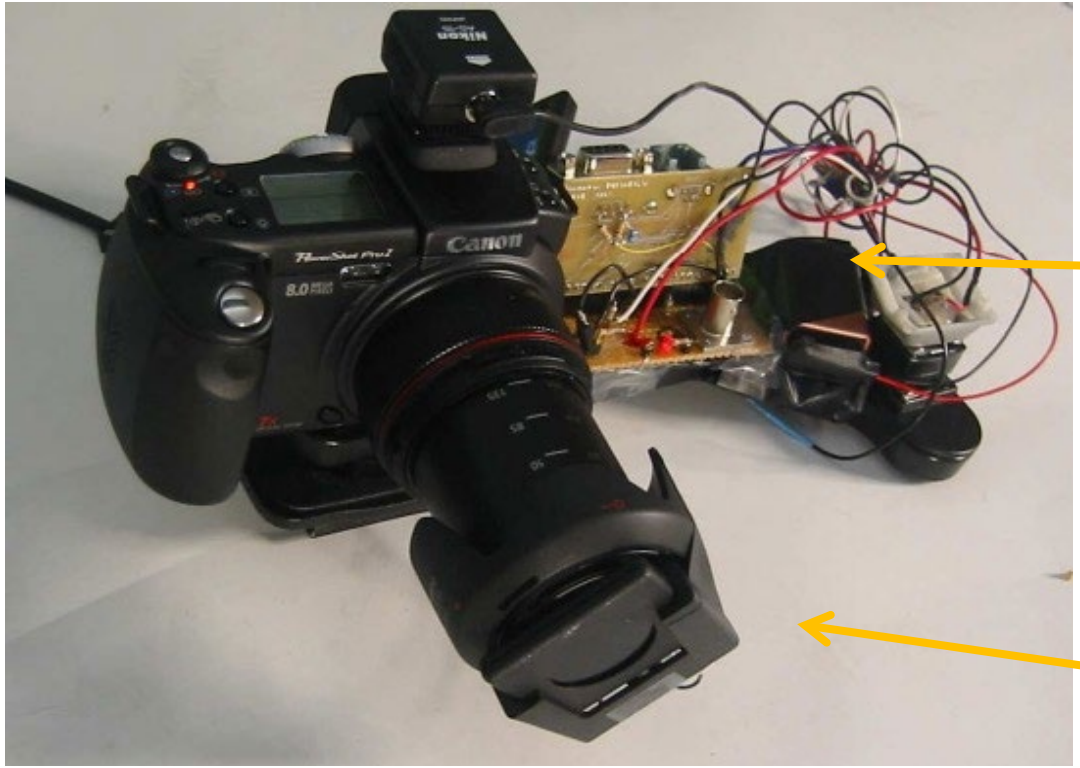
deconvolved output



blurry input



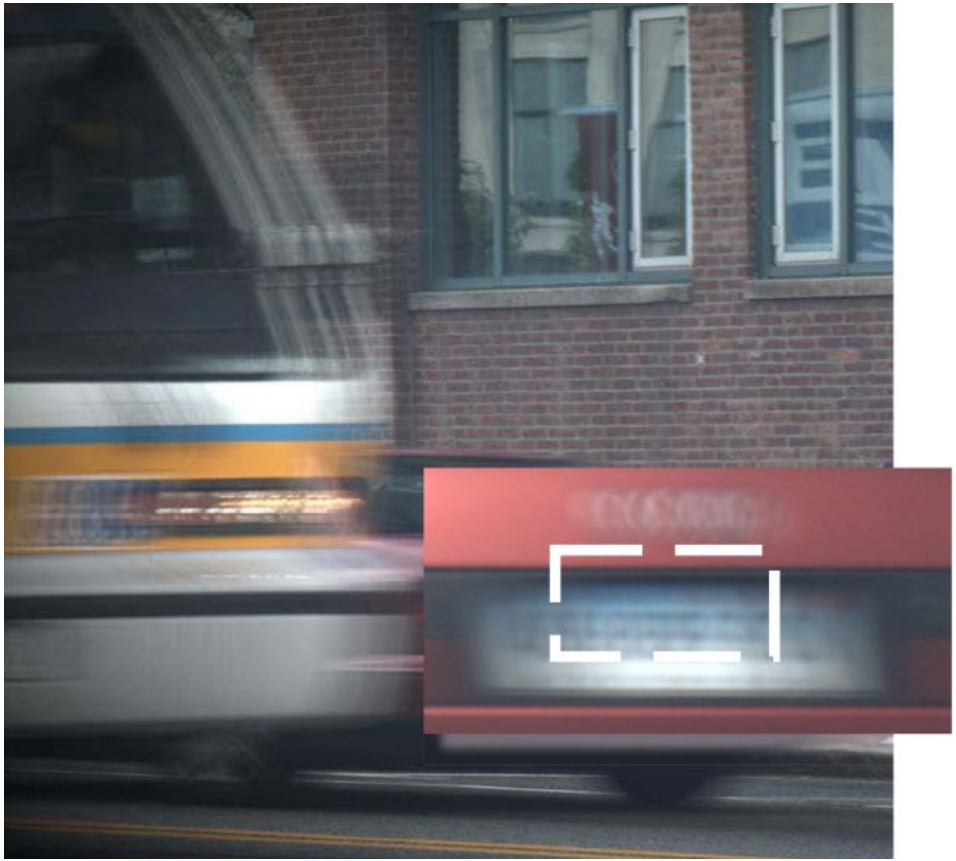
How Would You Implement Coded Exposure?



electronics for external shutter control

very fast external shutter

Results



[Raskar et al. 2006]

License Plate Retrieval

Results

[Raskar et al. 2006]

License Plate Retrieval

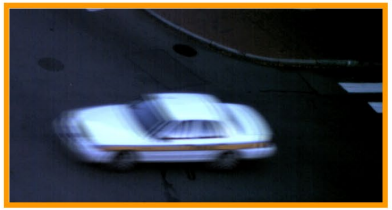
Challenges of Motion Deblurring

- Blur kernel is not invertible.
- Blur kernel is unknown.
- Blur kernel is different for different objects.

