Single view depth, normal, and boundaries

3D Vision

University of Illinois

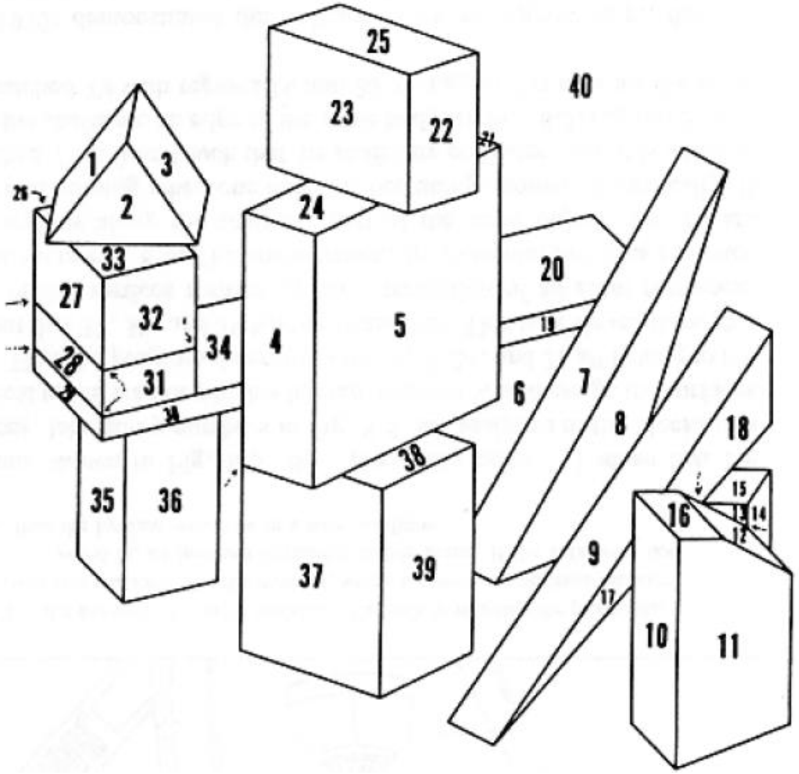
Derek Hoiem

Agenda

- Early computer vision representations and machine learning approaches

- Deep machine learning approaches
  - Depth
  - Normals
  - Boundaries

Early goals of computer vision



[Guzman 1968]

