Image Based Lighting

CS273 Final Project

Zhiyu Xiao
Haowen Liu
Eammon Littler

Motivation

Investigate the application of HDR in image-based lighting (IBL), which is a rendering technique in computer graphics that allows us to insert synthetic 3D models into photographs in a realistic way.

Workflow

HDR Image (After Merge)

1/20 1/10 1/5 1/2

HDR Environment Map

Physically Based Rendering

Photorealistic Image

HDR Image (After Merge)

Tone-mapped Background

Recovering HDR Image

Recovering HDR Image

Image Capture

Camera Response Curve Calibration

Image Merging

Image Capture

Scenes of different Exposures

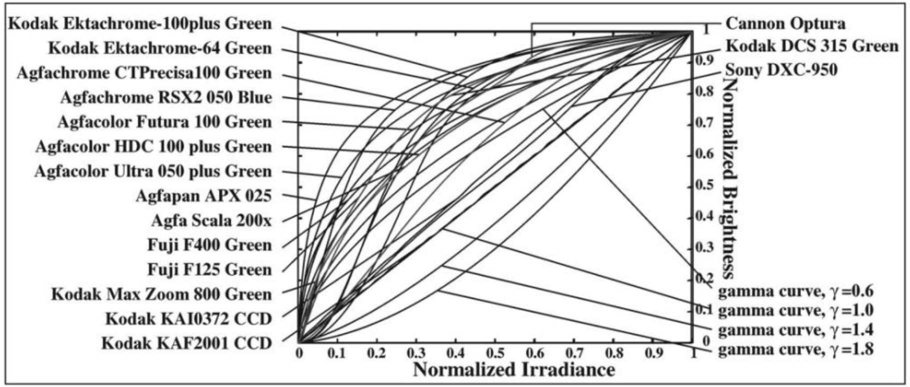
Different shutter speeds

1/80 1/40 1/20 1/10 1/5 1/2 0.77 1.6

Camera Response Curve Calibration

Use camera response function (CRF) to recover actual dynamic range

$$Z = f(E\Delta t)$$



CRFs of different cameras

Camera Response Curve Calibration

CRF can be calculated by solving an optimization process that minimizing the least square error

$$\ln f^{-1}(Z) \;= g(Z) = lnE + ln\Delta t$$

$$\min\left(\sum_i^N \sum_j^P \{w(Z_{ij})[g(Z_{ij}) - lnE_i - ln\Delta t_j]\}^2\right)_{g,E} + \lambda \sum_{z=Z_{\min}}^{Z_{\max}} [w(z)g''(z)]^2$$

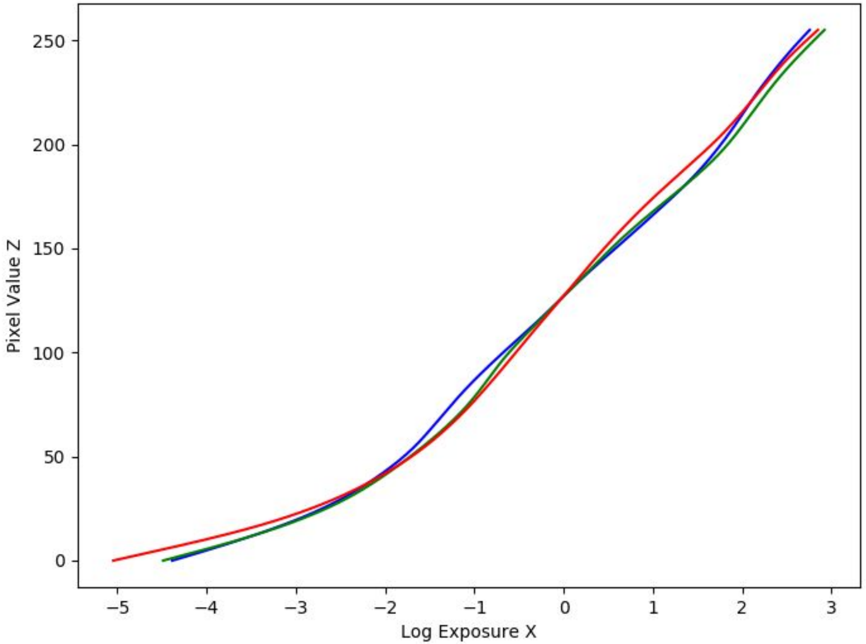
Camera Response Curve Calibration

Subsample pixels to be used in calculation to avoid numerical instabilities and to speed up computation

