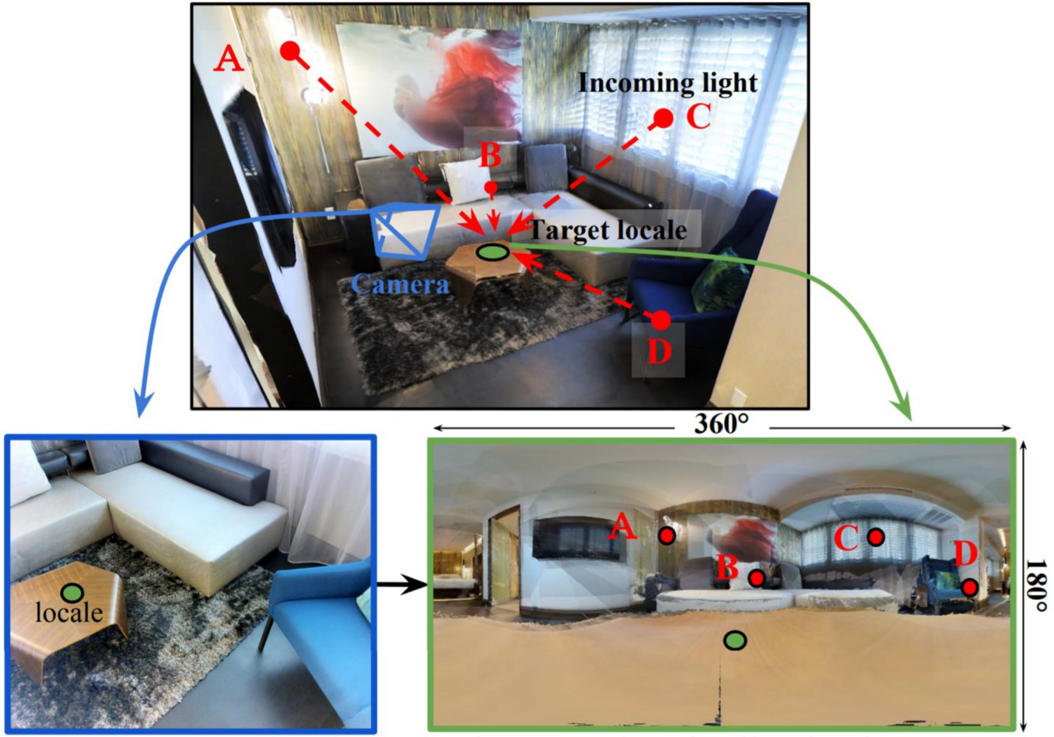
Neural Illumination: Lighting Prediction for Indoor Environments

Authors: Shuran Song, Thomas Funkhouser
Presenters: Clarice Roo, Shivang Soni, Nicolas Buxbaum

Goal:

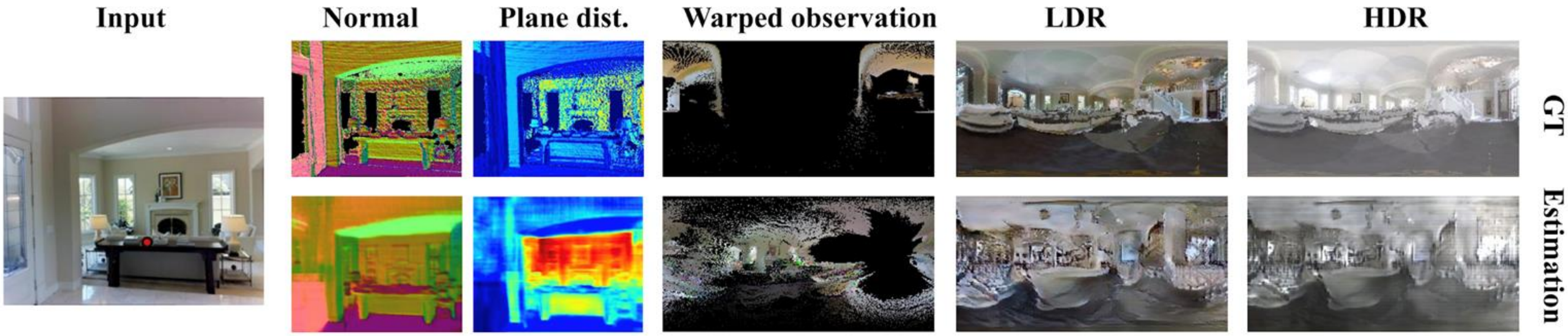
Estimate a high dynamic range panoramic illumination map of the entire scene from an input image and chosen locale



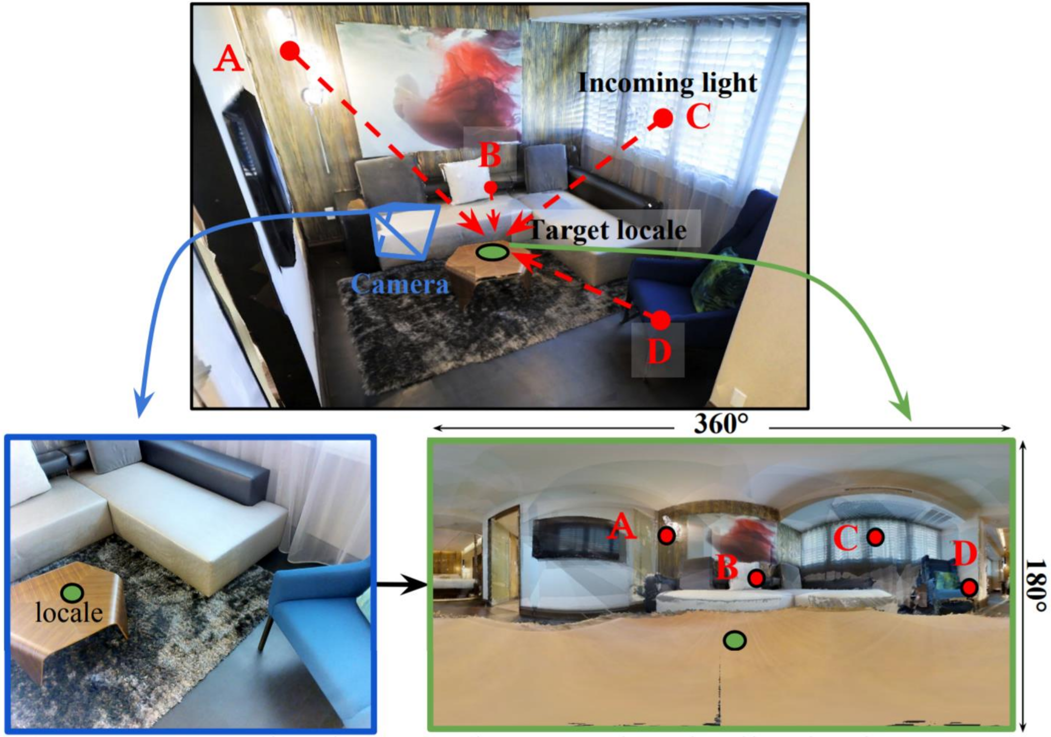
(a) Input: Image+locale (b) Ground truth: Illumination map

Background

- Illumination map- a map that encodes the incident radiance arriving from every direction at the 3D location associated with the selected pixel
- Dynamic range is the ratio of the highest value to lowest value of the pixels in an image
- Low dynamic range (LDR)- dynamic range 1:255
- High dynamic range (HDR)- dynamic range 1:70,000