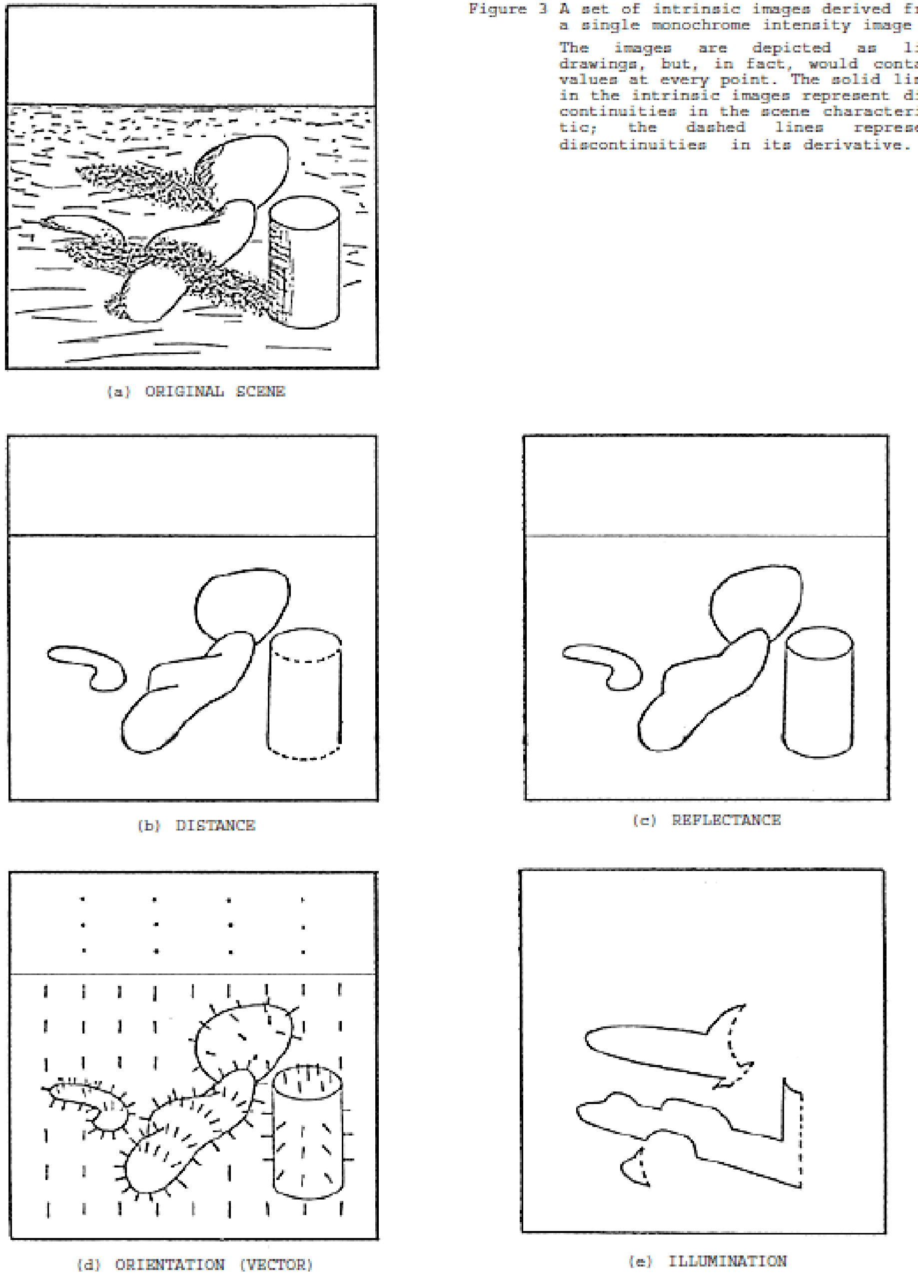
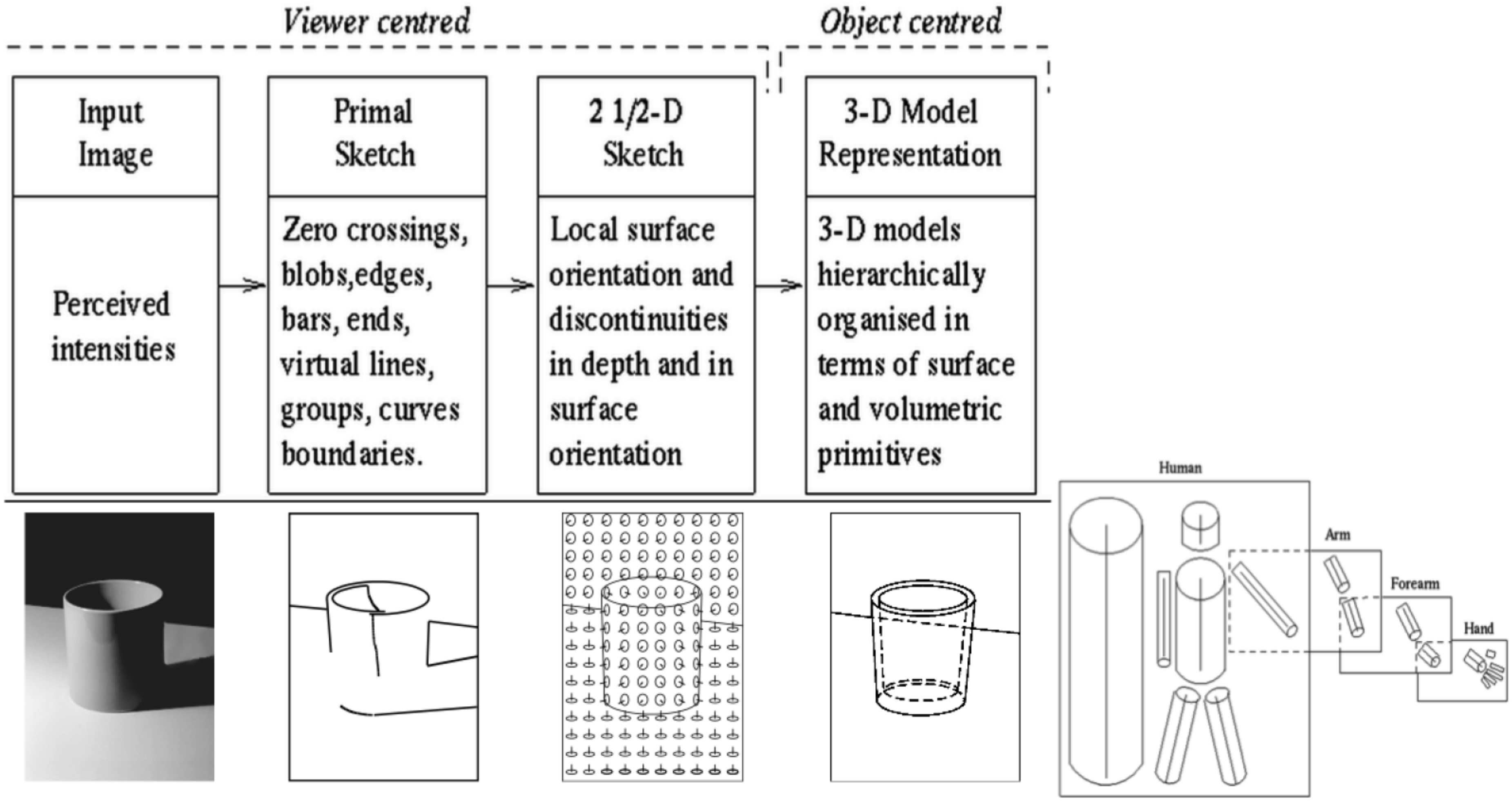