Regular subsampling - can miss important samples

Sampling from CDF histograms - more robust to moderate movements

Camera Response Curve Calibration



CRF recovered

Image Merging

Recover scene irradiance *E* using CRF for each exposure and taking average

$$\ln(E_i) = \frac{1}{W_i} \sum_{j=1}^{P} w(Z_{ij}) ln E_{ij}$$

$$W_i = \sum_{j=1}^{P} w(Z_{ij})$$

Weighting function to give higher weight to middle pixels which are more reliable

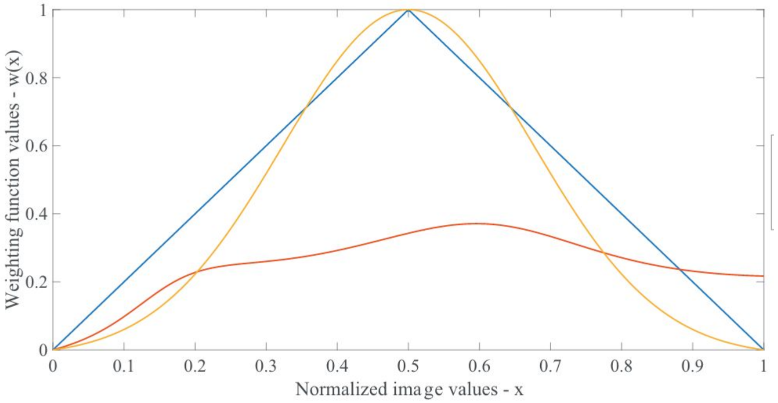
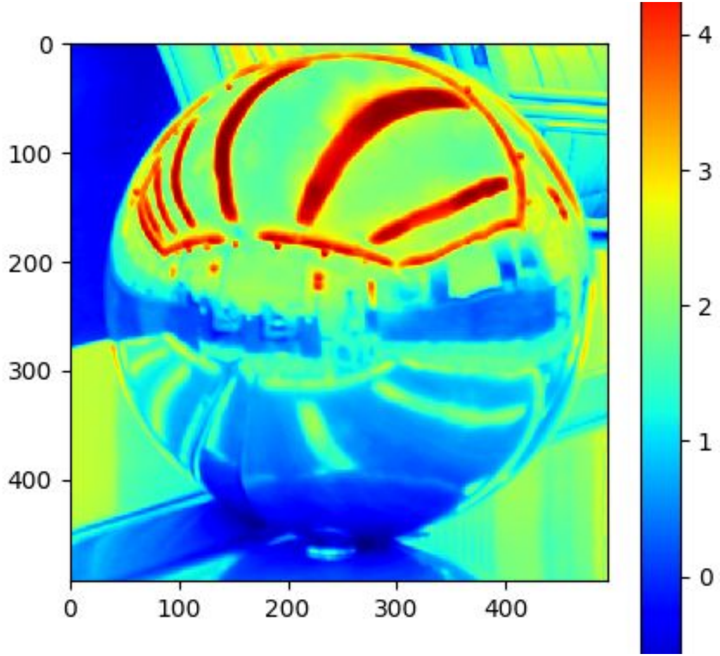