Motivations and Challenges



(a) Input: Image+locale
(b) Ground truth: Illumination map

- Used to improve lighting in rendering
- Requires comprehensive understanding of the lighting environment
  - 3D location of selected pixel
  - 3D scene geometry to fill in occlusions
  - Distribution of unobserved light sources
  - Missing high dynamic range information

Related Work- Capture Based Methods

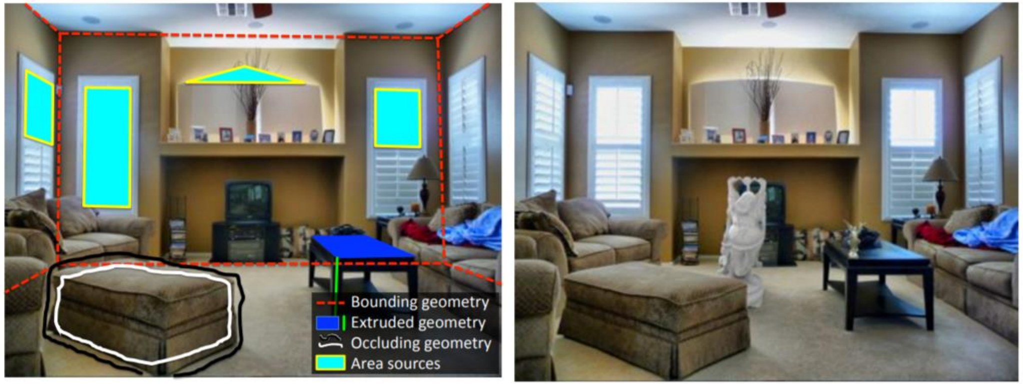
Capture Based Methods for obtaining illumination of an environment

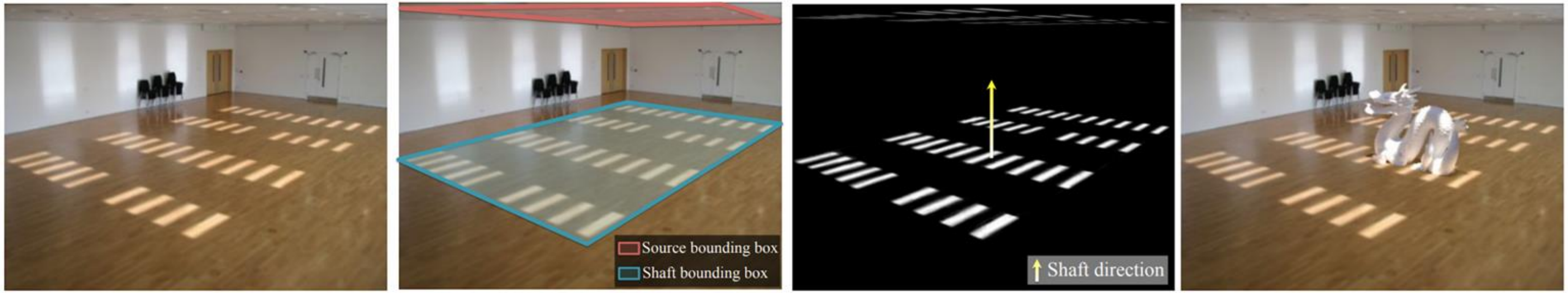
- Physical probe

https://www.pauldebevec.com/Probes/

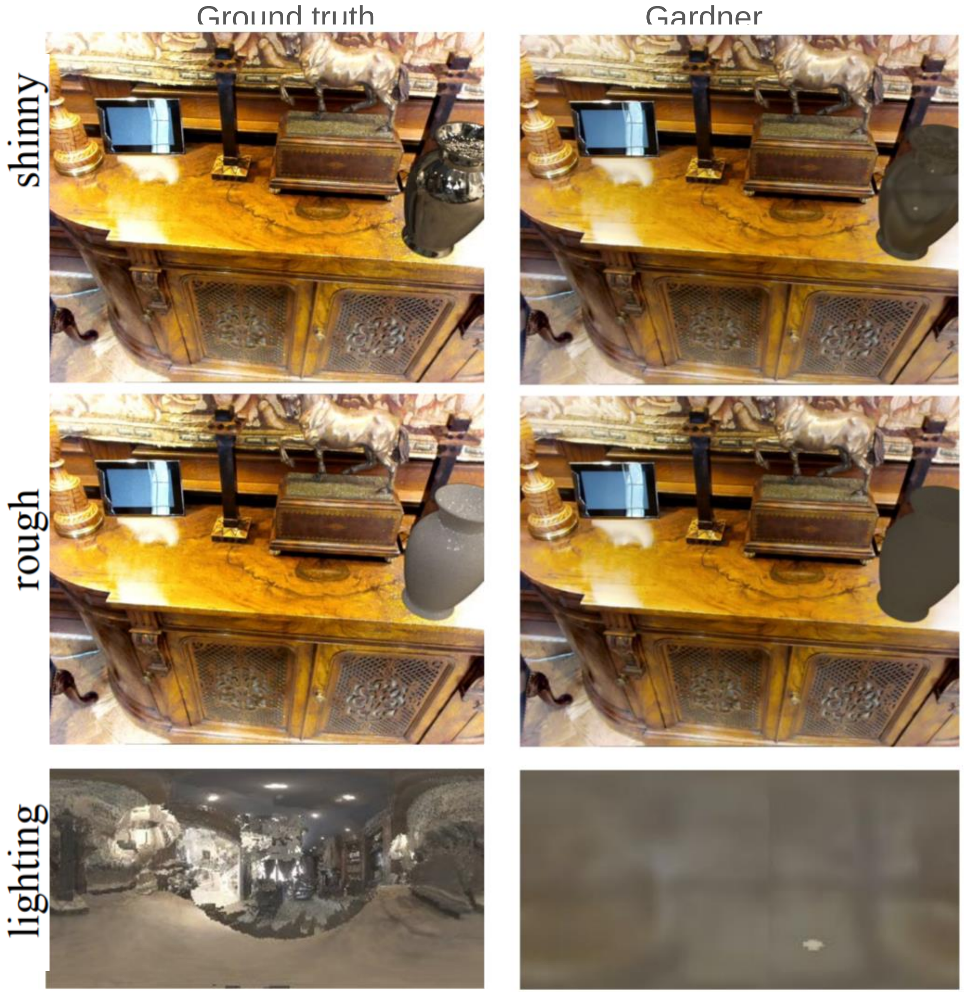
Related Work- Optimization Based Methods

“Rendering synthetic objects into Legacy Photographs”