Figure 3 A set of intrinsic images derived from a single monochrome intensity image

The images are depicted as line drawings, but, in fact, would contain values at every point. The solid lines in the intrinsic images represent discontinuities in the scene characteristic; the dashed lines represent discontinuities in its derivative.

[Barrow Tenenbaum 1978] (Intrinsic Images)

Early goals of computer vision

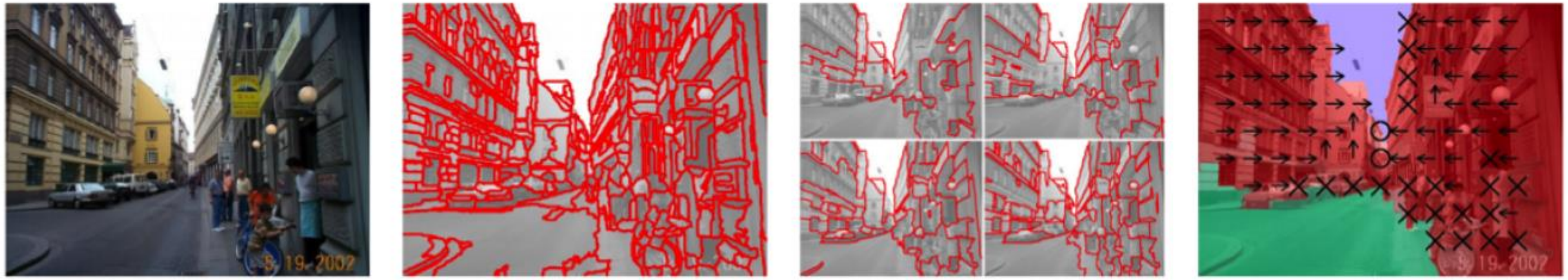


[Marr 1982] (Primal, 2 ½D sketch)

Early learning: Surface Normals

SURFACE CUES
Location and Shape
L1. Location: normalized x and y, mean
L2. Location: normalized x and y, 10^{th} and 90^{th} pctl
L3. Location: normalized y wrt estimated horizon, 10^{th}, 90^{th} pctl
L4. Location: whether segment is above, below, or straddles estimated horizon
L5. Shape: number of superpixels in segment
L6. Shape: normalized area in image
Color
C1. RGB values: mean
C2. HSV values: C1 in HSV space
C3. Hue: histogram (5 bins)
C4. Saturation: histogram (3 bins)
Texture
T1. LM filters: mean absolute response (15 filters)
T2. LM filters: histogram of maximum responses (15 bins)
Perspective
P1. Long Lines: (number of line pixels)/sqrt(area)
P2. Long Lines: percent of nearly parallel pairs of lines
P3. Line Intersections: histogram over 8 orientations, entropy
P4. Line Intersections: percent right of image center
P5. Line Intersections: percent above image center
P6. Line Intersections: percent far from image center at 8 orientations
P7. Line Intersections: percent very far from image center at 8 orientations
P8. Vanishing Points: (num line pixels with vertical VP membership)/sqrt(area)
P9. Vanishing Points: (num line pixels with horizontal VP membership)/sqrt(area)
P10. Vanishing Points: percent of total line pixels with vertical VP membership
P11. Vanishing Points: x-pos of horizontal VP - segment center (0 if none)
P12. Vanishing Points: y-pos of highest/lowest vertical VP wrt segment center
P13. Vanishing Points: segment bounds wrt horizontal VP
P14. Gradient: x, y center of mass of gradient magnitude wrt segment center