Image Merging



Visualization of HDR recovered

Panoramic Transformation

Different Exposures

1/80 1/40 1/20 1/10 1/5 1/2 0.77 1.6

HDR image of spherical mirror

Panoramic Transformation

Latitude/longitude format

Panoramic Transformation

Real mirror sphere

Virtual mirror sphere

Tone Mapping

Tone Mapping

Tonemap HDR image as background image for the IBL scene

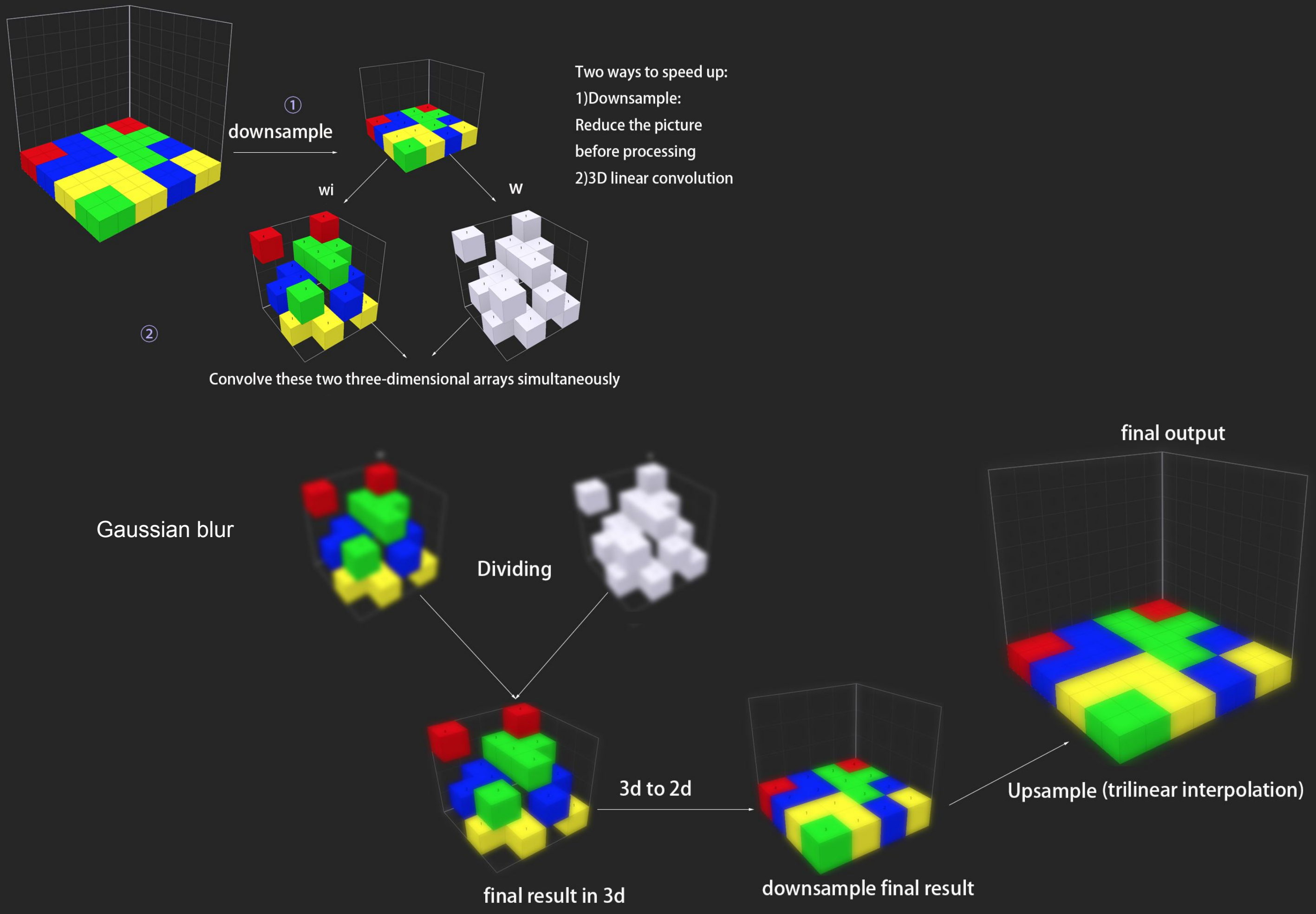
Fast Bilateral Filter & Local Laplacian Filter

Fast Bilateral Filter

Bilateral filtering is very useful but very slow because it is nonlinear. A new interpretation of bilateral filtering is proposed, namely, high-dimensional convolution, followed by two nonlinear operations. The basic idea is to change the nonlinear bilateral filtering into separable linear operation and nonlinear operation.

A Fast Approximation of the Bilateral Filter using a Signal Processing Approach
Sylvain Paris and Frédo Durand
MIT technical report 2006 (MIT–CSAIL–TR–2006–073)
high–res. pdf (25MB) | low–res. pdf (2MB)

Reference link: A Fast Approximation of the Bilateral Filter using a Signal Process Approach


Two ways to speed up:
1)Downsample:
Reduce the picture
before processing
2)3D linear convolution
①
downsample
wi
w
②
Convolve these two three-dimensional arrays simultaneously
final output
Gaussian blur
Dividing
3d to 2d
Upsample (trilinear interpolation)
final result in 3d
downsample final result