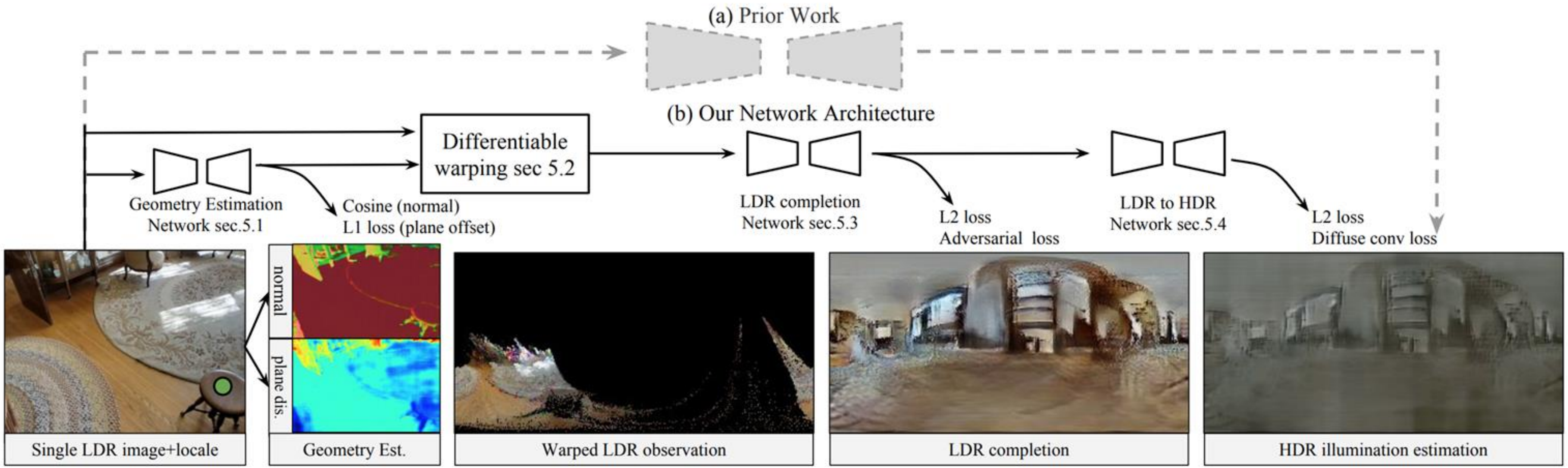


Related Work- Learning Based Methods



Problem Formulation

- 3 network method:
  - A geometry estimation network (via depth estimation) (this creates the warped image centered at chosen point)
  - An LDR completion map network (via an understanding of scene illumination and geometry)
  - LDR to HDR network (for improved accuracy)



Training Dataset Generation

Physical Probes → Costly and time consuming

Panoramic Datasets: illumination data only at **point of capture** → Limits data quantity

The authors leverage a RGB-D data sets (Matterport3D) to generate ground truth for any locale in the dataset!

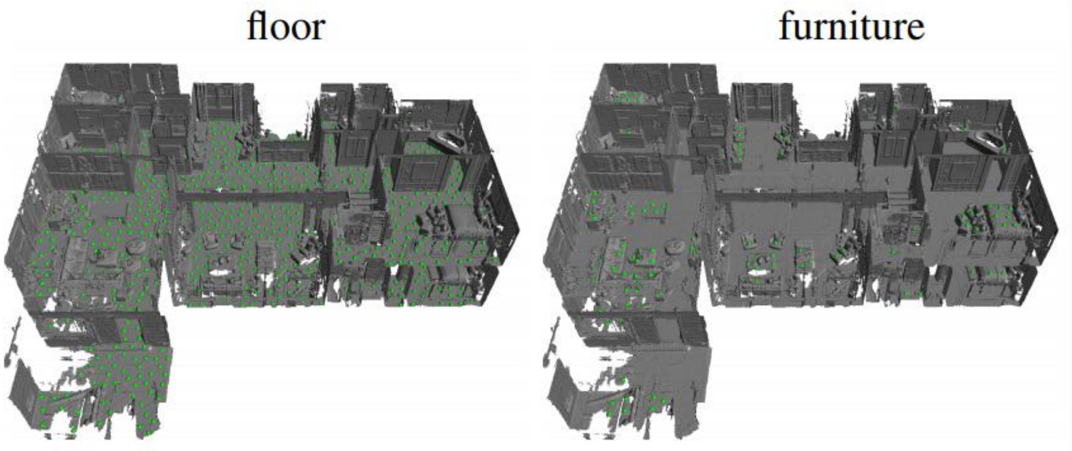
Training Dataset Generation

- Matterport3D contains panoramas composed of many densely acquired images
- Illumination maps can be generated at any locale by warping and compositing nearby panoramas

(a) Target locales (b) Resample "nearby" panoramas (c) Observations (d) Combined pano as illumination groundtruth

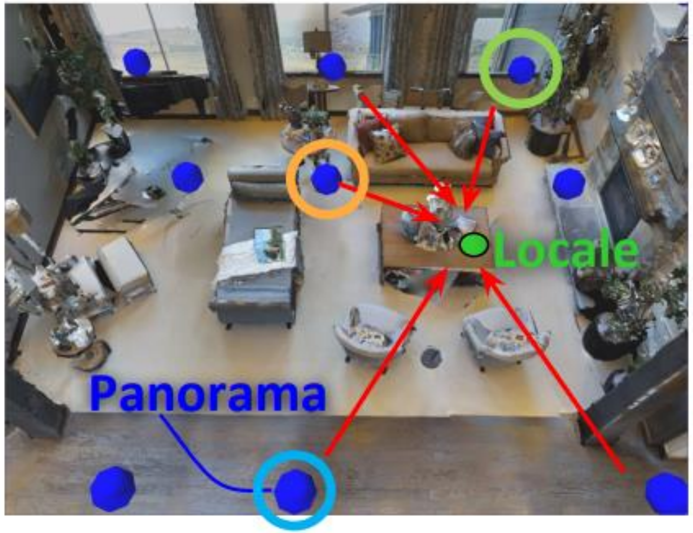
Training Dataset Generation: Selective Locales

- An application is virtual object placement, so locales are chosen according to where a “real” virtual object might logically be placed
  - Densely sample 10 cm above surface mesh
  - Criteria: horizontal surface (n<cos(π/8), semantic label “floor” or “furniture”, 10 cm object clearance, 50 cm minimum distance from previous locale



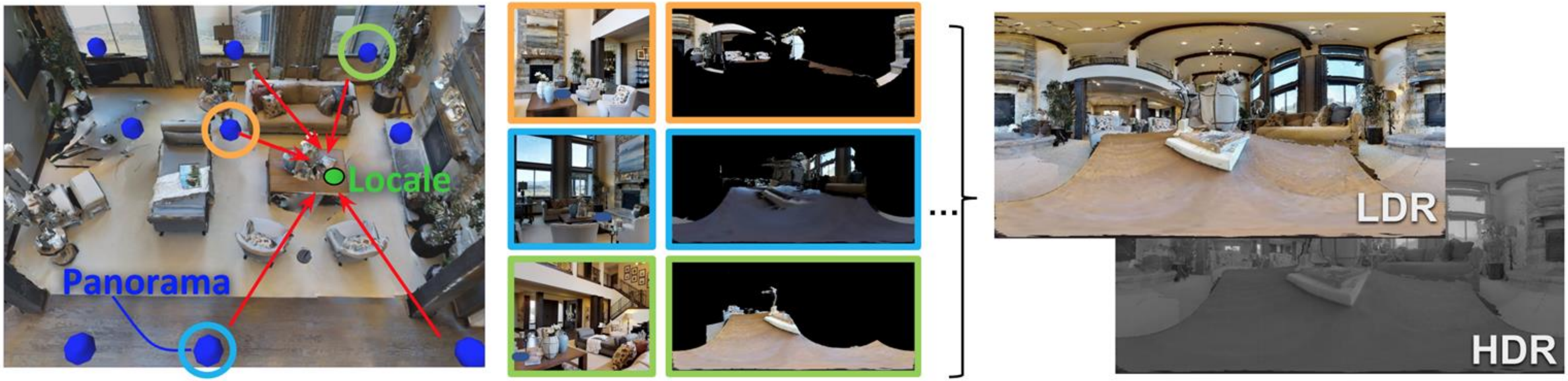
Training Dataset Generation: Forward Mapping