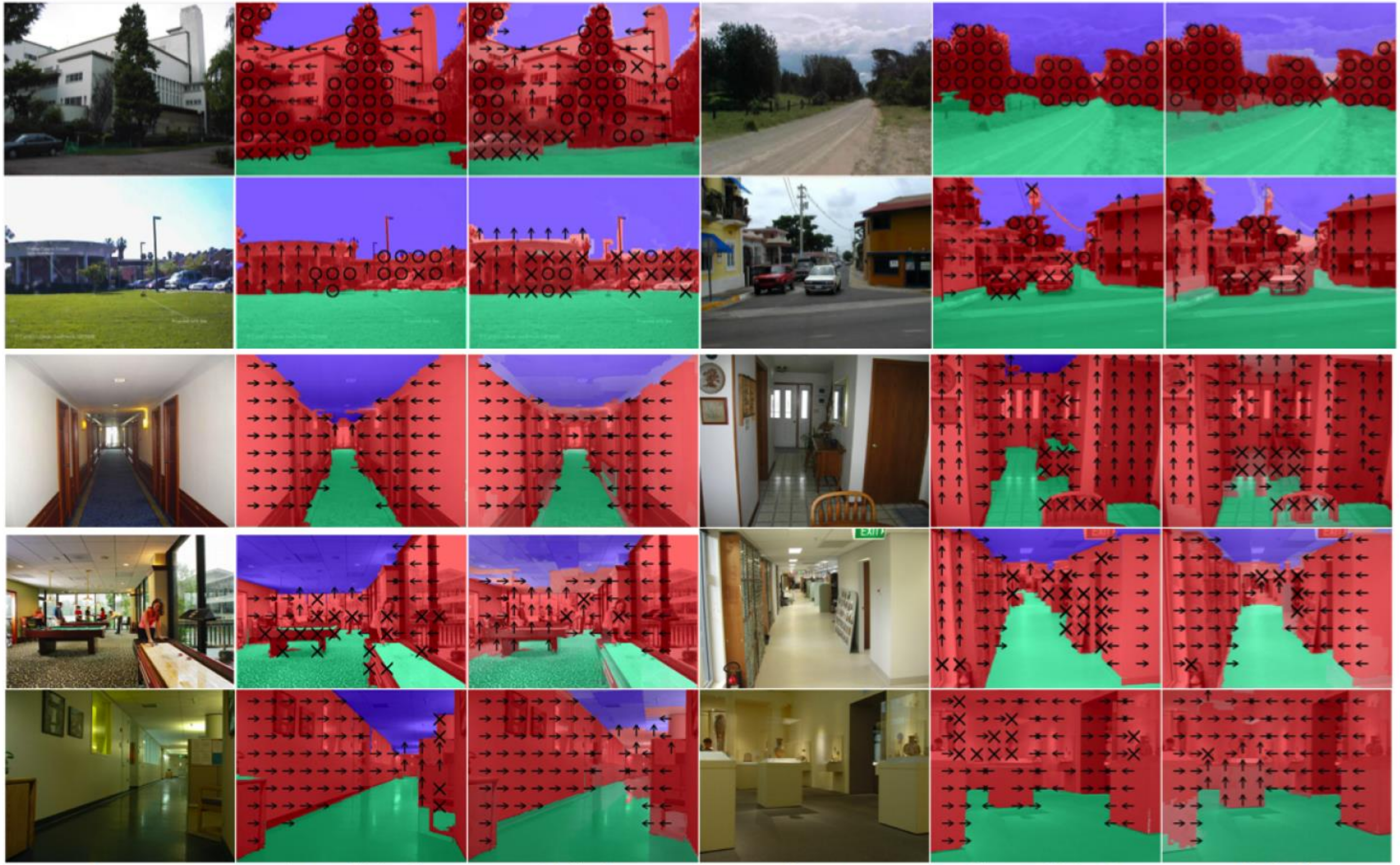
- Compute superpixels
- For each superpixel compute several interesting features that make use of vanishing points, color, texture, lines…
- Train classifiers to predict several geometric classes: support, vertical sky



