Coded photography

http://graphics.cs.cmu.edu/courses/15-463
Overview of today’s lecture

• The coded photography paradigm.
• Dealing with depth blur: coded aperture.
• Dealing with depth blur: focal sweep.
• Dealing with depth blur: generalized optics.
• Dealing with motion blur: coded exposure.
• Dealing with motion blur: parabolic sweep.
Most of these slides were adapted from:

- Fredo Durand (MIT).
- Anat Levin (Technion).
- Gordon Wetzstein (Stanford).
The coded photography paradigm
Conventional photography

- Optics capture something that is (close to) the final image.
- Computation mostly “enhances” captured image (e.g., deblur).
Coded photography

- Generalized optics encode world into intermediate representation.
- Generalized computation decodes representation into multiple images.

Can you think of any examples?
Early example: mosaicing

- Color filter array encodes color into a mosaic.
- Demosaicing decodes color into RGB image.
Recent example: plenoptic camera

- Plenoptic camera encodes world into lightfield.
- Lightfield rendering decodes lightfield into refocused or multi-viewpoint images.
Why are our images blurry?

- Lens imperfections.
- Camera shake.
- Scene motion.
- Depth defocus.

- last lecture: deconvolution
- last lecture: blind deconvolution
- flutter shutter, motion-invariant photo
- coded aperture, focal sweep, lattice lens

conventional photography
coded photography
Why are our images blurry?

- Lens imperfections.
- Camera shake.
- Scene motion.
- Depth defocus.

Last lecture: deconvolution
Last lecture: blind deconvolution
Flutter shutter, motion-invariant photo
Coded aperture, focal sweep, lattice lens

Conventional photography
Coded photography
Dealing with depth blur: coded aperture
Defocus blur

Point spread function (PSF): The blur kernel of a (perfect) lens at some out-of-focus depth.

What does the blur kernel depend on?
Defocus blur

Point spread function (PSF): The blur kernel of a (perfect) lens at some out-of-focus depth.

- Aperture determines shape of kernel.
- Depth determines scale of blur kernel.
Depth determines scale of blur kernel

object distance $D$  focus distance $D'$

PSF
Depth determines scale of blur kernel

object distance $D$  

focus distance $D'$  

PSF
Depth determines scale of blur kernel

PSF

object distance $D$

focus distance $D'$
Depth determines scale of blur kernel.
Depth determines scale of blur kernel
Aperture determines shape of blur kernel

object distance $D$

focus distance $D'$

PSF
Aperture determines shape of blur kernel

- Photo of aperture
- Shape of aperture (optical transfer function, OTF)
- Blur kernel (point spread function, PSF)

What causes these lines?

How do the OTF and PSF relate to each other?
Removing depth defocus

measured PSFs at different depths

input defocused image

How would you create an all in-focus image given the above?
Removing depth defocus

Defocus is local convolution with a depth-dependent kernel.

How would you create an all in-focus image given the above?
Removing depth defocus

Defocus is *local* convolution with a depth-dependent kernel.

\[
\begin{align*}
\text{depth 3} & \quad = \quad \ast \\
\text{depth 2} & \quad = \quad \ast \\
\text{depth 1} & \quad = \quad \ast 
\end{align*}
\]

How would you create an all in-focus image given the above?
Removing depth defocus

- Deconvolve each image patch with all kernels
- Select the right scale by evaluating the deconvolution results

How do we select the correct scale?
Removing depth defocus

Problem: With standard aperture, results at different scales look very similar.
Coded aperture

Solution: Change aperture so that it is easier to pick the correct scale

Wrong scale

Correct scale

Wrong scale
Build your own coded aperture
Voila!
Coded aperture changes shape of kernel

object distance $D$

focus distance $D'$

PSF
Coded aperture changes shape of kernel

object distance $D$

focus distance $D'$

PSF
Coded aperture changes shape of PSF
Coded aperture changes shape of PSF

New PSF preserves high frequencies
• More content available to help us determine correct depth

conventional

fft

coded
All-focused
(deconvolved)
Comparison between standard and coded aperture

Ringing due to wrong scale estimation
Comparison between standard and coded aperture
Refocusing
Refocusing
Refocusing
Depth estimation
All-focused (deconvolved)
Refocusing
Refocusing
Refocusing
Depth estimation
Any problems with using a coded aperture?
Any problems with using a coded aperture?

- We lose a lot of light due to blocking.
- The deconvolution becomes harder due to more diffraction/zero-s in frequency domain.
- We still need to select correct scale.
Dealing with depth blur: focal sweep
The difficulty of dealing with depth defocus

At every focus setting, objects at different depths are blurred by different PSF.
The difficulty of dealing with depth defocus varying in focus distance

At every focus setting, objects at different depths are blurred by different PSF.

PSFs for object at depth 1
The difficulty of dealing with depth deblur

At every focus setting, objects at different depths are blurred by different PSF.
The difficulty of dealing with depth defocus

varying in-focus distance

At every focus setting, objects at different depths are blurred by different PSF

As we sweep through focus settings, each point every object is blurred by all possible PSFs
Focal sweep

varying in-focus distance

linear motion: exposure

distance sensor-lens
time

Go through all focus settings during a single exposure

What is the effective PSF in this case?

PSFs for object at depth 1
PSFs for object at depth 2
Focal sweep

varying in-focus distance

\[ \int dt = \text{effective PSF for object at depth 1} \]
\[ \int dt = \text{effective PSF for object at depth 2} \]

Go through all focus settings during a single exposure

Anything special about these effective PSFs?
Focal sweep

The effective PSF is:
1. Depth-invariant – all points are blurred the same way regardless of depth.
2. Never sharp – all points will be blurry regardless of depth.