- For each locale, the distance to the closest surface in every direction is estimated
  - This is done using a forward mapping of every image in the panorama to the locale



Training Dataset Generation: Reverse Mapping

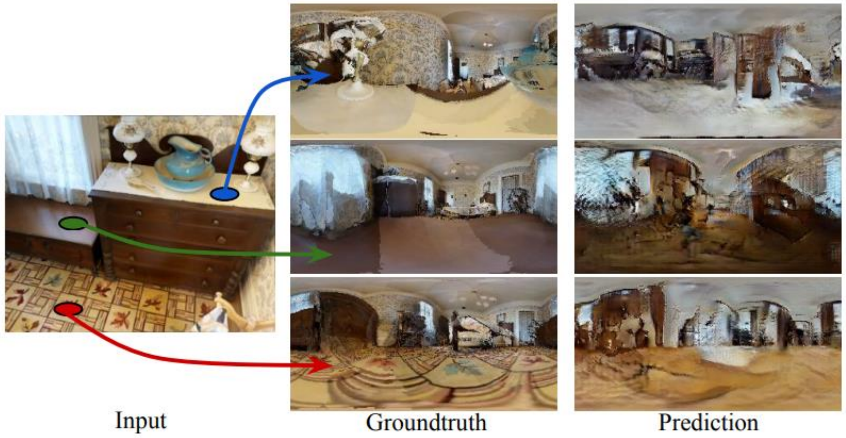
- Reconstruct illumination map by resampling input images via reverse mapping
  - Sample pixel values are blended proportionally to their distance from the locale



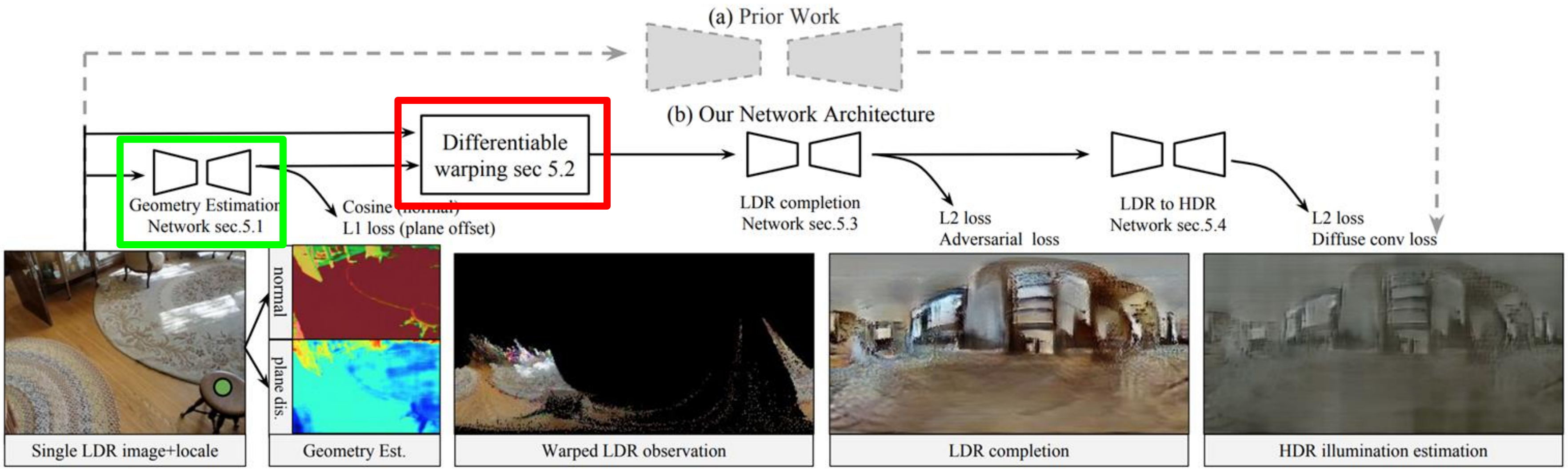
Training Dataset Generation: Advantages

1. Large variety of sampling sources gives varying illumination environments

1. Multiple illumination maps are generated for a single input image
    a. Model learns spatial dependencies between pixel selections and generated illumination maps

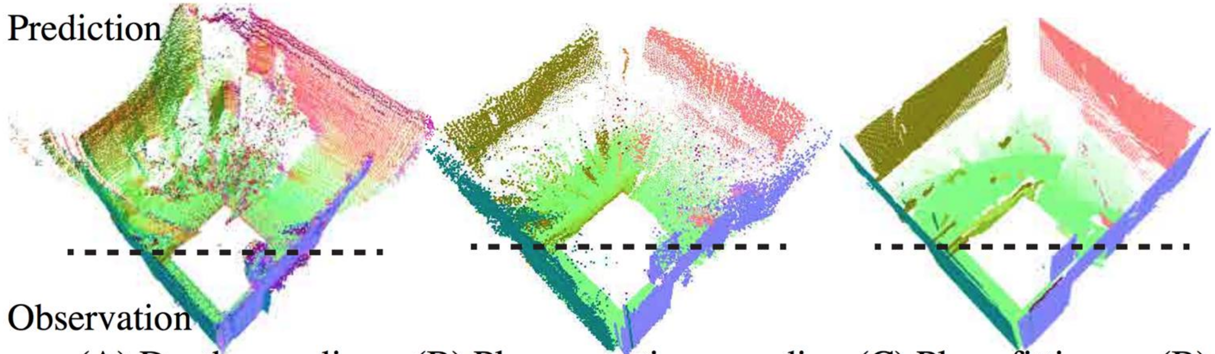


Network Architecture



Geometry Estimation

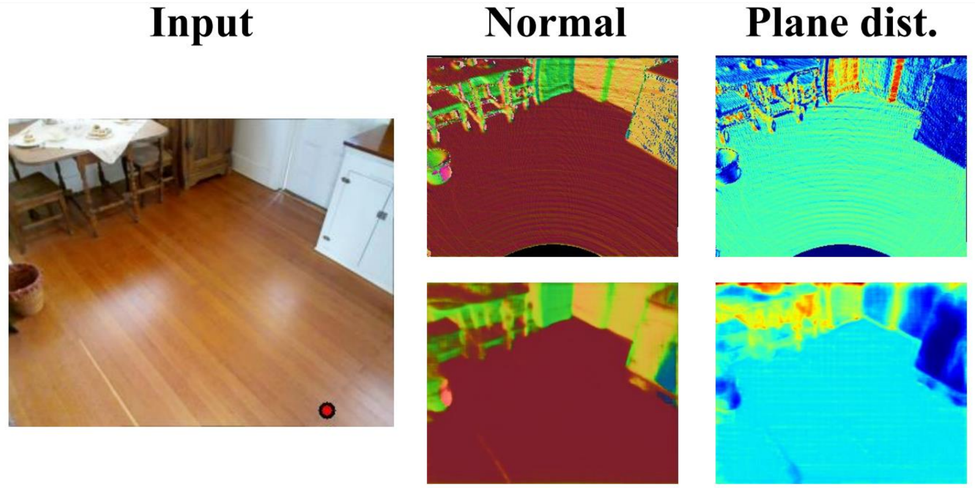
- This module generates a pixel-wise prediction of geometry represented as a plane equation: $aX + bY + cZ = d$
- Well suited for representing the large planar surfaces of indoor environments compared with raw depth values



(A) Depth encoding (B) Plane equation encoding (C) Plane fitting on (B)

Geometry Estimation: U-Net Model

- Color image as input
- Surface normal and distance-to-origin plane distance as supervision
  - Calculated directly from Matterport3D depth images