Input | Superpixels | Multiple Segmentations | Surface Layout

Hoiem Efros Hebert 2005,2007

Slide from Sanja Fidler

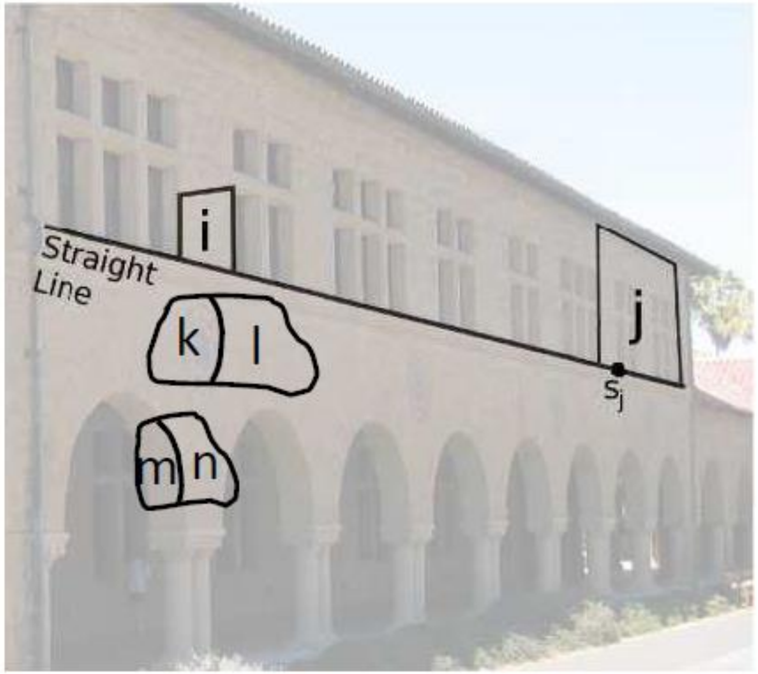


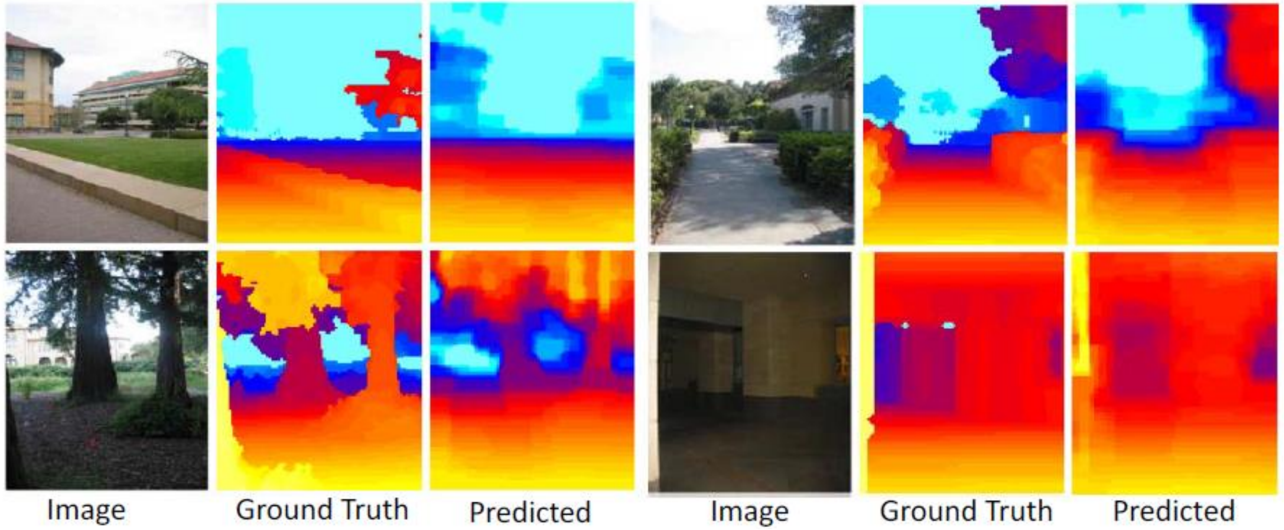
Input Ground Truth Labels Input Ground Truth Labels

