Learning to Predict Indoor Illumination from a Single Image

Chih-Hui Ho

Outline

i-clicker

- Which picture is lit by groundtruth?
- (A)(C)
- (A)(D)
- (B)(C)
- (B)(D)
- (A)(B)

A

B

C

D

i-clicker

- Which picture is lit by groundtruth?
- (A)(C)
- (A)(D)
- (B)(C)
- (B)(D)
- (A)(B)

A

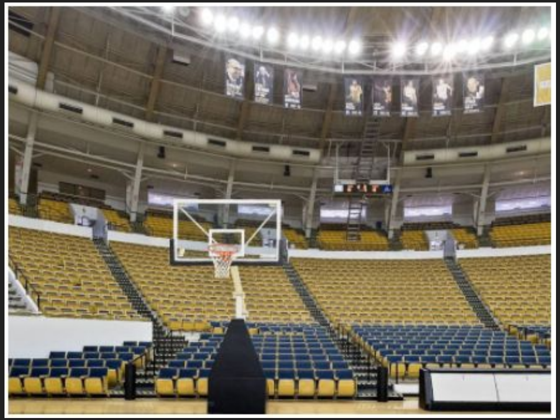
B

C

D

Introduction

- The goal is to render a virtual 3D object and make it realistic
- Inferring scene illumination from a single photograph is a challenging problem
- The pixel intensities observed in an image are a complex function of scene geometry, materials properties, illumination and the imaging device
- Harder from a single limited field-of-view image



Introduction

- Some methods
  - Assume that scene geometry or reflectance properties are given
    - Measured using depth sensors, or annotated by a user
  - Impose strong low-dimensional models on the lighting
    - Same scene can have wide range of illuminants
- State-of-the-art techniques are still significantly error-prone
- Is it possible to infer the illumination from an image ?

Introduction

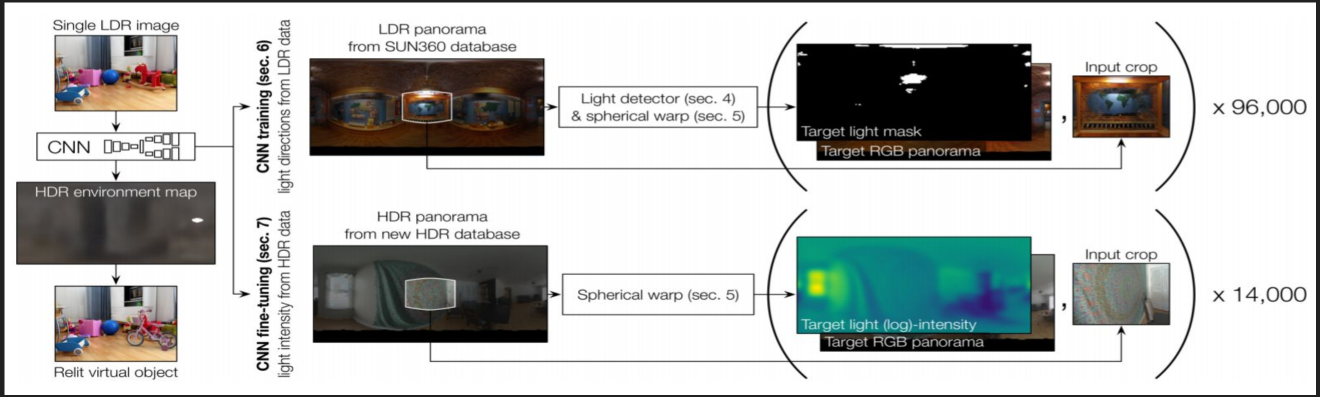
- Dynamic range is the ratio between brightest and darkest parts in the image
- High dynamic range (HDR) vs Low dynamic range (LDR)
- HDR image stores pixel values that span the whole range of real world scene
- LDR image stores pixel value within some range (i.e. JPEG 255:1)

Introduction

- An automatic method to infer HDR illumination from a single, limited field-of-view, LDR photograph of an indoor scene
  - Model the range of typical indoor light sources
  - Robust to errors in geometry, surface reflectance, and scene appearance
  - No strong assumptions on scene geometry, material properties, or lighting
- Introduce an end-to-end deep learning based approach
  - Input: A single, limited field-of-view,LDR image
  - Output: A relit virtual object in HDR image
- Application: 3D object insertion
- Everything looks perfect so far

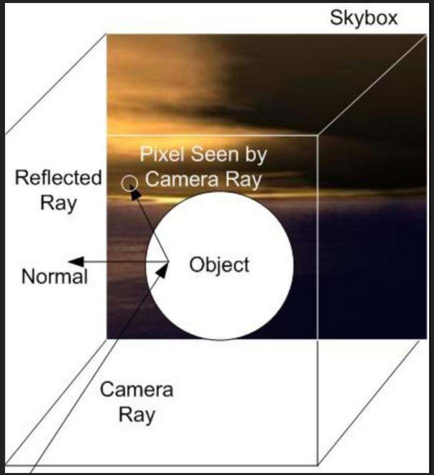
Method Overview

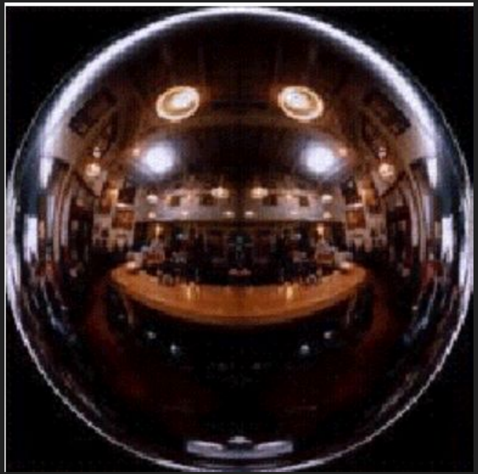
- Two stage training scheme is proposed to train the CNN
  - Stage 1 (96000 training data)
    - Input : LDR, limit field-of-view image
    - Output: target light mask, target RGB panorama
  - Stage 2 (fine tuning) (14000 training data)
    - Input: HDR, limit field-of-view image
    - Output: target light (log) intensity, target RGB panorama



Environment Map

- In computer graphics, environment mapping is an image based lighting technique for approximating a reflective surface
- Cubic mapping
- Sphere mapping
    - Consider the environment to be an infinitely far spherical wall
    - Orthographic projection is used
    - Used by the paper