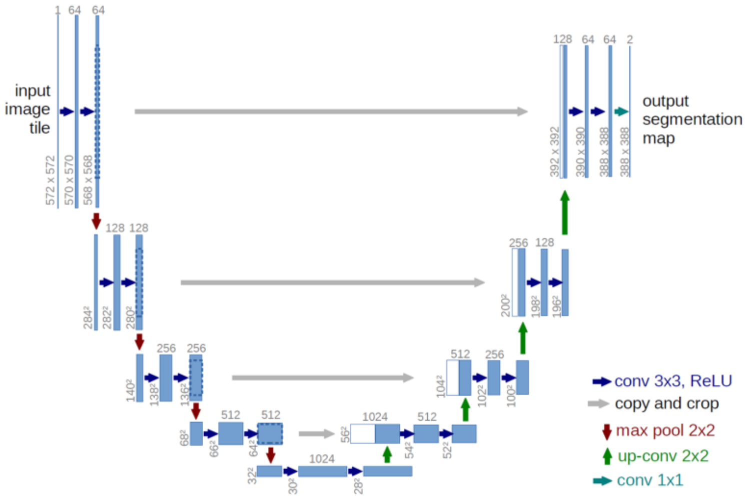


Ground Truth

Model Output

Geometry Estimation: U-Net Model

- Surface normal predictions via a cosine loss
  - Angle between predicted and GT normals
- Plane offset predictions via an l1 loss
  - Difference between predicted and GT plane distance



Geometry Estimation: U-Net Model - PN Layer

- Output from the U-NET is passed to an additional PN layer that converts the normal and plane distances into pixel-wise prediction of 3D locations (via plane equation)
- This layer is fully differentiable and can be trained via an **l1 loss**
- Enforces consistency between the normal and plane distance outputs
  - Reduces noise seen when reconstructing 3D surfaces

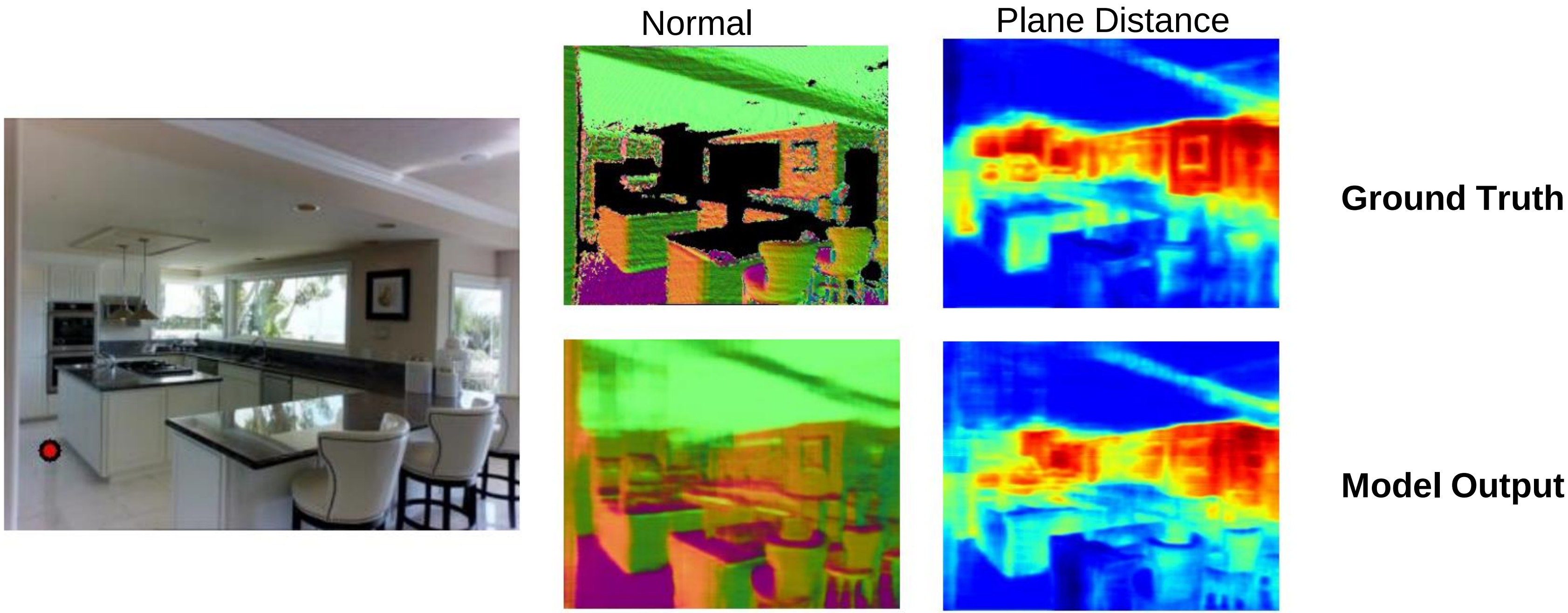
Camera intrinsics:

- *f* = F/p where F is focal length and p is real pixel size
- *c* is the optical center

$$\vec{P} = (x, y, z)$$

$$\vec{P} = -\frac{p}{\vec{v}\cdot\vec{n}}\vec{v}, \text{ where } \vec{v} = \left(\frac{x_i - c_x}{f_x}, \frac{y_i - c_y}{f_y}, 1\right)$$

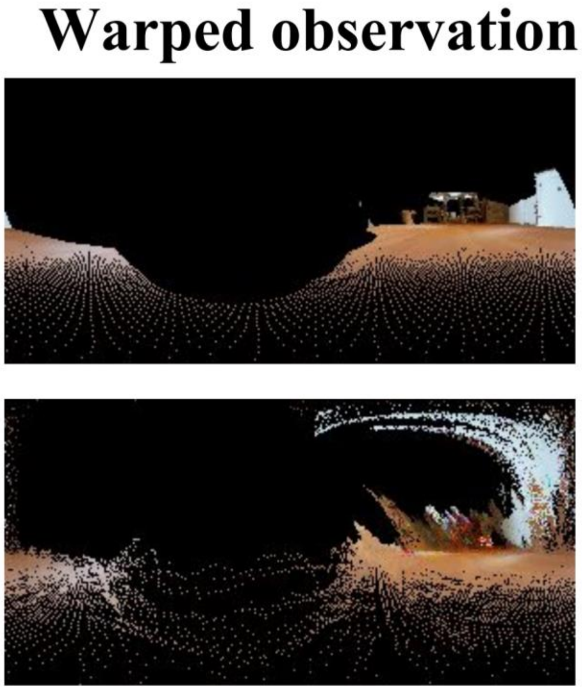
Geometry Estimation: Examples



Geometry-Aware Warping: Single Layer Module

- This maps the input image pixels to a spherical panoramic image, *h(φ, θ),* of the light arriving at *l*
- Pixels without a projected value are set to -1