Early learning: Depth

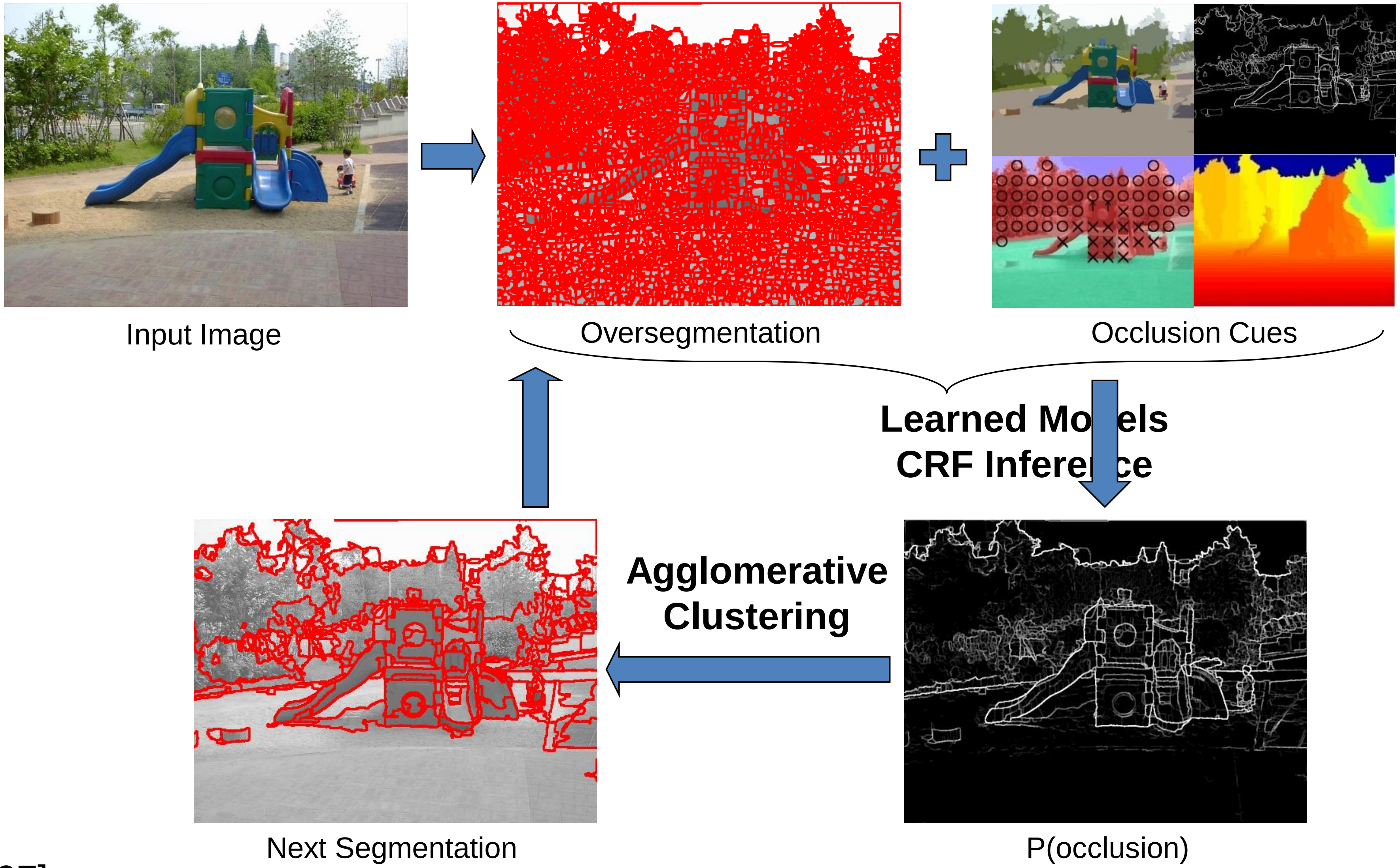
Make3D: Saxena et al. 2008

- Divide image into small regions
- Compute features for each superpixel
- Predict 3D plane parameters for each superpixel
- Compute confidence for each prediction
- Perform global inference with constraints: connectedness, coplanarity, colinearity