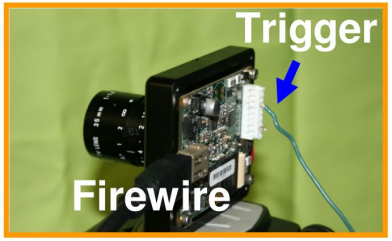
How would you deal with these two?

Coded Exposure Deblurring: Optimized Codes for PSF Estimation and Invertibility

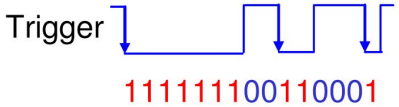
Optimized Codes

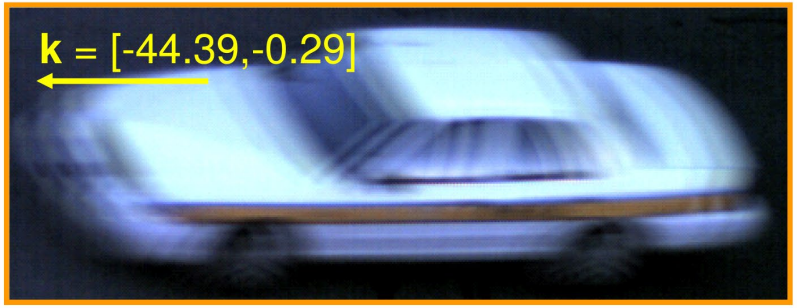


Captured frame



Coded exposure camera





Cropped input frame

Deblurring result

Using a carefully designed code, one can achieve both PSF estimation and invertibility for motion deblurring via coded exposure camera.

Blur Estimation Using Alpha Matting

The captured blurred image I is given by the sum of blurred foreground object and partial background

$$I = s * h + (1 - M * h)b$$

$s(x, y)$ image of the object

$h(x, y)$ the motion PSF

$M(x, y)$ binary indicator function

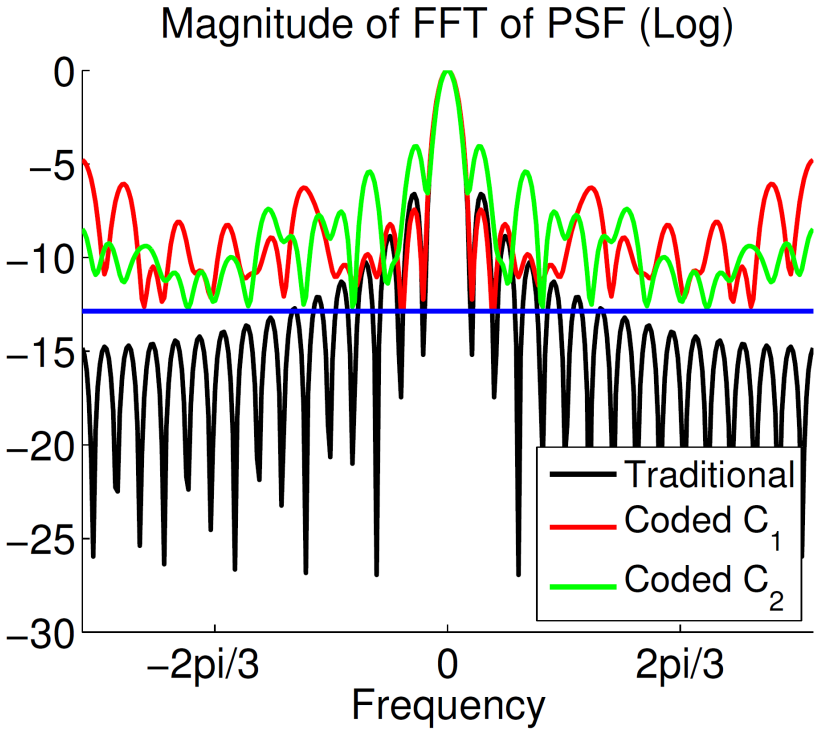
$b(x, y)$ background

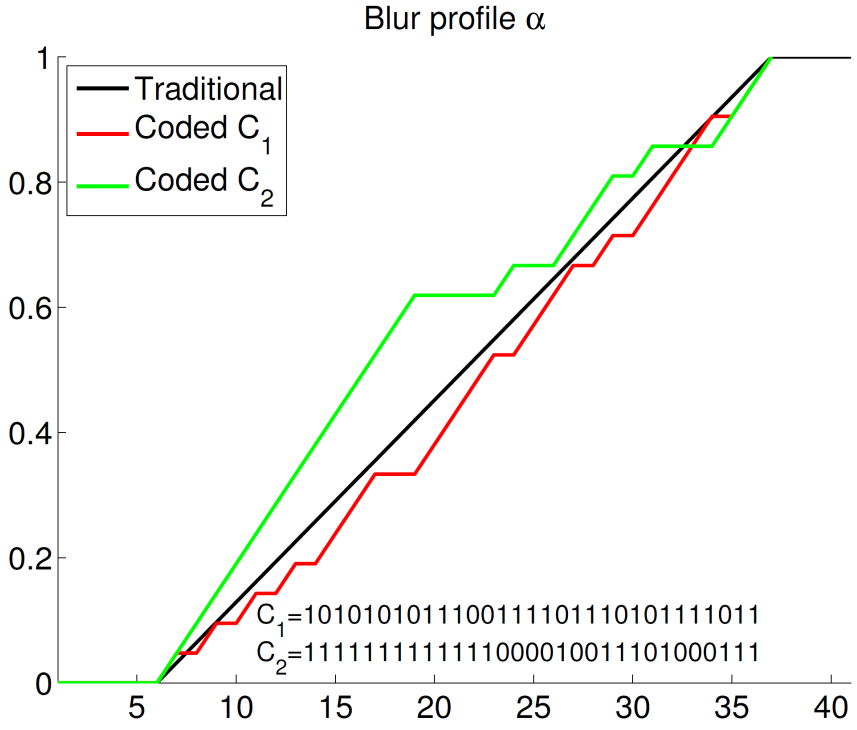
The matting equation:

$$I = F + (1 - \alpha)B$$

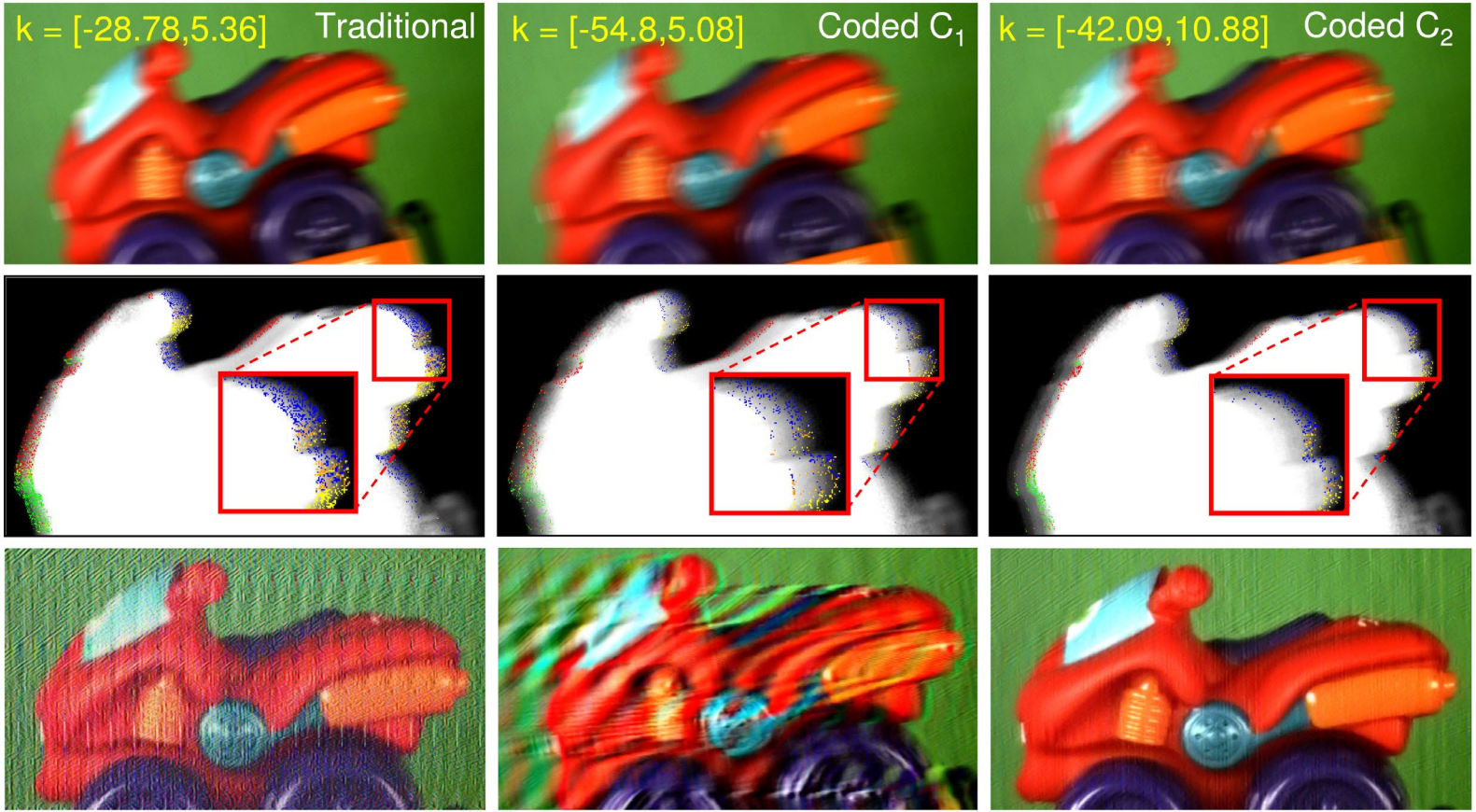
$$B = b, \quad \alpha = M * h, \quad F = (s * h)/(M * h)$$