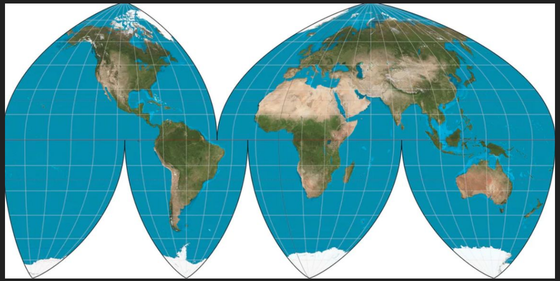


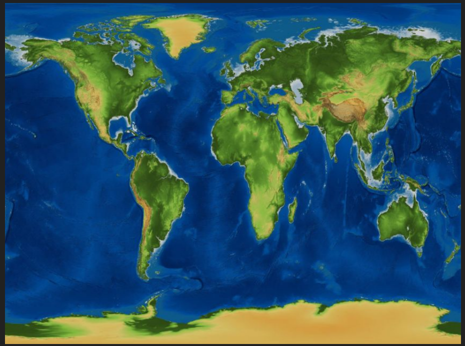
Method Overview

- What is the problem to train deep NN to learn image illuminations ?
    - Lots of HDR data (Not currently exists)
    - We do have lots of LDR data (Sun 360)
    - But light source are not explicitly available in LDR images
    - LDR images does not capture lighting properly
- Predict HDR lighting conditions from a LDR panoramas
- Now we have the ground truth for HDR lighting mask/ position
- We need an input image patch

Spherical Panorama

- Equirectangular projection: project a spherical image on to a flat plane
- Large distortion at pole
- Rectification is needed





Method Overview

- Extract the training patches from the panorama
- Rectify the cropped patches
- Now we have data {Image,HDR light probe} to train the lighting mask
- How about target RGB panorama ?

Method Overview

- There are still some problems
  - The panorama does not represent the lighting conditions in the cropped scene
  - Center of projection of panorama can be far from the cropped scene
- Panorama warping is needed
- What is warping ?
  - Image warping is a way to manipulate an image to the way we want
  - Image resampling/ mapping
- Now we are ready for stage 1

http://www.cs.princeton.edu/courses/archive/spr11/cos426/notes/cos426_s11_lecture03_warping.pdf

Method Overview

- In stage 2, light intensity is estimated
- LDR images are not enough
- 2100 HDR image dataset are collected
- Fine tune the CNN
- Use light intensity map and RGB panorama to create a final HDR environment map
- Relit the virtual objects

LDR Panorama Light Source Detection

- Goal: detect bright light sources in LDR panoramas and use them as CNN training data
- Data
  - Manually annotate a set of 400 panoramas from the SUN360 database
  - Light sources: spotlights, lamps, windows, and (bounce) reflections
  - Discard the bottom 15% of the panoramas because of watermarks and few light source
  - 80% data for training and 20% data for testing
  - Labeled lights as positive samples and random negative samples

LDR Panorama Light Source Detection

- Training phase
  - Convert panorama into grayscale
  - Panorama P is rotated to get P_rot
    - Large distortion caused by equirectangular projection
    - Aligning zenith with the horizontal line
  - Compute patch features over P and P_rot at different scale
    - Histogram of Oriented Gradient (HOG)
    - Mean, standard deviation and 99th percentile intensity values
  - Train 2 logistic regression classifiers
    - Small light sources (spotlight, lamps)
    - Large light sources (window, reflections)
    - Hard negative mining is used over the entire training set

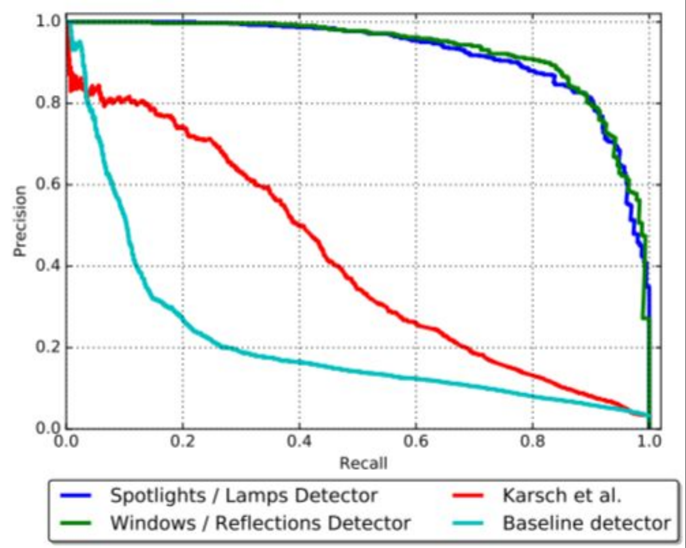
LDR Panorama Light Source Detection

- Testing phase
  - Logistic regression classifiers are applied to P and P_{rot} in a sliding-window fashion
  - Each pixel has 2 scores (one from each classifier)
  - Define S*rot is Srot rotated back to the original orientation
  - $S_{merged} = S^{*}\cos(theta)+S^{*}_{rot}{}^{*}\sin(theta)$, and theta is pixel elevation
  - Threshold the score to obtain a binary mask
    - Optimal threshold is obtained by maximizing the intersection over union (IoU) score between the resulting binary mask and the ground truth labels on the training set
  - Refined with a dense CRF
  - Adjusted with opening and closing morphological operations

LDR Panorama Light Source Detection

LDR Panorama Light Source Detection

- Results
  - A baseline detector relying solely on the intensity of a pixel
  - The proposed method has high recall and precision



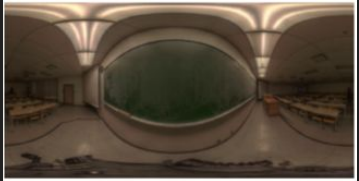
$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

Panorama Recentering Warp

- Goal: To solve problem that panorama does not represent the lighting conditions in the cropped scene
- Treating this original panorama as a light source is incorrect
- No access to the scenes to capture ground truth lighting
- Approximate the lighting in the cropped photo by warping

Original



Groundtruth

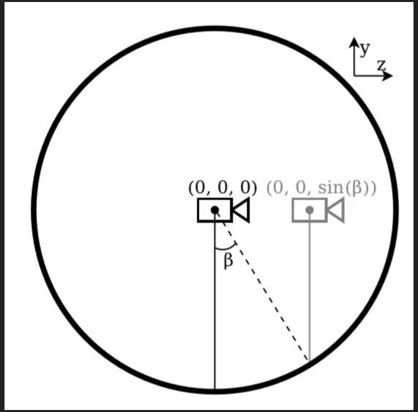
Warp result

Panorama Recentering Warp

- Generate a new panorama by placing a virtual camera at a point in the cropped photo
- No scene geometry information is given
- Assumption
  - All scene points are equidistant from the original center of projection
  - Image warping suffices to model the effect of moving the camera
  - Lights that illuminate a scene point, but are not visible from the original camera are not handled (Occlusion)
  - Panorama is placed on a sphere
- $x^2 + y^2 + z^2 = 1$ must hold

Panorama Recentering Warp

- Outgoing rays emanating from a virtual camera placed at (x_0,y_0,z_0)
- $x(t) = v_x{*}t + x_0$, $y(t) = v_y{*}t + y_0$, $z(t) = v_z{*}t + z_0$
- $(v_x t + x_0)^2 + (v_y t + y_0)^2 + (v_z t + z_0)^2 = 1$
- Example: Model the effect of using a virtual camera whose nadir is at β (translate along z axis)
- $\{x_0,y_0,z_0\}=\{0,0,\sin\beta\}$.
- $(v^2_x + v^2_y + v^2_z)t^2 + 2 v_z t \sin\beta + \sin^2\beta - 1 = 0$
- Solve t
- Maps the coordinates to warped camera coordinate system
- How can we determine β ?