Fast Bilateral Filter

Input Image

Bilateral Filter

2.761s

sigmaDomain=1.0 sigmaRange = 0.1

TruncateDomain=3.0

Fast Bilateral Filter

0.845s

sigmaDomain=1.0 sigmaRange = 0.1

TruncateDomain=3.0 downSample=2.0

Fast Bilateral Filter

sigma Range =0.1 sigma Domain = 1.0 Truncate =3.0

downSample=1.0

2.805s

downSample=2.0

0.984s

downSample=4.0

0.579s

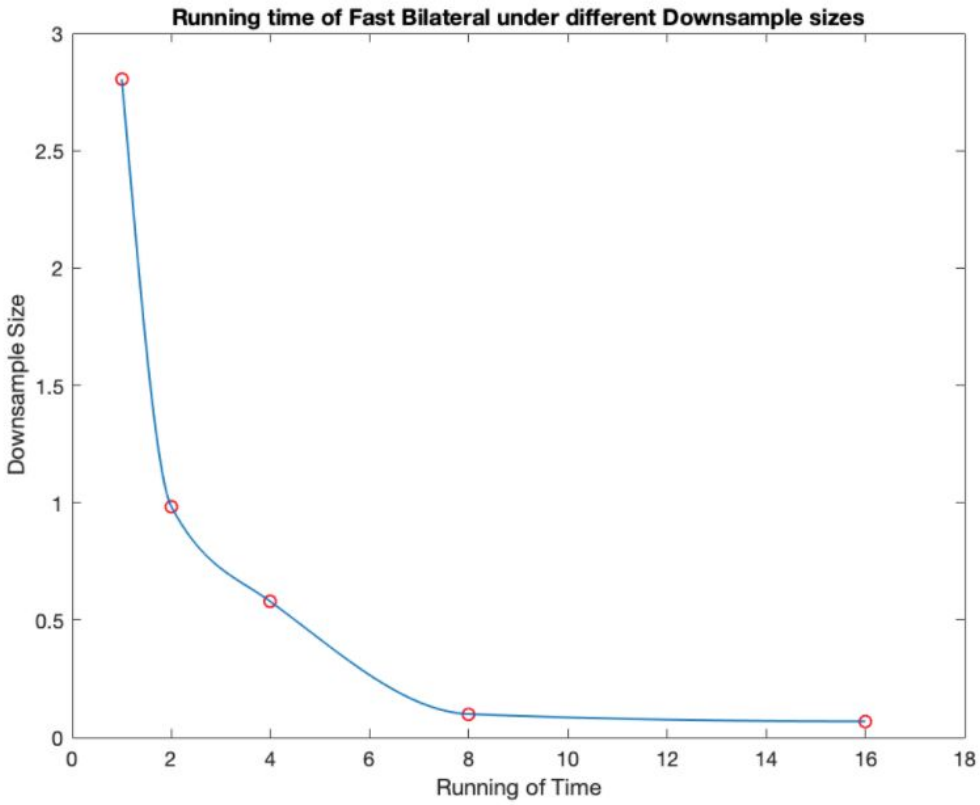
downSample=8.0

0.1s

downSample=16.0

0.069s

Fast Bilateral Filter



Fast Bilateral Filter

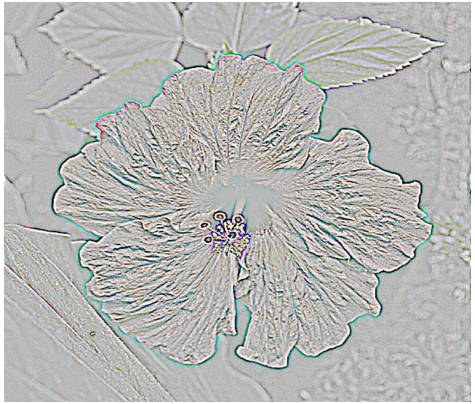
Image size:
1478*919

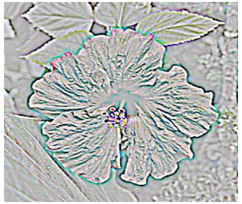
Downsample size:
8
Running time:
5.79s

Local Laplacian Filters

Edge-aware image filters based on the manipulation of Laplacian pyramids

Produces high quality results without degrading edges or introducing halos even at extreme setting


level 0


level 1


level 2


level 3
(residual)