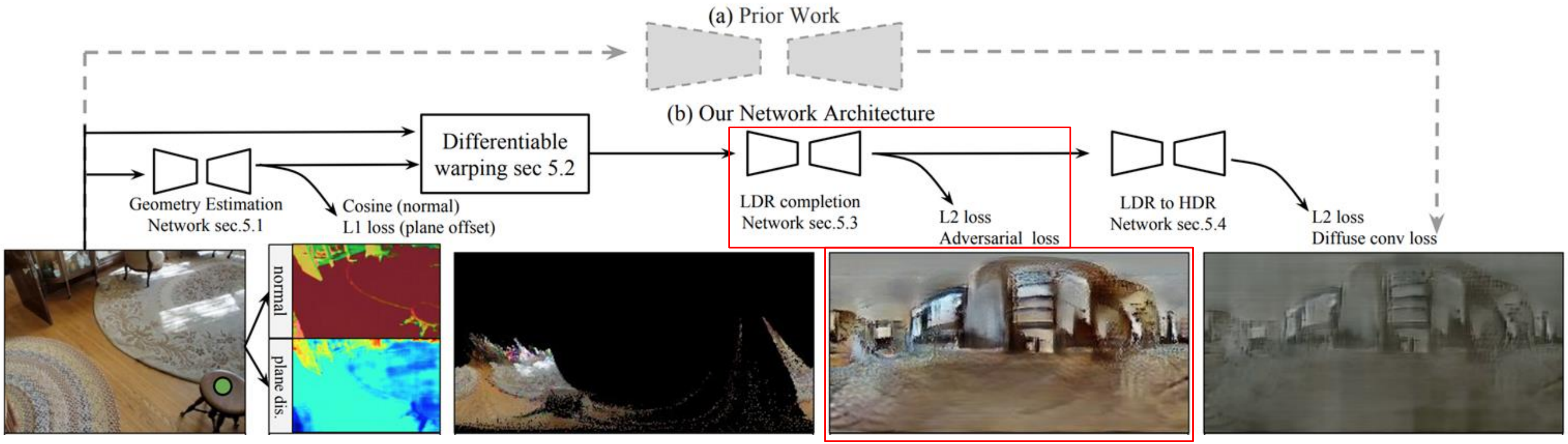
l, the chosen point



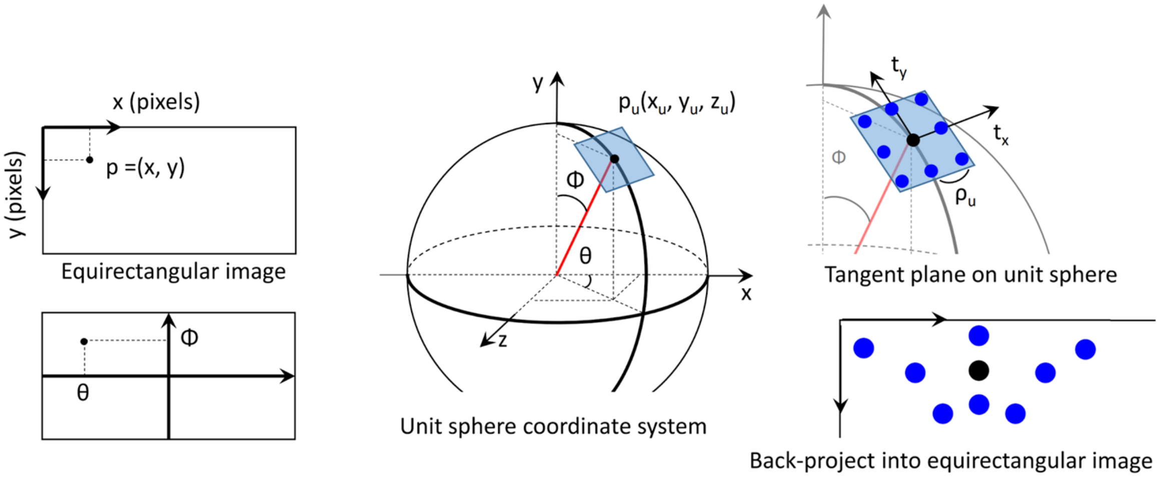
Top: Ground Truth

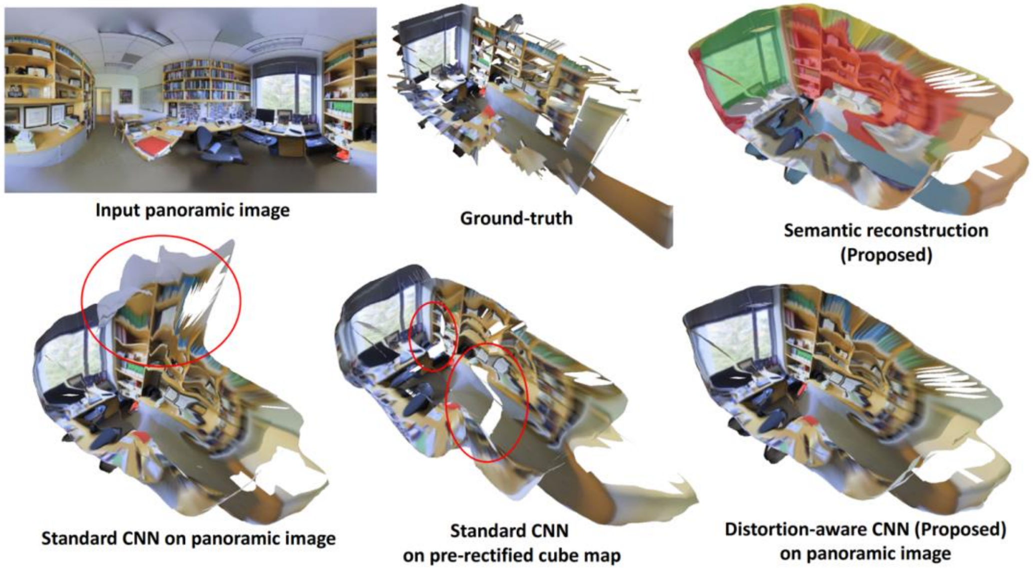
Step 2: LDR Panorama Completion

- 2nd module of this system
- Fully Convolutional ResNet50
- Input: mapped observed pixels
- Outputs: dense pixel wise prediction of illumination