Two Different Codes

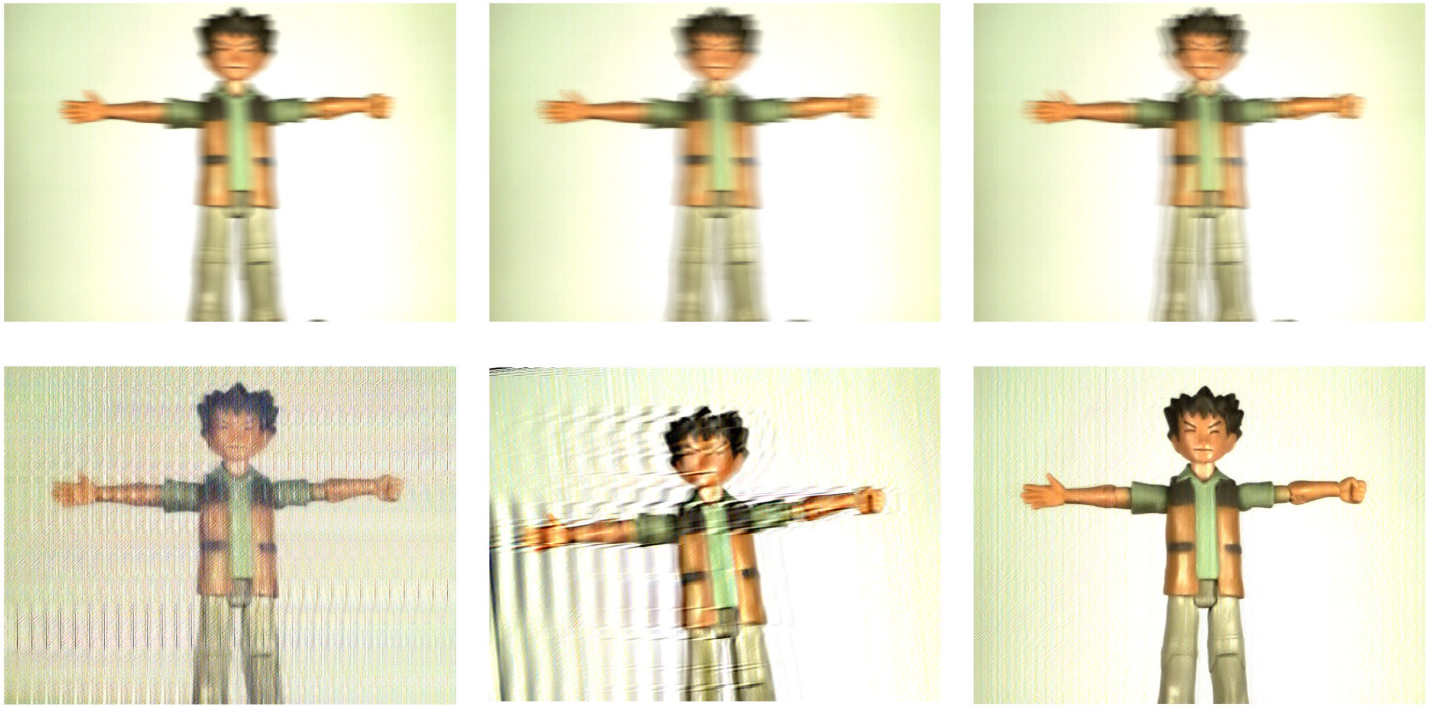




Implementation

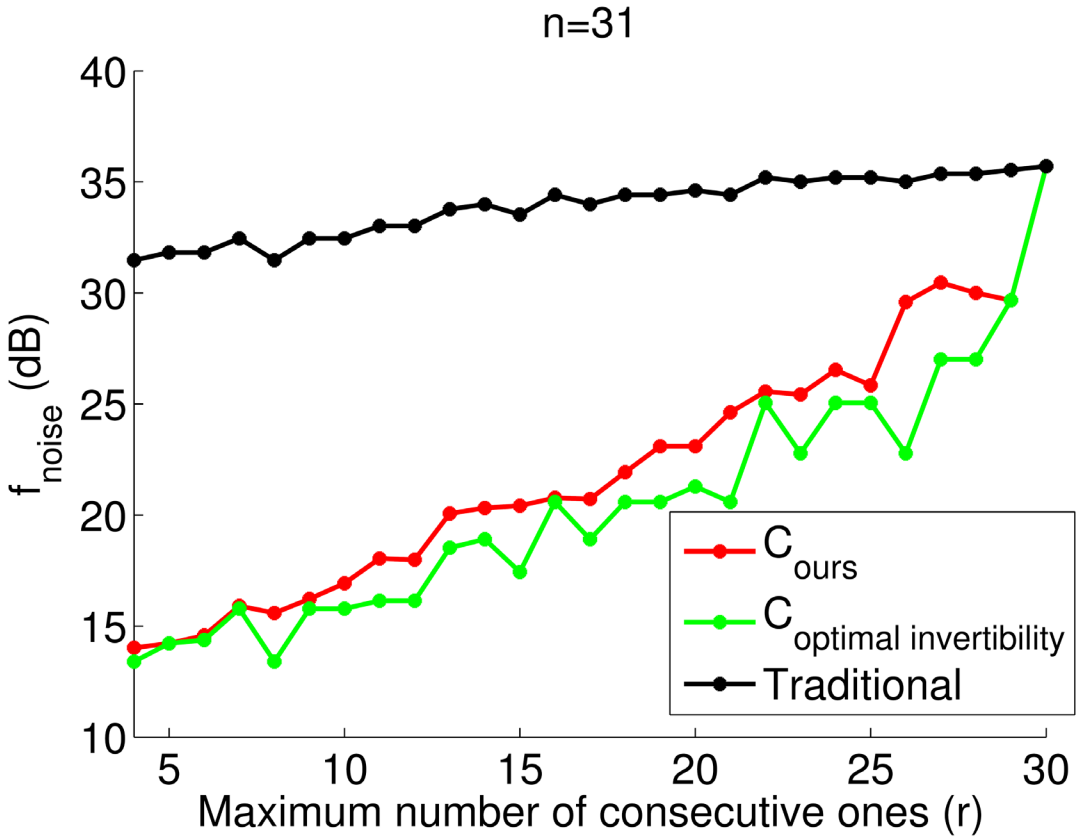


Implementation

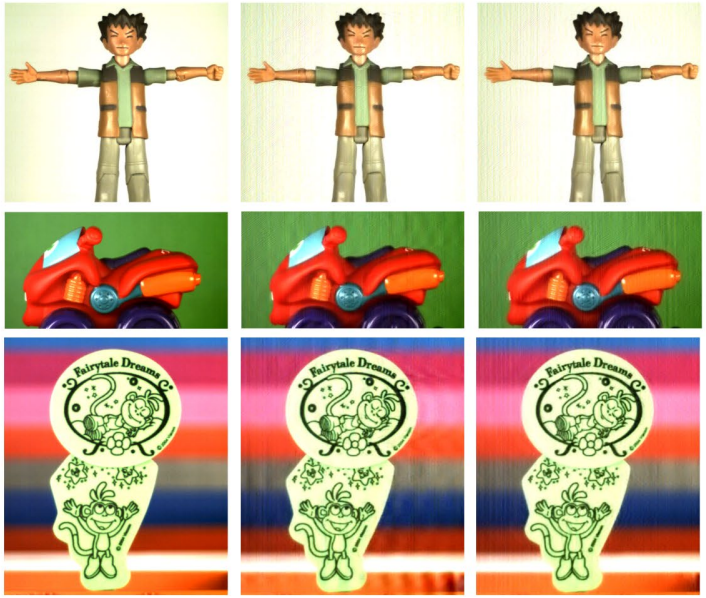


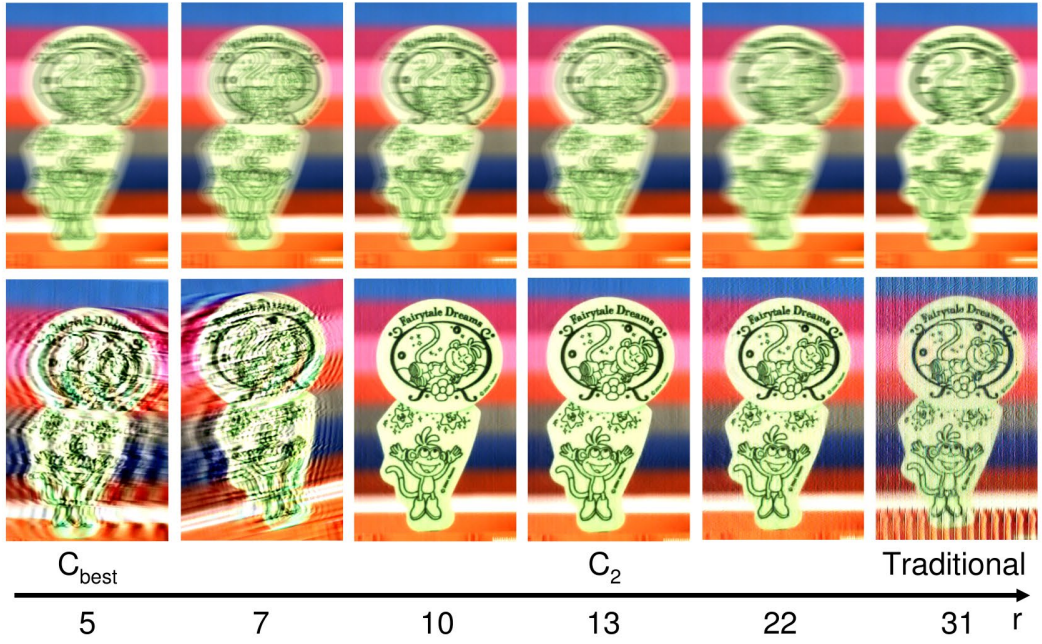
Traditional | Coded C_1 | Coded C_2

Fast Binary Code Search



Best Codes


Ground Truth
Coded C_{best}
Coded C_2


C_{best}
C_2
Traditional