Panorama Recentering Warp

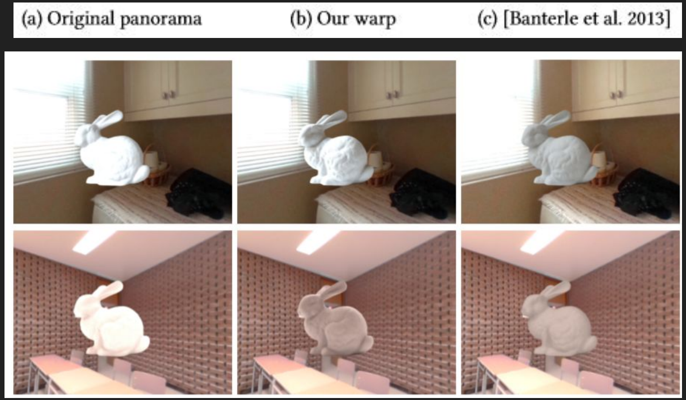
- Assume users want to insert objects on to flat horizontal surfaces in the photo
- Detect surface normals in the cropped image [Bansal et al. 2016]
- Find flat surfaces by thresholding based on the angular distance between surface normal and the up vector
- Back project the lowest point on the flattest horizontal surface onto the panorama to obtain β

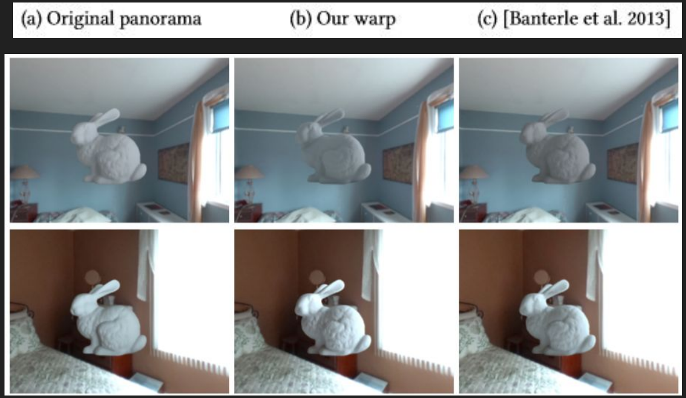
(a) Input image (b) Normals

(c) Original panorama (d) Warped panorama

Panorama Recentering Warp

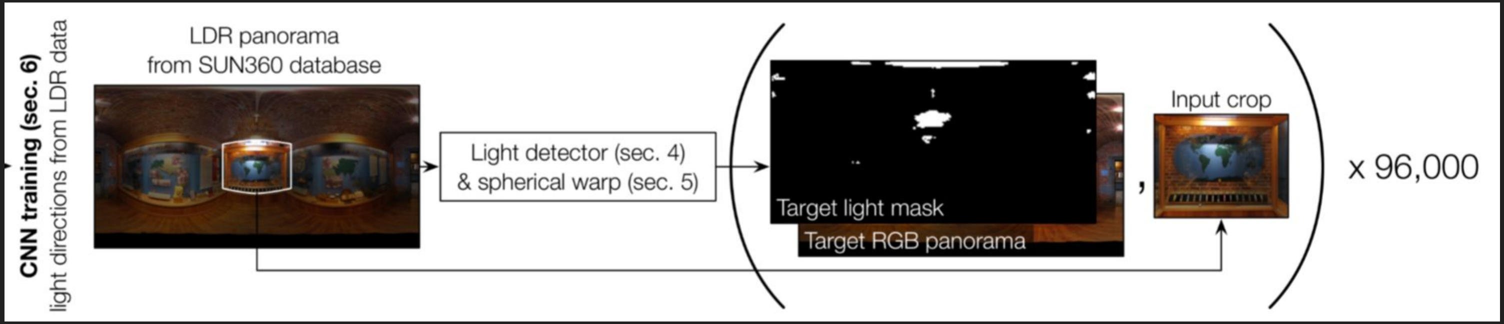
- EnvyDepth [Banterle et al. 2013] is a system that extracts spatially varying lighting from environment maps (ground truth approximation)
- EnvyDepth needs manual annotating, requires access to scene geometry and takes about 10 min per panorama
- The proposed system is automatic and does not require scene information
- Comparable result with EnvyDepth





Learning from LDR Panoramas

- Ready to train a CNN
- Input: a LDR photo
- Output: a pair of warped panorama and corresponding light mask
- Data
  - For each SUN360 indoor panorama, compute the groundtruth light mask
  - For each SUN360 indoor panorama, take 8 crops with random elevation between +/−30º
  - 96,000 input-output pairs



Learning from LDR Panoramas

- Learn the low-dimensional encoding (FC-1024) of input (256×192)
- 2 individual decoders are composed of deconvolution layers
  - RGB panorama prediction (256×128)
  - Binary light mask prediction (256×128)
- Loss

RGB panorama prediction

$$\mathcal{L}_{L2}(\mathbf{y},\mathbf{t}) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{s}_i(\mathbf{y}_i - \mathbf{t}_i)^2$$

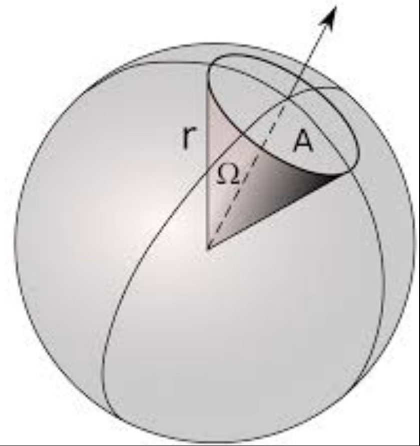
Binary light mask prediction

$$\mathcal{L}_{\cos}(\mathbf{y},\mathbf{t},e) = \frac{1}{N}\sum_{i=1}^{N}(\mathcal{F}(\mathbf{y},i,e) - \mathcal{F}(\mathbf{t},i,e))^2$$

$$\mathcal{F}(\mathbf{p},i,e) = \frac{1}{K_i}\sum_{\omega\in\Omega_i} \mathbf{p}(\omega)s(\omega)\,(\omega\cdot n_i)^{\alpha e}$$

Layer (stride)	
Input	
conv9-64 (2)	
conv4-96 (2)	
res3-96 (1)	
res4-128 (2)	
res4-192 (2)	
res4-256 (2)	
FC-1024	
FC-8192	FC-6144
deconv4-256 (2)	deconv4-192 (2)
deconv4-128 (2)	deconv4-128 (2)
deconv4-96 (2)	deconv4-64 (2)
deconv4-64 (2)	deconv4-32 (2)
deconv4-32 (2)	deconv4-24 (2)
conv5-1 (1)	conv5-3 (1)
Sigmoid	Tanh
Output: light mask $\mathbf{y}_{\text{mask}}$	Output: RGB panorama $\mathbf{y}_{\text{RGB}}$