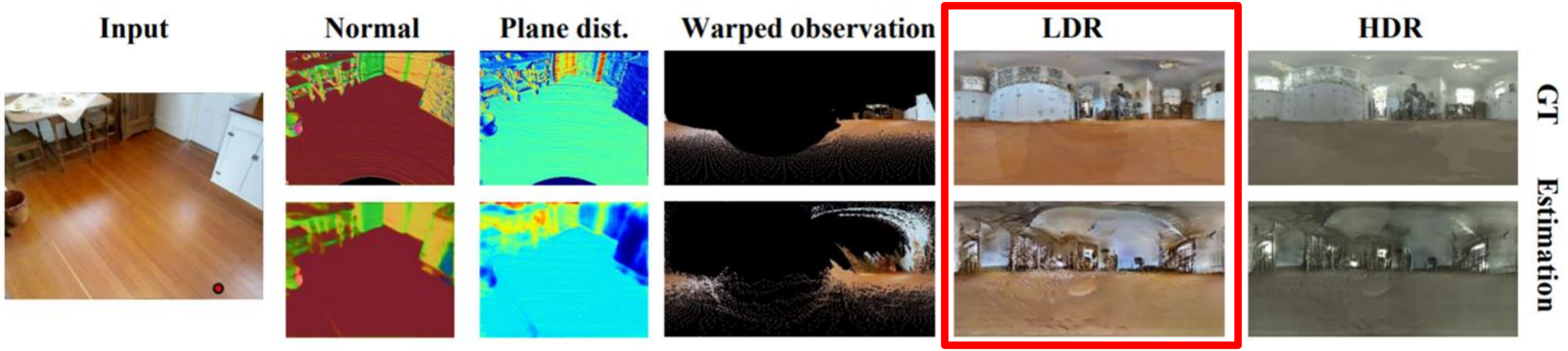
Distortion Aware Convolutional Filters



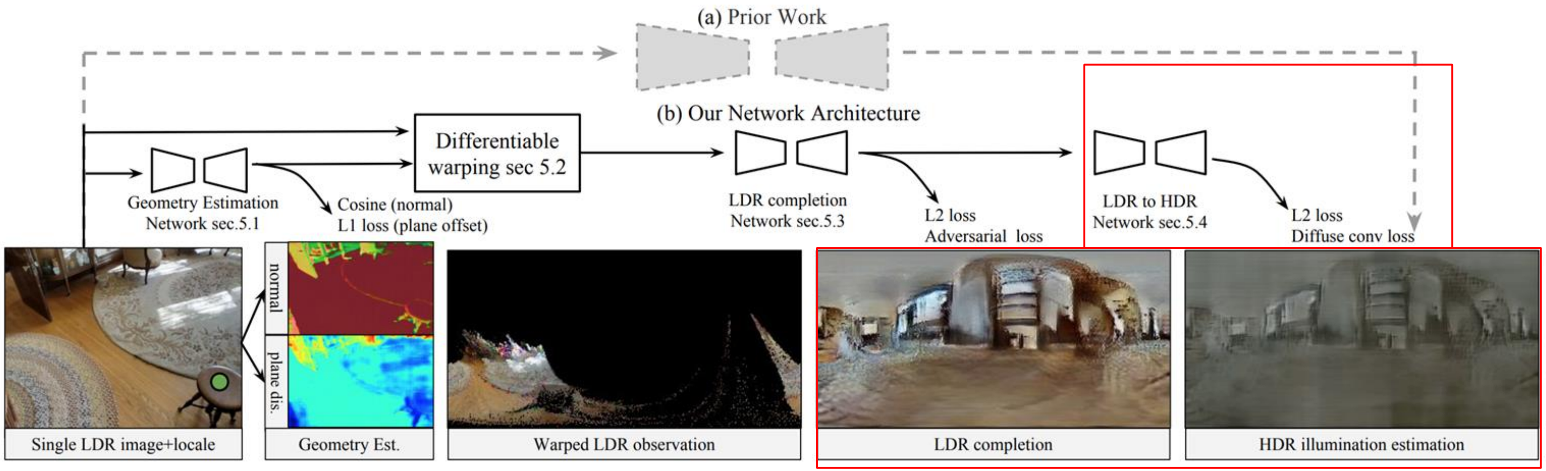


LDR Panorama Completion

- One of the biggest challenges: multi-model nature of the problem
- To address this: along with pixel wise supervision the module is trained with adversarial loss using a discriminator network

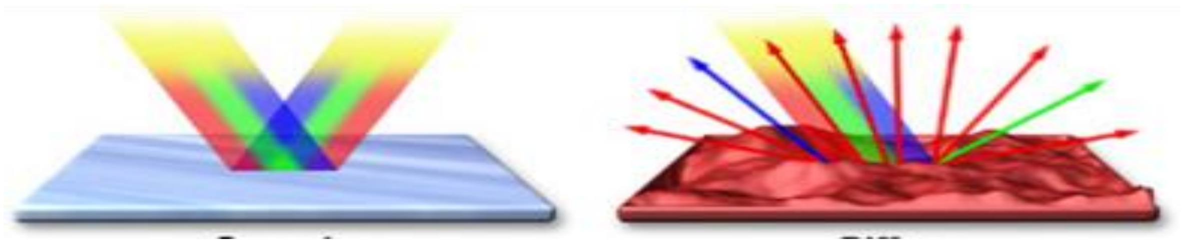


Step 3: LDR-to-HDR Estimation



- This module takes predicted LDR illumination as input and outputs a dense pixel-wise prediction of HDR illumination intensities.

LDR-to-HDR Estimation (Cont ..)



Specular Surface Diffuse Surface

- The LDR-to-HDR module learns the mapping function for all pixels from the LDR space to the HDR space. The module is trained with supervision from: 1) a pixel-wise l_2 loss and 2) a diffuse convolutional loss L.

LDR-to-HDR Estimation (Cont ..)

1. Pixel-wise l_2 loss measures the visual error when re-lighting a perfectly specular surface.

$$L_{\ell 2} = \frac{1}{N}\sum_{i=1}^{N}(J(i) - J^*(i))$$

$$H(i) = \begin{cases} J(i) * 65536 * 8e^{-8}, & J(i) \leq 3000 \\ 2.4e^{-4} * 1.0002^{(J(i)*65536-3000)}, & J(i) > 3000 \end{cases}$$

Notations:
J: log-scaled image of the final light intensity.

J*: log-scaled ground truth image of the final light intensity.

H: This is the output HDR illumination map.

i:Target local or specified pixel in an image

LDR-to-HDR Estimation (Cont ..)

2. Diffuse convolutional loss measures the visual error when re-lighting a perfectly diffuse surface.

$$L_d = \frac{1}{N}\sum_{i=1}^{N}\left(D(H(i)) - D(H^*(i))\right)$$

$$D(H,i) = \frac{1}{K_i}\sum_{\omega\in\Omega_i} H(\omega)s(\omega)(\omega\cdot\vec{n_i})$$

H: Expected HDR illumination map produced by LDR-to-HDR module.

H*: Ground truth HDR illumination map

D: Diffuse Convolution function.

L_d: Diffuse convolution loss.

Ω_i: hemisphere centered at pixel i.

K_i: the sum of solid angles on Ω_i.

$\vec{n}$:the unit normal at pixel i

s(ω): the solid angle for the pixel in the direction ω

LDR-to-HDR Estimation (Cont ..)

- Add diffuse convolution loss and pixel-wise l_2 loss to compute final loss:

$$L = \lambda_1 L_{\ell 2} + \lambda_2 L_d$$

where,

$$\lambda_1 = 0.1 \text{ and } \lambda_2 = 0.05.$$

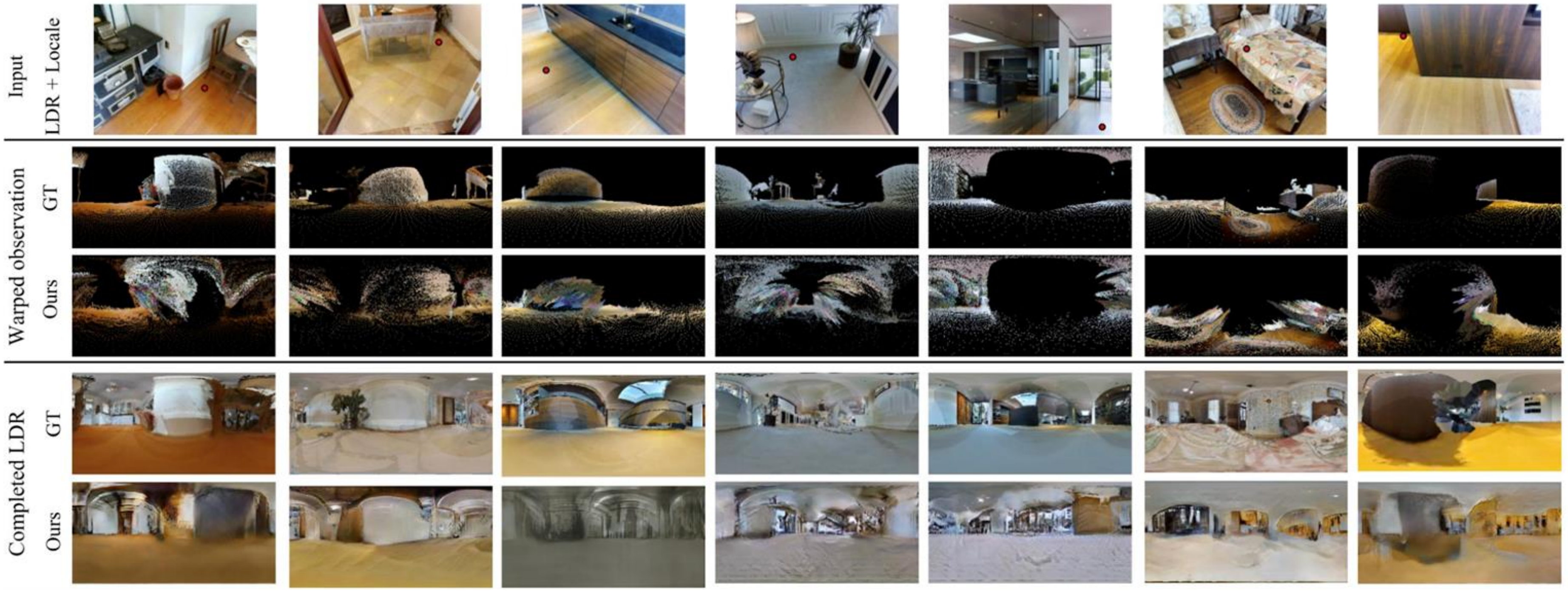
Evaluation:

- Matterport3D dataset of HDR RGB-D is leveraged to generate the training data for the arbitrary locale.

- Training and testing is done by using same train/test split provided in Matterport3D dataset.

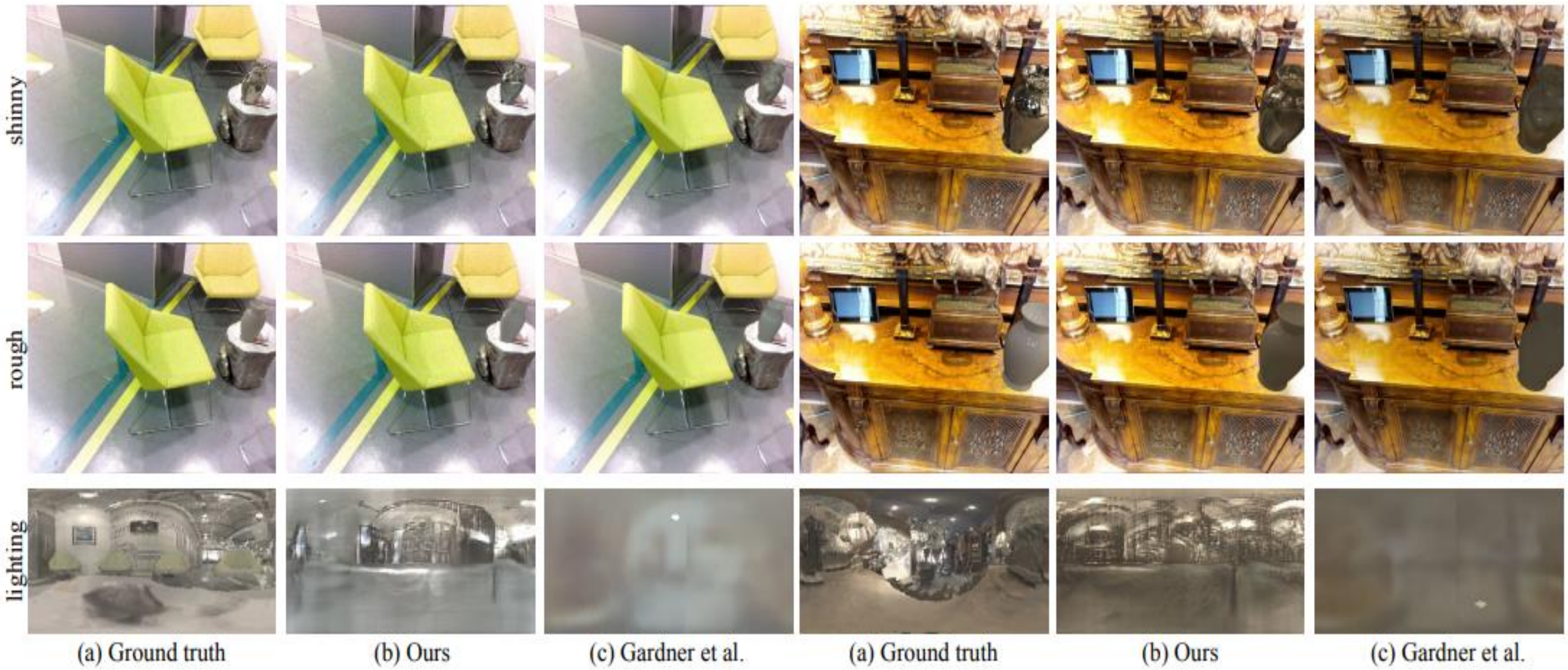
- The experiment makes quantitative and qualitative comparisons with the models proposed in the prior work.

Comparisons to state-of-the-art



Comparisons to state-of-the-art

Comparisons to state-of-the-art



Evaluation Metrics:

- **Pixel-wise l_2 distance error:** Sum of all the pixel-wise l_2 distances between the predicted H_l and the ground truth H_l^* illumination maps.

- **Pixel-wise diffuse convolution error:** Sum of all the pixel-wise l_2 distance between $D(H_l)$ and $D(H_l^*)$.

Comparisons to state-of-the-art

Method	$\ell 2$(log)	$\ell 2$	diffuse
Gardner *et al.* [7]	0.375	0.977	1.706
Im2Im network	0.229	0.369	0.927
Nearest Neighbour	0.296	0.647	1.679
Ours	**0.202**	**0.280**	**0.772**

Table 1. Comparing the quantitative performance of our method to that of Gardner *et al*. [7] and a nearest neighbour retrieval method.

Modularization v.s. Additional supervision:

	$\ell 2(\log)$	$\ell 2$	diffuse
without	0.213	0.319	0.856
with (ours)	**0.202**	**0.280**	**0.772**

Table 2. Effects of modularization.

Comparisons to variants:

LDR + D $\longrightarrow$ HDR (first two modules are omitted)

HDR(wrapped) + D $\longrightarrow$ HDR (last modules are omitted)

Method	$\ell 2$(log)	$\ell 2$	diffuse
LDR→HDR	0.202	0.280	0.772
LDR+D→HDR	0.188	0.269	0.761
HDR+D→HDR	**0.131**	**0.212**	**0.619**

Table 3. Comparisons to variants with oracles.

Effect of different losses:

l2 only | l2+gan | l2+gan+df | Ground truth

loss	$\ell 2$(log)	$\ell 2$	diffuse
l2	**0.116**	0.235	0.691
l2+gan	0.224	0.275	0.713
l2+gan+df	0.131	**0.212**	**0.619**

Table 4. Effects of different losses.

Strengths and Weaknesses

Strengths:

- This model is separated into 3 separate modules which increases performance (3 more doable subtasks rather than one larger problem)
- Produces richer/sharper detailed estimations

Weaknesses:

- Produces plausible illumination maps rather than accurate ones when no lights are observed directly in the input

Extensions

- Future work:
  - Include explicit modeling of surface material and reflective properties
  - explore alternative 3D geometric representations that facilitate out-of-view illumination estimation through whole scene understanding.

Thank You For Listening!