5
7
10
13
22
31
r

Dealing With Motion Blur:

Parabolic Sweep

Motion-invariant Photography

Introduce extra motion so that:

- Everything is blurry; and
- The blur kernel is *motion invariant* (same for all objects).

How would you achieve this?

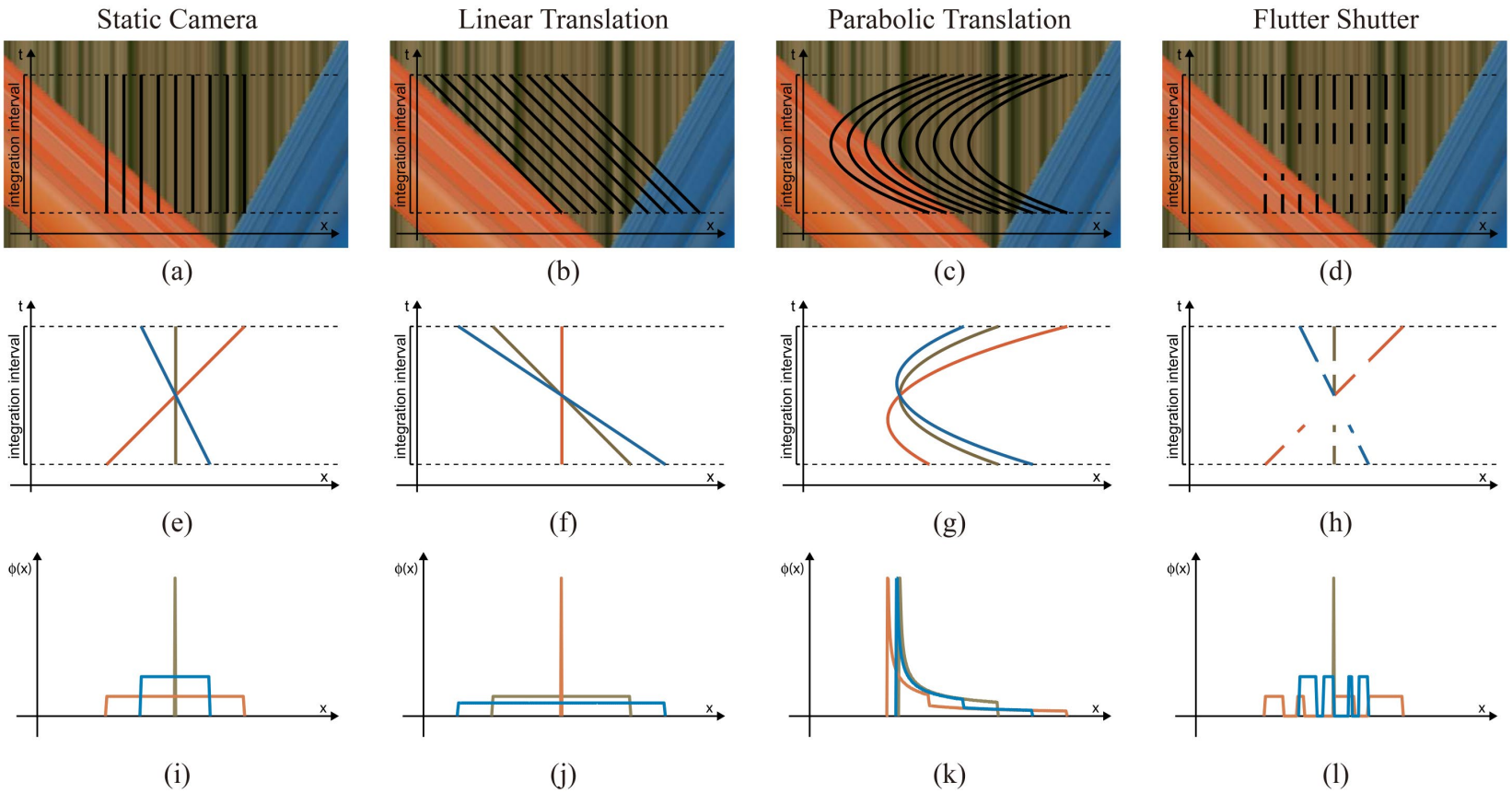
Motion-invariant Photography

Blurred motion captured by a static camera.