Closer Look to RGB Loss

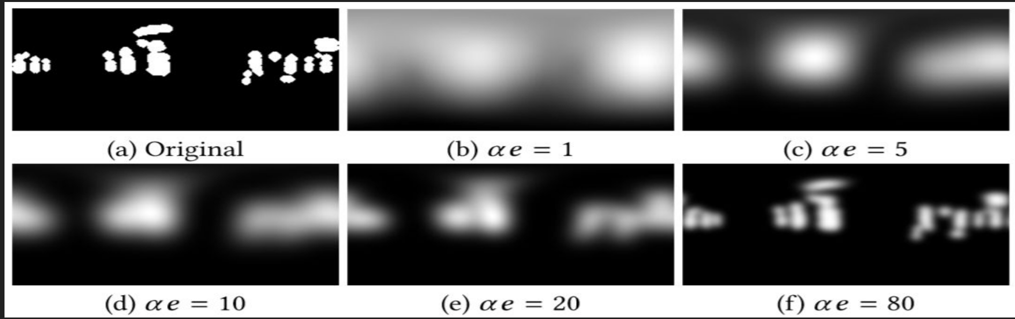


- What is solid angle?
- Informal definition
  - Take a surface
  - Project it onto a unit sphere (a sphere of radius 1)
  - Calculate the surface area of your projection.
- It is defined as **$\Omega = A / r^2$**
- Every pixel in the image corresponds to certain solid angle in the sphere
- This is a weighted loss

$$\mathcal{L}_{\mathrm{L2}}(\mathbf{y}, \mathbf{t}) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{s}_i (\mathbf{y}_i - \mathbf{t}_i)^2$$

Closer Look to Mask Loss

- Why not L2 loss ?
- If a spotlight is predicted to be slightly off its ground truth location, a huge penalty will incur
- Pinpointing the exact location of the light sources is not necessary
- Instead, learn the mask gradually by blurring the groundtruth and progressively sharpens it over training time
- Blurriness is a function of epoch

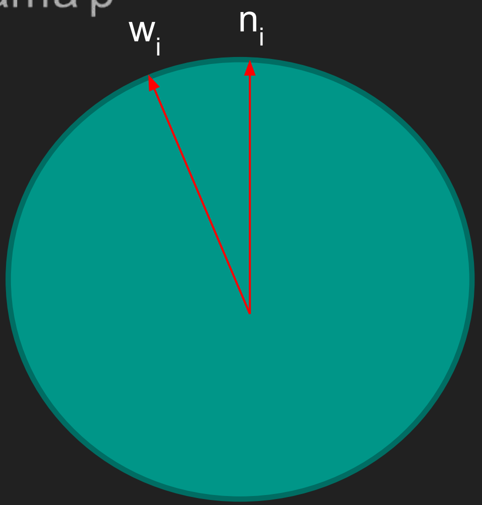


(a) Original (b) $\alpha e = 1$ (c) $\alpha e = 5$ (d) $\alpha e = 10$ (e) $\alpha e = 20$ (f) $\alpha e = 80$

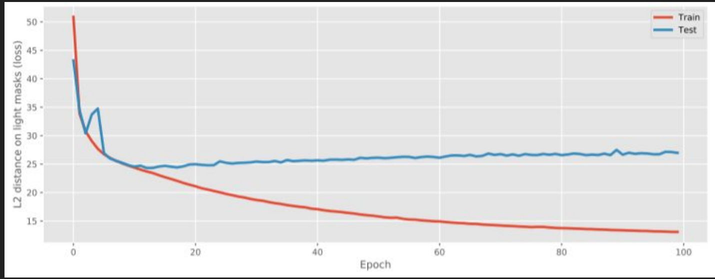
Closer Look to Mask Loss

$$\mathcal{F}(\mathbf{p}, i, e) = \frac{1}{K_i} \sum_{\omega \in \Omega_i} \mathbf{p}(\omega) s(\omega) (\omega \cdot n_i)^{\alpha e}$$

- Cosine distance filter
- Ω_i is the hemisphere centered at pixel i on the panorama p
- n_i the unit normal at pixel i
- K the sum of solid angles on Ω_i
- ω is a unit vector in a specific direction on Ω_i
- s(ω) the solid angle for the pixel in the direction ω
- p(ω) is the pixel value in the direction ω
- Note that (w*n_i) is the angle between neary pixels
- This is cos(theta)
- 0 <= cos(theta) <= 1
- So as α*e increase, we only blur the pixels that is closed to pixel i



Learning from LDR Panoramas



- Global loss function
- w1 = 100, w2 = 1, and α = 3
- Training phase
  - 85% of the panoramas as training data and 15% as test data
- Testing phase
  - All tests are performed for scenes and lighting conditions that have not been seen by the network
  - Lighting inference (both mask and RGB) from a photo takes approximately 10ms on an Nvidia Titan X Pascal GPU

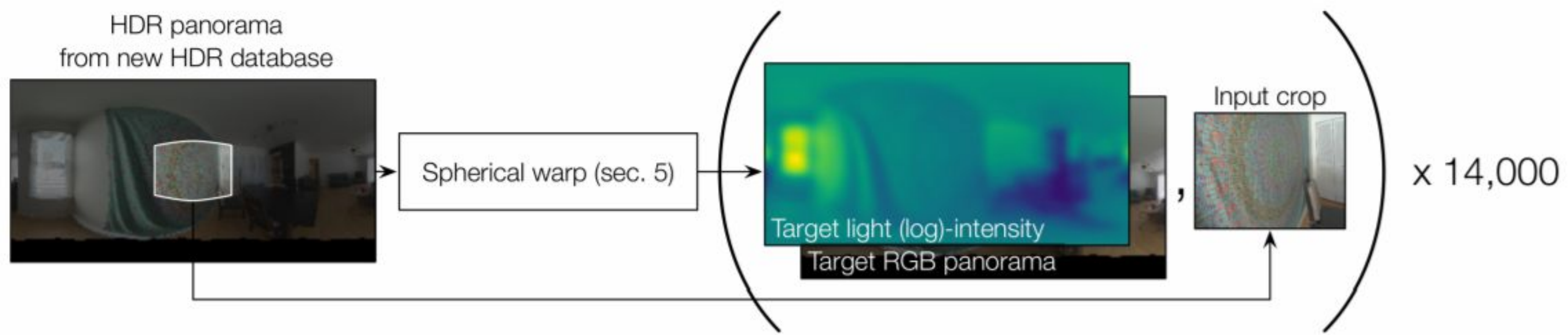
$$\mathcal{L}(\mathbf{y}, \mathbf{t}, e) = w_1 \mathcal{L}_{\mathrm{L2}}(\mathbf{y}_{\mathrm{RGB}}, \mathbf{t}_{\mathrm{RGB}}) + w_2 \mathcal{L}_{\cos}(\mathbf{y}_{\mathrm{mask}}, \mathbf{t}_{\mathrm{mask}}, e)$$

Learning High Dynamic Range Illumination

- Goal: Predict intensities of the light sources
- LDR data is not enough
- 2100 HDR indoor panoramas dataset (high-resolution (7768 × 3884))
- The dynamic range is sufficient to correctly expose all pixels in the scenes, including the light sources.