Local Laplacian Filters

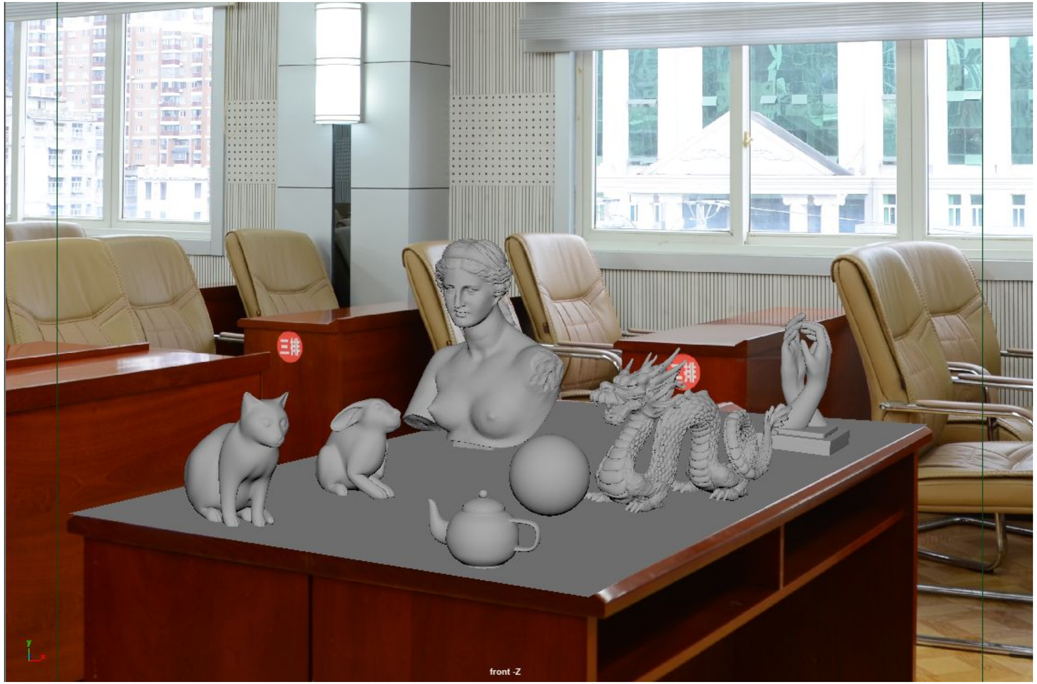
Tone-mapping (preserve details, α=1) result