Blurred motion captured with a specially designed motion that causes both the static and dynamic regions to blur **identically**.

Deconvoluted Image of the entire image with a single known PSF.

From Integration Curves to PSF



PSF

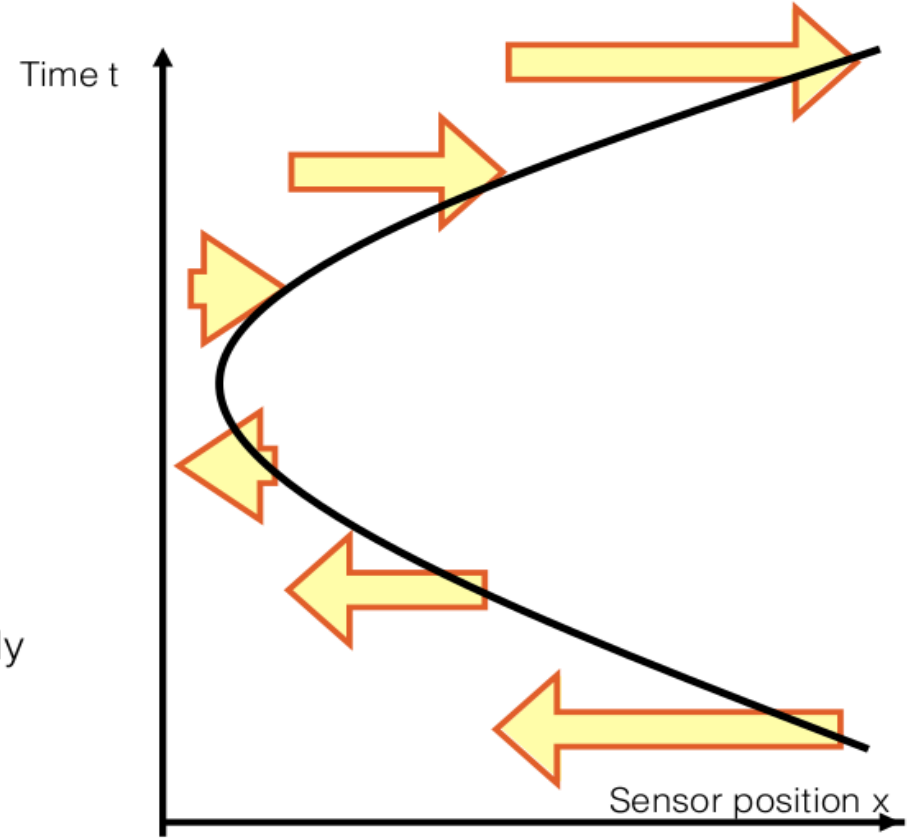
De
e

Parabolic Sweep

Sensor position $x(t)=a\ t^2$

- start by moving very fast to the right
- continuously slow down until stop
- continuously accelerate to the left

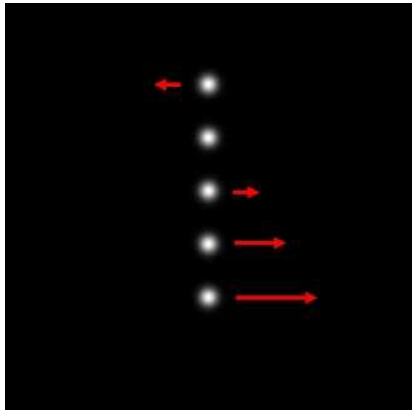
- Intuition:
  - for any velocity, there is one instant where we track perfectly
  - all velocities captured same amount of time



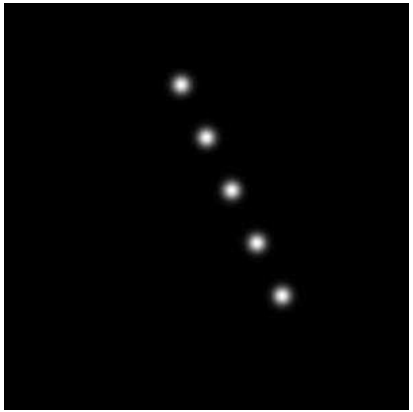
Goal: Create a Velocity Invariant PSF

- We note that a sheared parabola is also a parabola with the same scale, only the center is shifting

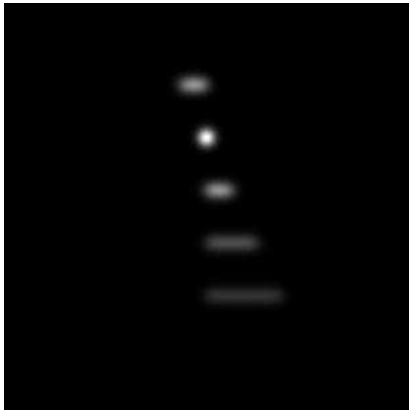
$$f_s(t) = f(t) - st = a_0(t - \frac{s}{2a_0})^2 - \frac{s^2}{4a_0}$$