Learning High Dynamic Range Illumination

- Data
    - 85% of the HDR data was used for training and 15% for testing
    - 8 crops were extracted from each panorama in the HDR dataset, yielding 14,000 input-output pairs
    - Panoramas are warped using the same procedure as LDR

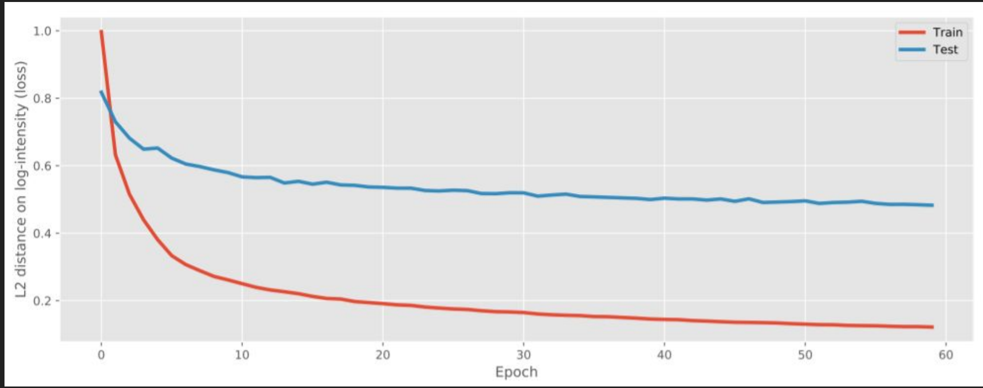


Learning High Dynamic Range Illumination

$$\mathcal{L}_{\mathrm{HDR}}(\mathbf{y}, \mathbf{t}, e) = w_1 \mathcal{L}_{\mathrm{L2}}(\mathbf{y}_{\mathrm{RGB}}, \mathbf{t}_{\mathrm{RGB}}) + w_2 \mathcal{L}_{\cos}(\mathbf{y}_{\mathrm{int}}, \mathbf{t}_{\mathrm{int}}, e) + w_3 \mathcal{L}_{\mathrm{L2}}(\mathbf{y}_{\mathrm{int}}, \mathbf{t}_{\mathrm{int}}, e)$$

- Training phase
  - Fine tuning on HDR dataset to learn the light source intensities
  - Conv5-1 weights are randomly re-initialized
  - Fix weights before FC 1024
  - Target intensity t_{int} is defined as the log of the HDR intensity
  - Low intensities are clamped to 0
  - Epoch e is continued from training on the LDR data

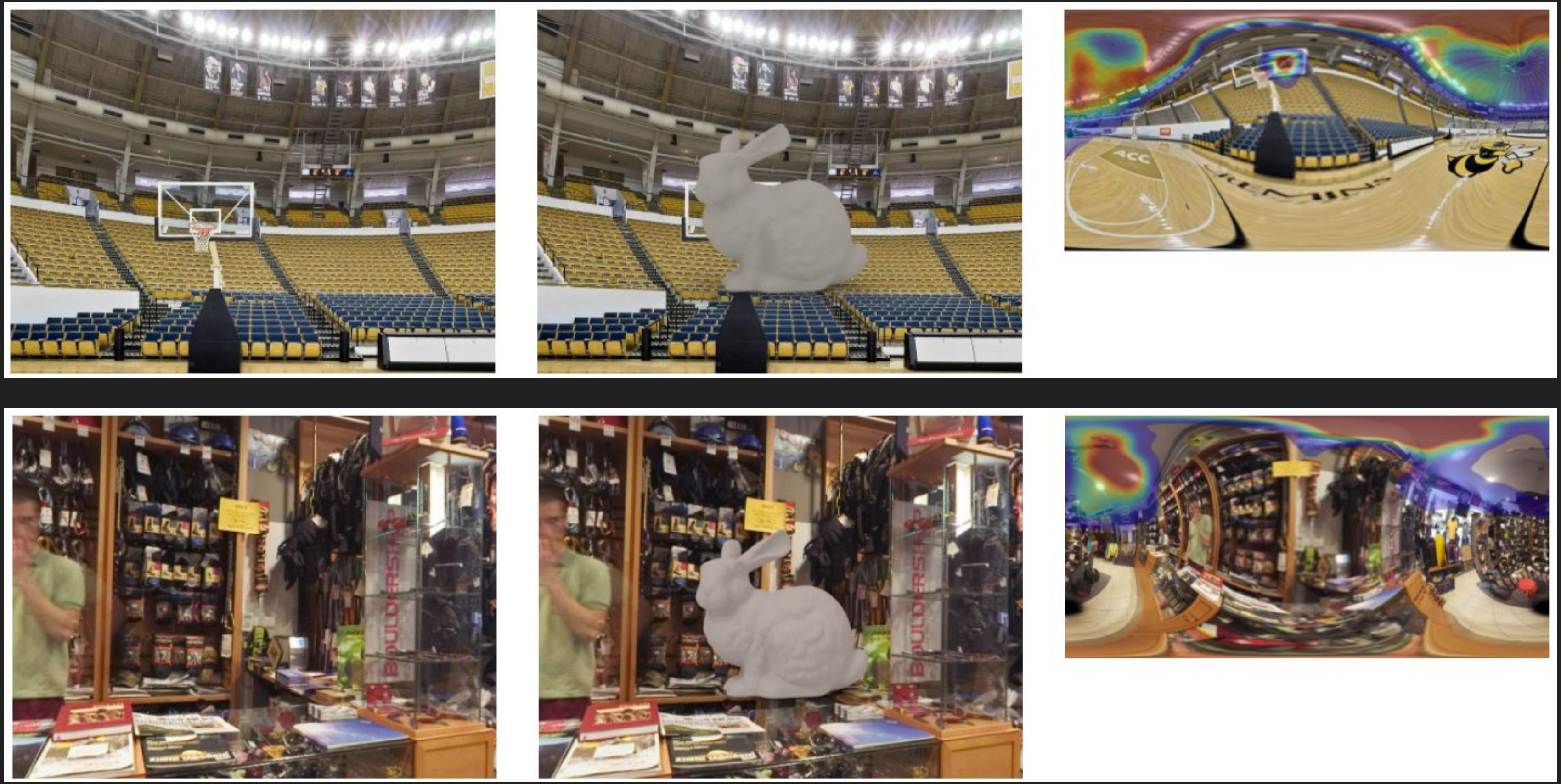
Layer (stride)
Input
conv9-64 (2)
conv4-96 (2)
res3-96 (1)
res4-128 (2)
res4-192 (2)
res4-256 (2)
FC-1024
FC-8192
deconv4-256 (2)
deconv4-128 (2)
deconv4-96 (2)
deconv4-64 (2)
deconv4-32 (2)
conv5-1 (1)
Sigmoid
Output: light mask $\mathbf{y}_{\mathrm{mask}}$