Local Laplacian Filters

Different artistic styles by changing parameters

Modeling and Rendering

Modeling and Rendering

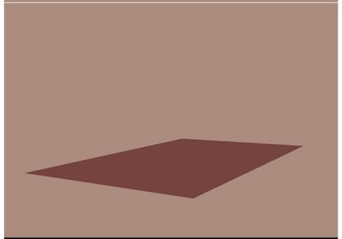


We imported the model in Maya and rendered the scene using the Arnold renderer

Background image

Rendered scene with objects

Rendered scene without objects

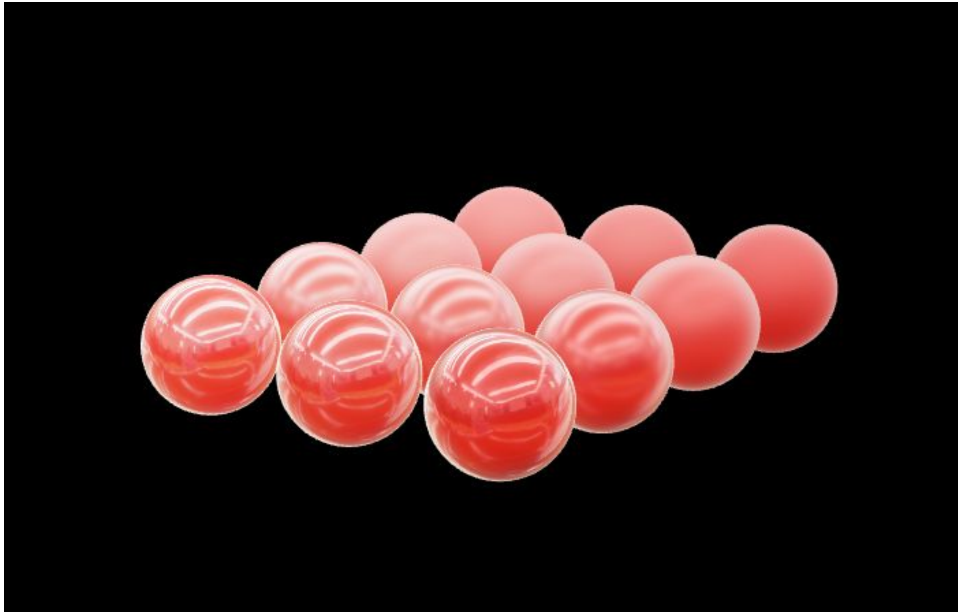


Mask

OpenGL and Custom Rendering

Objects composited onto scene

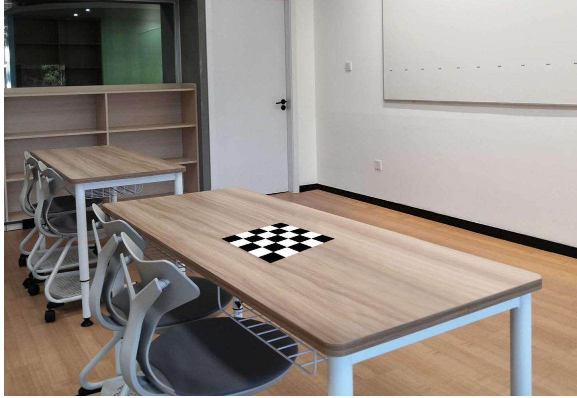
HDR "light source"

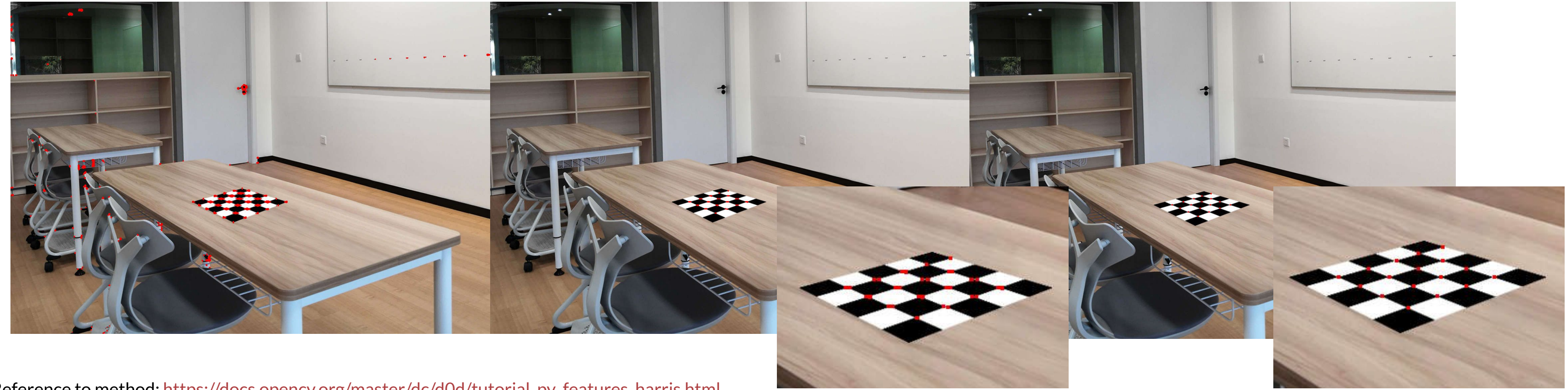


Spheres rendered in OpenGL

Harris Corner Detection

$$Change \text{ in intensity} = \sum_{x,y} \left[I(x+u, y+v) - I(x,y)\right]^2$$





Reference to method: https://docs.opencv.org/master/dc/d0d/tutorial_py_features_harris.html

Task Division

1. **Recovering HDR Image**
   a. Shoot and crop scenes with different exposures[Zhiyu Xiao]
   b. Response function estimation[Haowen Liu]
   c. LDR image Merging [Haowen Liu]
2. **Panoramic Transformation**
   a. Converted from mirror ball to latitude-longitude[Zhiyu Xiao]
3. **Tone Mapping**
   a. Fast bilateral filter[Zhiyu Xiao]
   b. Local laplacian filters [Haowen Liu]
4. **Modeling and Rendering**
   a. Model and render the scene in Maya [Zhiyu Xiao]
   b. Use rendered images to perform "differential render" compositing [Eammon Littler]
   c. OpenGL based local rendering [Eammon Littler]
   d. Harris corner detection [Eammon Littler]

References

1. Programming Project #4: Image-Based Lighting CS498
2. Advanced High Dynamic Range Imaging Book
3. A Fast Approximation of the Bilateral Filter using a Signal Processing Approach
4. Rendering Synthetic Objects into Real Scenes, by Paul Debevec, Siggraph 1998
5. Recovering High Dynamic Range Radiance Maps from Photographs, by Paul Debevec, Siggraph 1997
6. Local Laplacian Filters: Edge-aware Image Processing with a Laplacian Pyramid
7. Harris Corner Detection