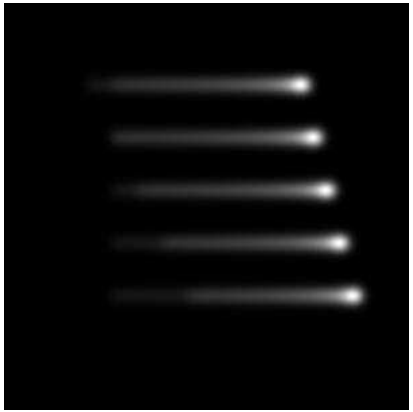
Time 1



Time 2

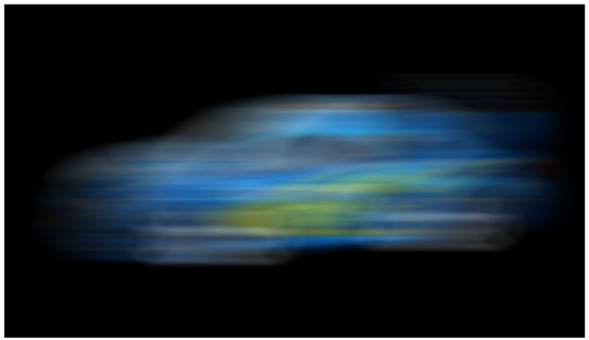


Static camera



Parabolic camera

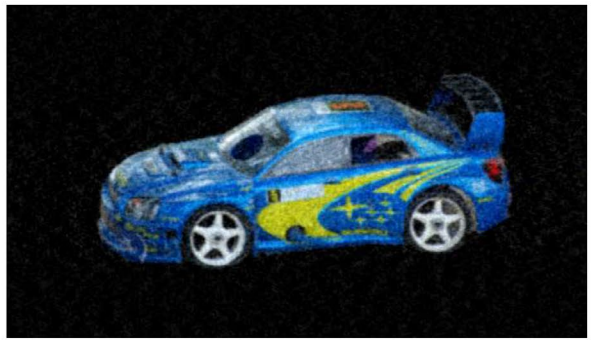
Synthetic visualization of PSF information loss



Static

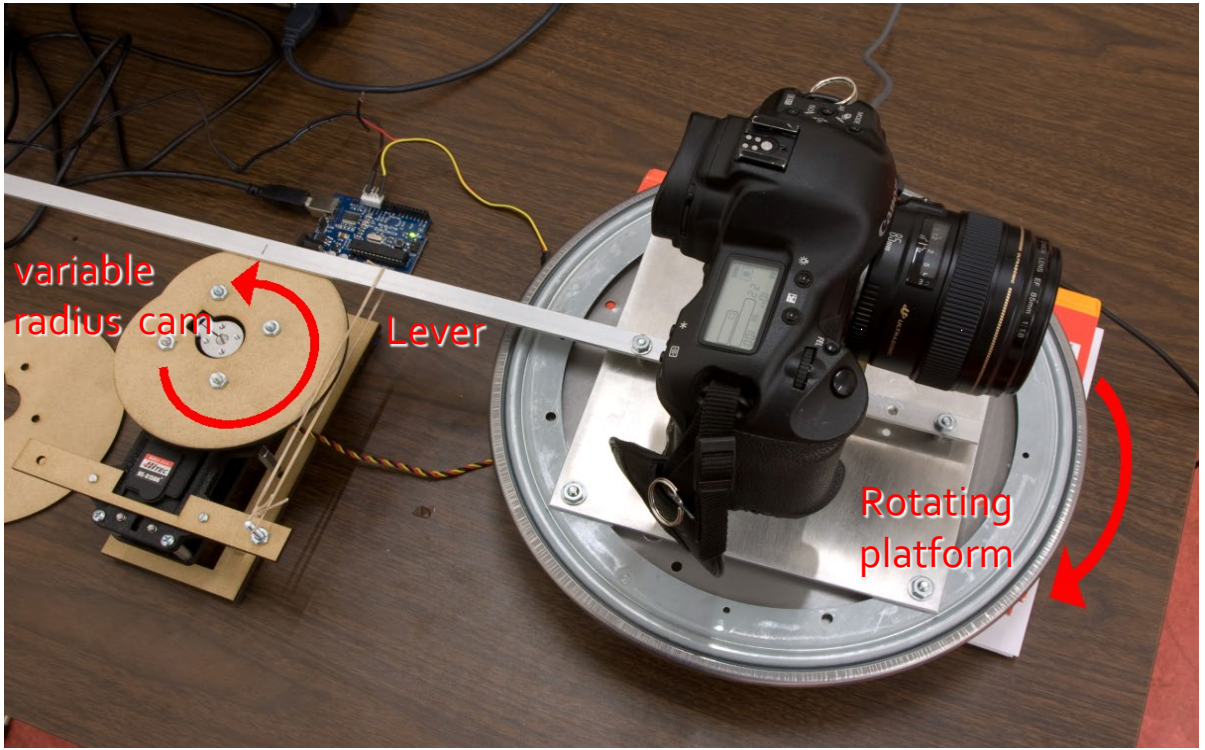


Parabola



Flutter Shutter

Prototype Camera Setup

Approximate small translation by small rotation



Results

static camera input – unknown and variable blur

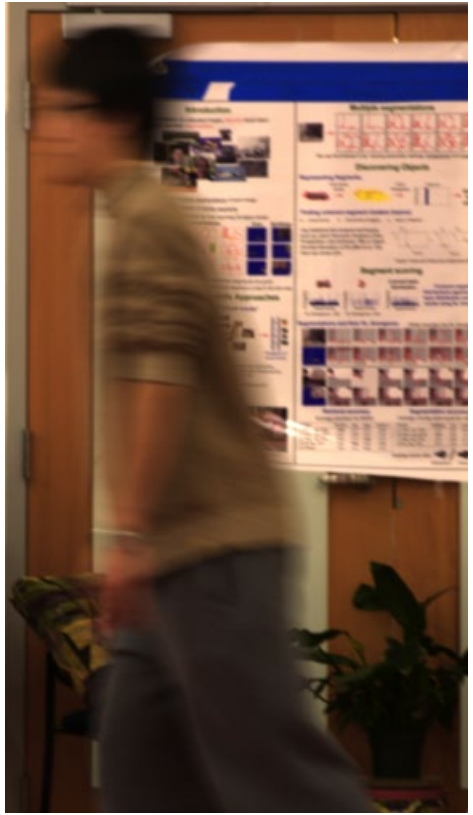
parabolic input – blur is invariant to velocity