What are the implications of this?
1. The image we capture will be sharp nowhere; but
2. We can use simple (global) deconvolution to sharpen parts we want

1. Can we estimate depth from this?
2. Can we do refocusing from this?
Focal sweep

The effective PSF is:
1. Depth-invariant – all points are blurred the same way regardless of depth.
2. Never sharp – all points will be blurry regardless of depth.

What are the implications of this?
Focal sweep

The effective PSF is:
1. Depth-invariant – all points are blurred the same way regardless of depth.
2. Never sharp – all points will be blurry regardless of depth.

What are the implications of this?
1. The image we capture will not be sharp anywhere; but
2. We can use simple (global) deconvolution to sharpen parts we want

1. Can we estimate depth from this?
2. Can we do refocusing from this?
Focal sweep

The effective PSF is:
1. Depth-invariant – all points are blurred the same way regardless of depth.
2. Never sharp – all points will be blurry regardless of depth.

What are the implications of this?
1. The image we capture will not be sharp anywhere; but
2. We can use simple (global) deconvolution to sharpen parts we want

1. Can we estimate depth from this?
2. Can we do refocusing from this?

Depth-invariance of the PSF means that we have lost all depth information
How can you implement focal sweep?
How can you implement focal sweep?

Use translation stage to move sensor relative to fixed lens during exposure

Rotate focusing ring to move lens relative to fixed sensor during exposure
Comparison of different PSFs

Normal Camera PSF

EDOF Camera IPSF

Scene Depth

Image Location (x,y)
Depth of field comparisons

captured focal sweep always blurry!

conventional photo (small DOF)

EDOF image

conventional photo (large DOF, noisy)
Any problems with using focal sweep?
Any problems with using focal sweep?

- We have moving parts (vibrations, motion blur).
- Perfect depth invariance requires very constant speed.
- We lose depth information.
Dealing with depth blur: generalized optics
Change optics, not aperture

object distance $D$

focus distance $D'$

PSF
Wavefront coding

Replace lens with a cubic phase plate
Wavefront coding

- Rays no longer converge.
- Approximately depth-invariant PSF for certain range of depths.
Lattice lens

Add lenslet array with varying focal length in front of lens
Lattice lens

Does this remind you of something?
Lattice lens

- Effectively captures only the “useful” subset of the 4D lightfield.

- PSF is not depth-invariant, so local deconvolution as in coded aperture.

**Light field spectrum:** 4D

**Image spectrum:** 2D

**Depth:** 1D

→ Dimensionality gap (Ng 05)

Only the 3D manifold corresponding to physical focusing distance is useful

PSFs at different depths
Results

Standard lens
Results

Lattice lens
Results

Standard lens
Results

Lattice lens
Results

Standard lens
Results

Lattice lens
Refocusing example
Refocusing example
Comparison of different techniques

Depth of field comparison:

- **standard lens** < **coded aperture** << **focal sweep** < **wavefront coding** < **lattice lens**

Object at in-focus depth

Object at extreme depth
Diffusion coded photography

- can also do EDOF with diffuser as coded aperture, has better inversion characteristics than lattice focal lens

Can you think of any issues?
Dealing with motion blur
Why are our images blurry?

- Lens imperfections.
- Camera shake.
- Scene motion.
- Depth defocus.

- last lecture: deconvolution
- last lecture: blind deconvolution
- flutter shutter, motion-invariant photo
- coded aperture, focal sweep, lattice lens

conventional photography

coded photography
Motion blur

Most scene is static

Can moving linearly from left to right
What does the motion blur kernel depend on?
Motion blur

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.

• Blur kernel is unknown.

• Blur kernel is different for different objects.
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 this?
Dealing with motion blur: coded exposure
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
How would you implement coded exposure?
How would you implement coded exposure?

very fast external shutter electronics for external shutter control

very fast external shutter
Coded exposure a.k.a. flutter shutter

motion blur kernel in time domain

motion blur kernel in Fourier domain

Why is flutter shutter better?
Coded exposure a.k.a. flutter shutter

motion blur kernel in time domain

motion blur kernel in Fourier domain

zeros make inverse filter unstable

Why is flutter shutter better?

inverse filter is stable
Motion deblurring comparison

conventional photography

flutter-shutter photography

deconvolved output

blurry input
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?
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?
Sensor position $x(t) = at^2$

- 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
Hardware implementation

Approximate small translation by small rotation
Some results

static camera input - unknown and variable blur

parabolic input - blur is invariant to velocity
Some results

- static camera input - unknown and variable blur
- output after deconvolution

Is this blind or non-blind deconvolution?
Some results

- static camera input
- parabolic camera input
- deconvolution output
Some results

static camera input

output after deconvolution

Why does it fail in this case?
References

Basic reading:
  The two papers introducing coded aperture for depth and refocusing, the first covers deblurring in more detail, whereas the second deals with optimal mask selection and includes very interesting lightfield analysis.
  The focal sweep paper.
  The wavefront coding paper.
  The lattice focal lens paper, which also includes a discussion of wavefront coding.
  The diffusion coded photography paper.
  The flutter shutter paper.
  The motion-invariant photography paper.

Additional reading:
  This paper has a nice discussion of wavefront coding, in addition to analysis of lightfields and their relationship to wave optics concepts.
  This paper introduces the use of coded apertures for hyperspectral imaging, instead of depth and refocusing.