Experiment -- LDR Network

- Light prediction results on the SUN360 dataset (LDR data)
- Evaluate by rendering a virtual bunny model into the image

Experiment -- LDR Network

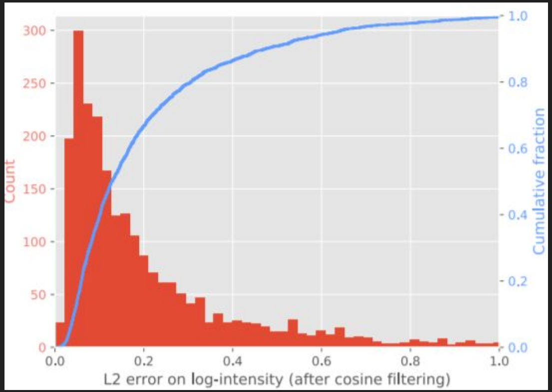


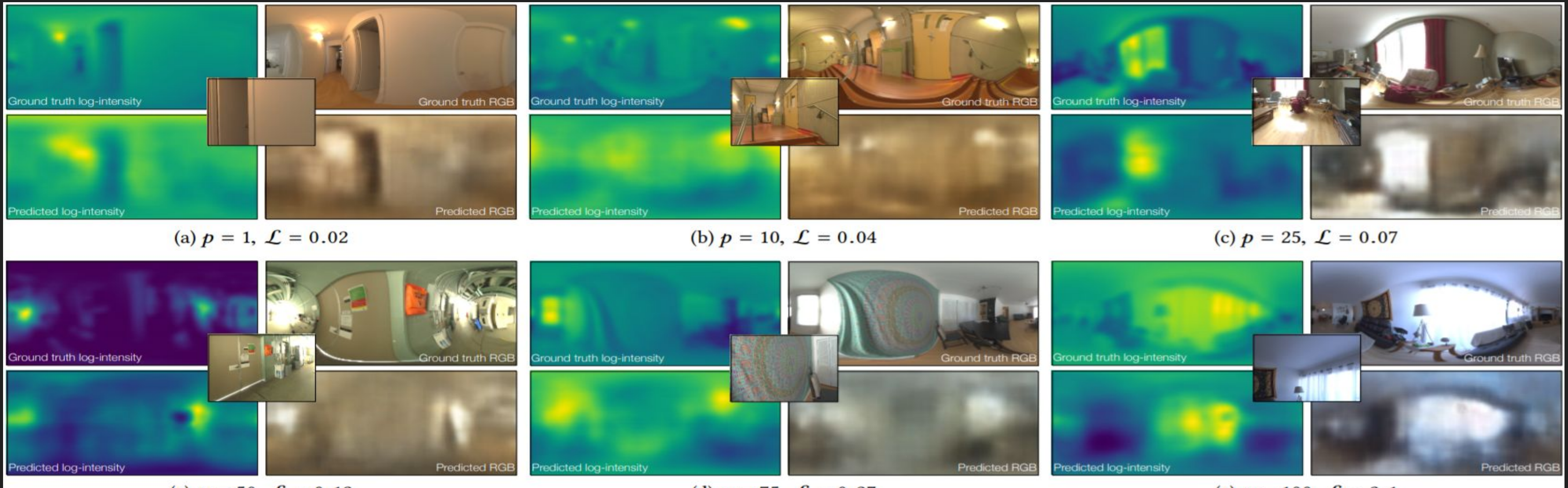
Experiment -- LDR Network

- Warping panorama cannot handle occlusions
- Even though the window causing the shadows on the handle in the image (left) is occluded in the panorama (right), the network places the highest probability of a light in this direction

Experiment -- HDR Network

- 2100 images are tested
- Ground truth log-intensities range is [0.04, 3.01]
- Yellow (high intensity) vs Blue (low intensity)