Results

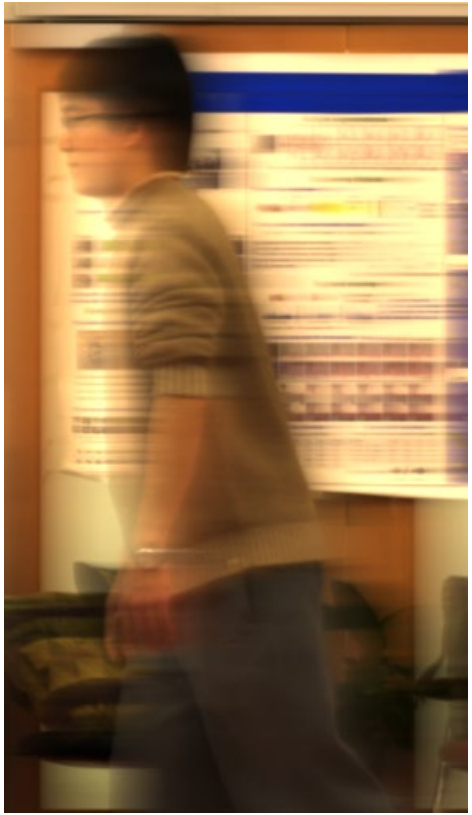
static camera input – unknown and variable blur

parabolic input –
blur is invariant to velocity

Results



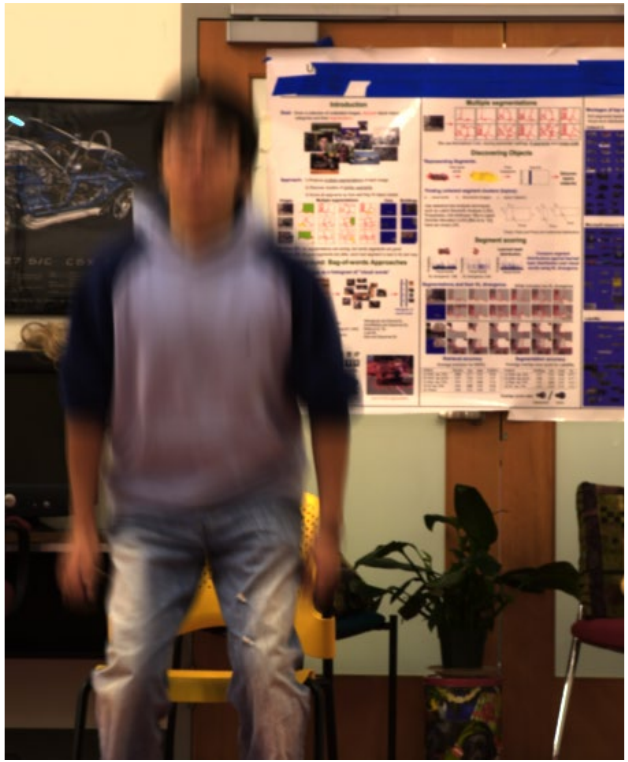
static camera input



parabolic camera input

deconvolution output

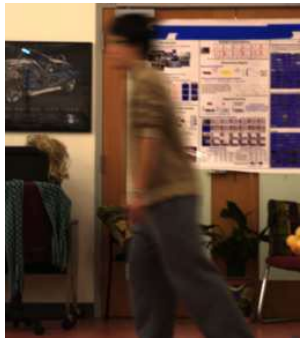
Results



static camera input

output after deconvolution
Why does it fail in this case?

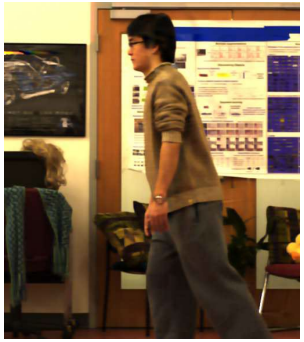
Experiments

Static camera input

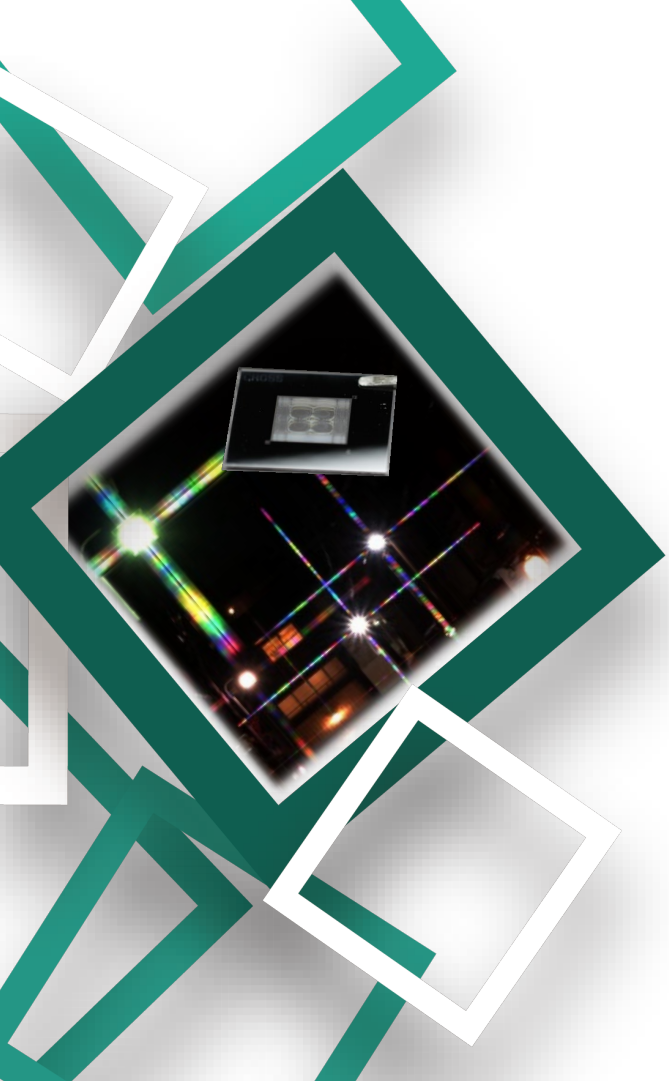

Debluring results by [Levin 2006]


Box filter fitted manually to moving layer and applied to deblur the entire image

Spatially uniform deconvolution of image from parabolic integration

- Dealing with Motion Blur: Coded Exposure
- Optimized Codes for PSF Estimation and Invertibility
- Dealing with Motion Blur: Parabolic Sweep



Thank You!

Qilin Sun（孙启霖）

香港中文大学（深圳）

点昀技术（Point Spread Technology）