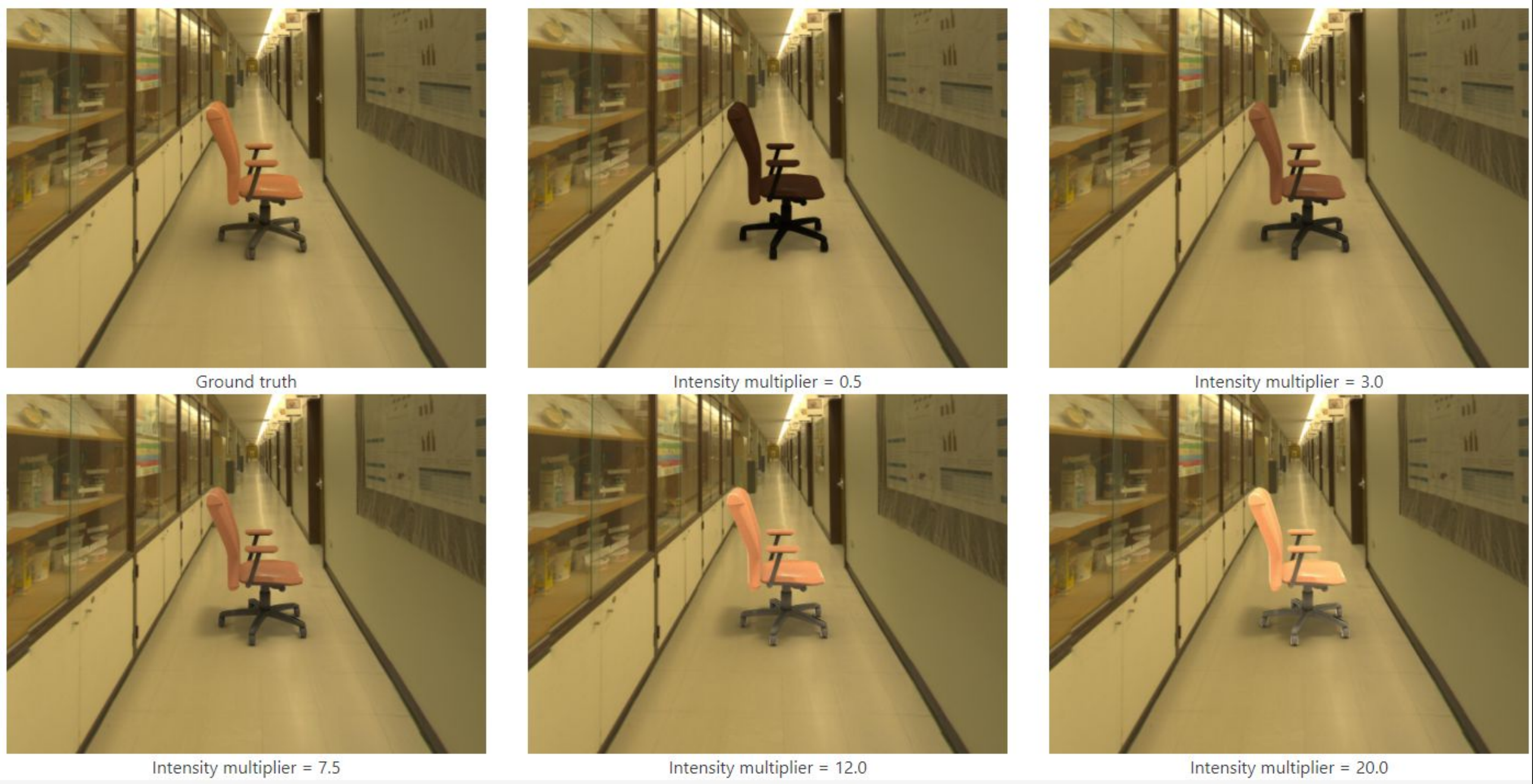


(a) $p = 1,\ \mathcal{L} = 0.02$
(b) $p = 10,\ \mathcal{L} = 0.04$
(c) $p = 25,\ \mathcal{L} = 0.07$
(c) $p = 50,\ \mathcal{L} = 0.13$
(d) $p = 75,\ \mathcal{L} = 0.27$
(e) $p = 100,\ \mathcal{L} = 3.1$

Experiment -- HDR Network

- The HDR network output can generate a HDR environment map
- $x_{combined} = 10^{x\_mask} + x_{RGB}$
- Recovering only the relative illumination intensities
- Matched the mean RGB value of the RGB prediction and the color of the light
- Able to select a global intensity scaling parameter

Experiment -- HDR Network


Ground truth
Intensity multiplier = 0.5
Intensity multiplier = 3.0
Intensity multiplier = 7.5
Intensity multiplier = 12.0
Intensity multiplier = 20.0

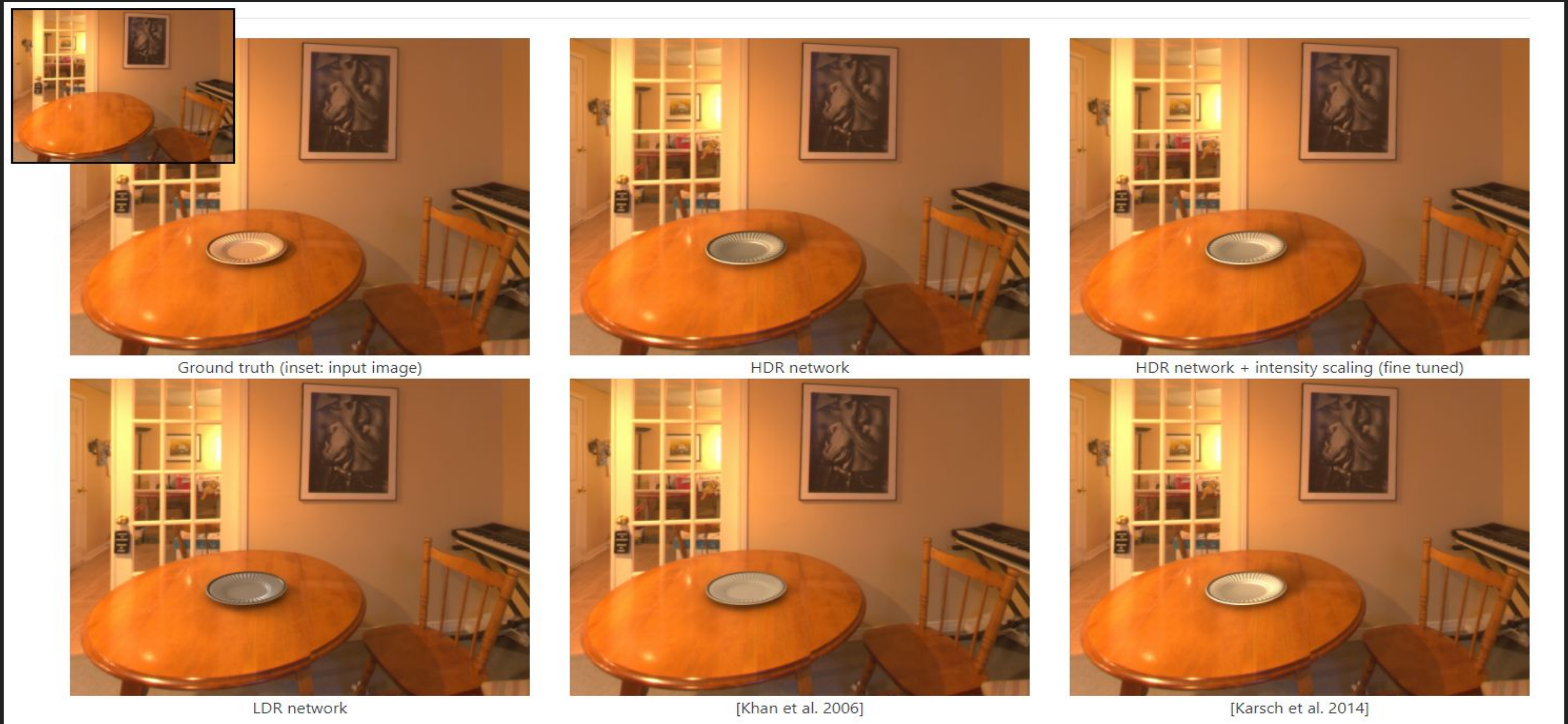
Experiment -- HDR Network

- Khan et al. [2006]
  - Estimate the illumination conditions by projecting the background image on a sphere
  - Fails to estimate the proper dynamic range and position of light sources
- Karsch et al. [2014]
  - Use a light classifier to detect in-view lights, estimate out-of-view light locations by matching the background image to a database of panoramas
  - Estimate light intensities using a rendering-based optimization
  - Relies on reconstructing the depth and the diffuse albedo of the scene
  - Panorama matching is based on image appearance features that are not necessarily correlated with scene illumination
- Proposed method
  - Robust estimates of lighting direction and intensity
  - Learn direct mapping between image appearance and scene illumination

Experiment -- HDR Network

(a) Ground truth lighting (b) Our HDR network (c) HDR network, intensity tuned (d) [Khan et al. 2006] (e) [Karsch et al. 2014]

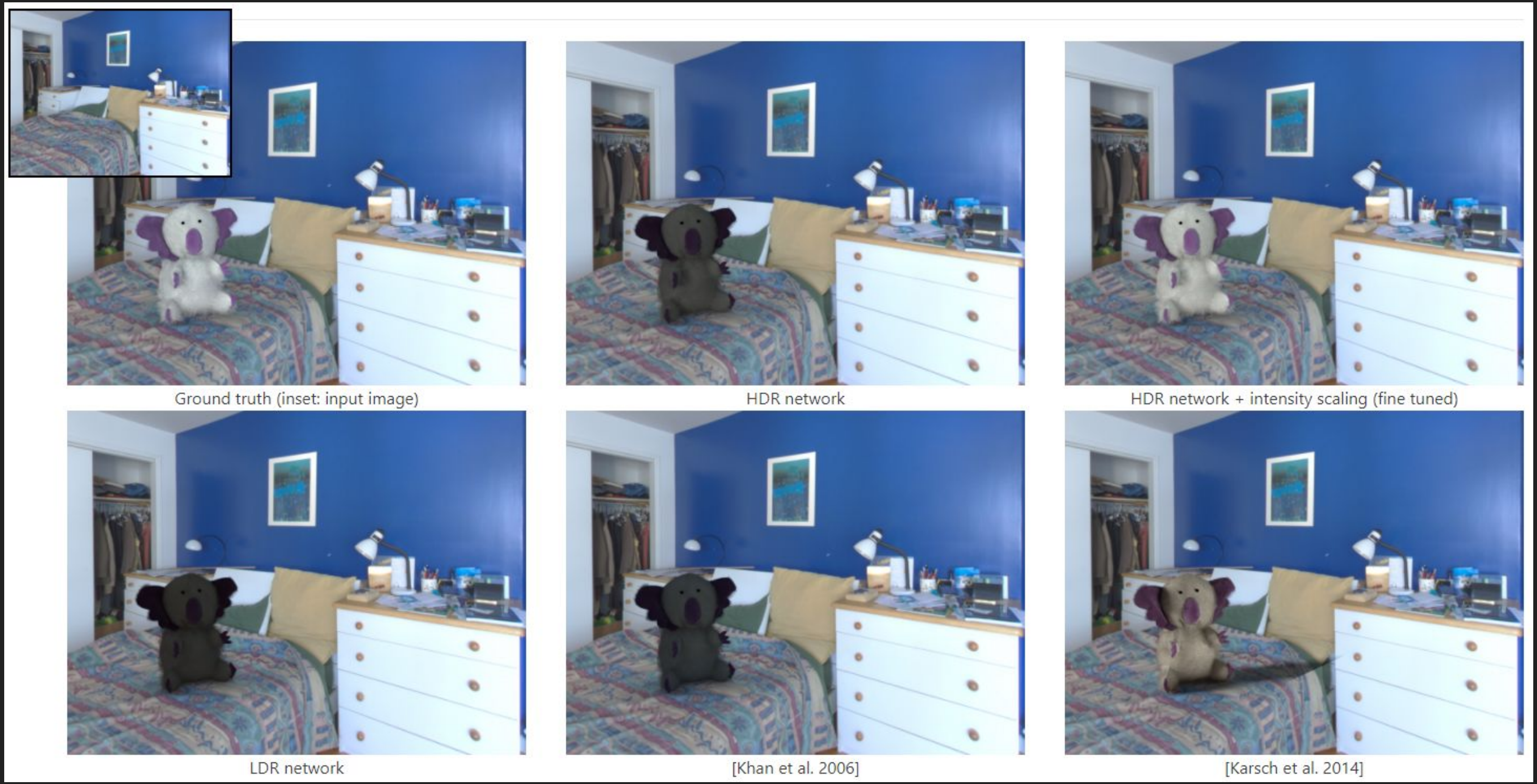
Experiment -- HDR Network



Experiment -- HDR Network

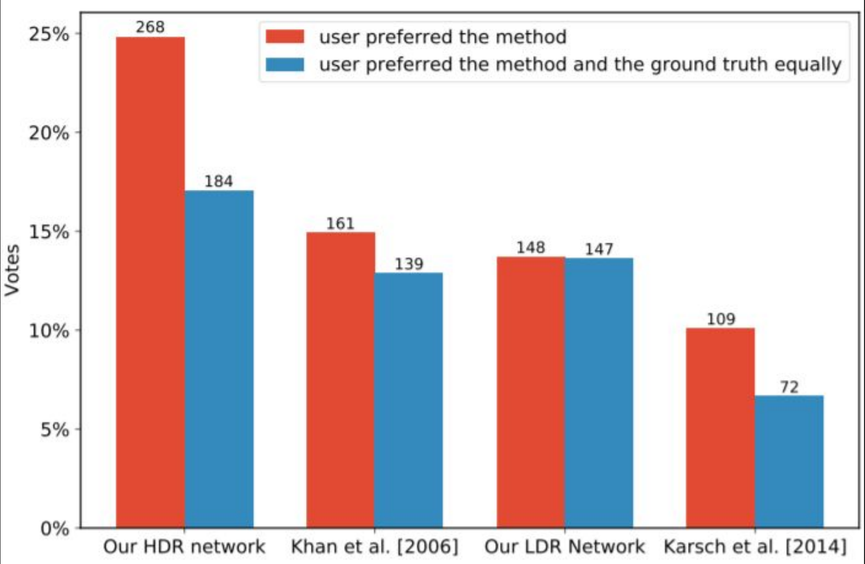
Experiment -- HDR Network

Experiment -- HDR Network



User study

- How realistic do synthetic objects lit by our estimates look when they are composited into input images?
- Showed users a pair of images — ground truth vs one of the methods



Conclusion and Future Work

- An end-to-end illumination estimation method that leverages a deep convolutional network to take a limited-field-of-view image as input and produce an estimation of HDR illumination
- A state-of-the-art light source detection method for LDR panoramas and a panorama warping method
- A new HDR environment map dataset

Conclusion and Future Work

- Some issues cause by filtering
  - Not accurate in inferring the spatial extent and orientation of light sources, particularly for out-of-view lights
  - Large area lights might be detected as smaller lights
  - Sharp light sources get blurred out
- Network is better at recovering the light source locations than intensity
  - Larger LDR training set than HDR training set fine-tuning step
- Indoor illumination is localized
  - Recovering spatially-varying lighting distribution is challenging

Reference

- http://vision.gel.ulaval.ca/~jflalonde/projects/deepIndoorLight/
- http://indoor.hdrdb.com/datapreview.html
- https://en.wikipedia.org/wiki/Tone_mapping
- https://computergraphics.stackexchange.com/questions/4185/why-is-spherical-harmonics-used-in-low-frequency-graphics-data-instead-of-a-sphe/4186
- https://en.wikipedia.org/wiki/Rendering_(computer_graphics)
- https://en.wikipedia.org/wiki/Rendering_equation
- https://en.wikipedia.org/wiki/Sphere_mapping
- https://en.wikipedia.org/wiki/Reflection_mapping
- https://www.youtube.com/watch?annotation_id=annotation_1471204287&feature=iv&src_vid=xutvBtrG23A&v=_Ix5oN8eC1E
- https://www.youtube.com/watch?v=xutvBtrG23A
- https://jmonkeyengine.github.io/wiki/jme3/advanced/pbr_part3.html
- http://people.csail.mit.edu/jxiao/SUN360/
- https://en.wikipedia.org/wiki/Equirectangular_projectio