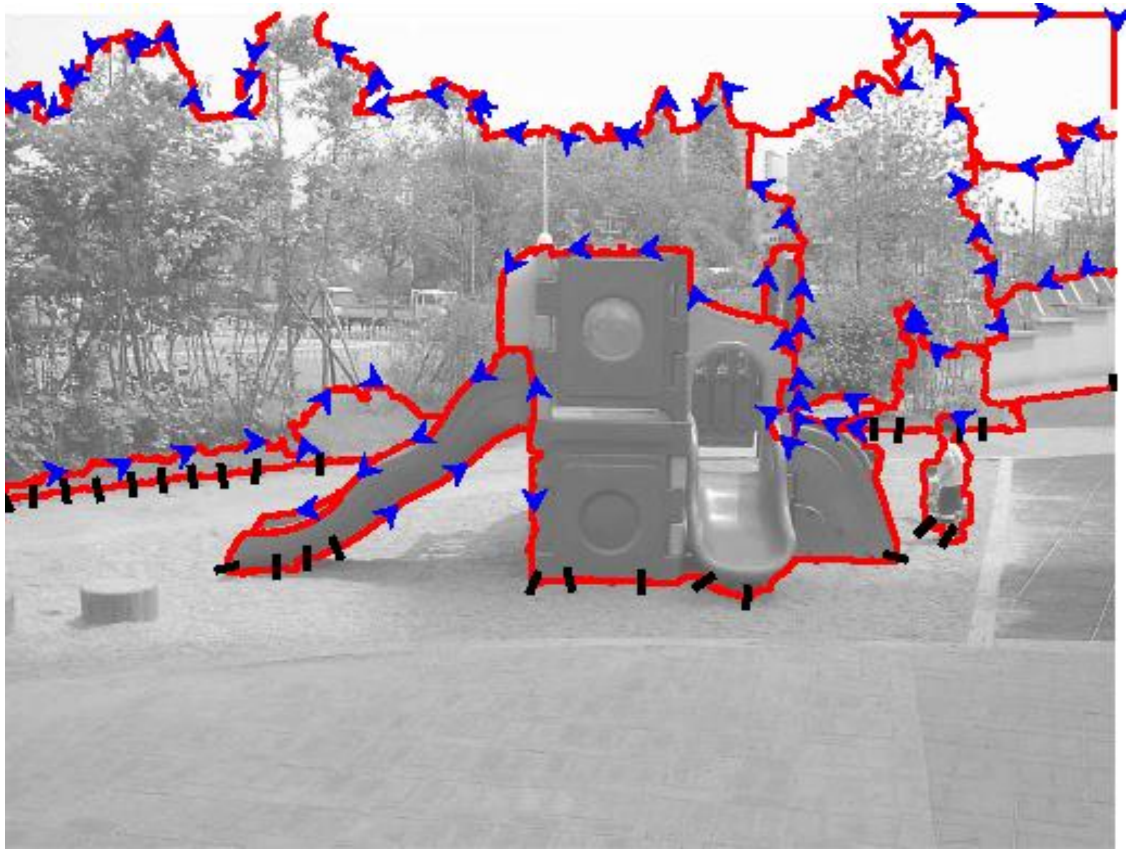


Early learning: occlusion boundaries

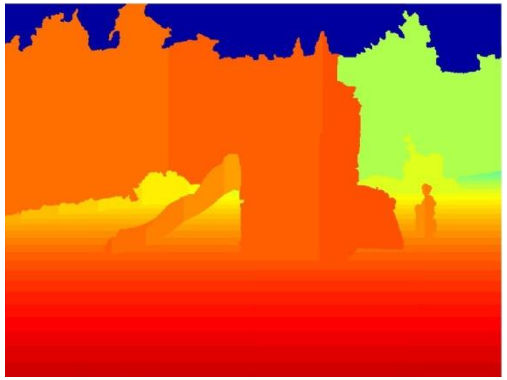


[Hoiem et al. 2007]

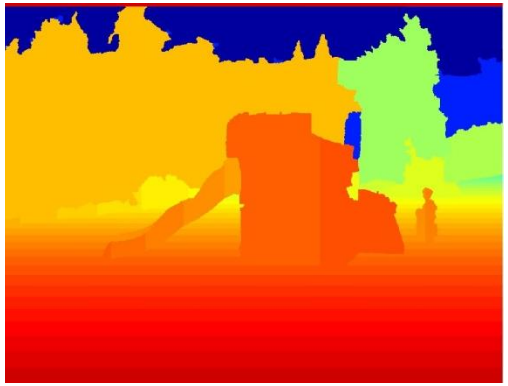
Early learning: occlusion boundaries



Boundaries, Foreground/Background, Contact

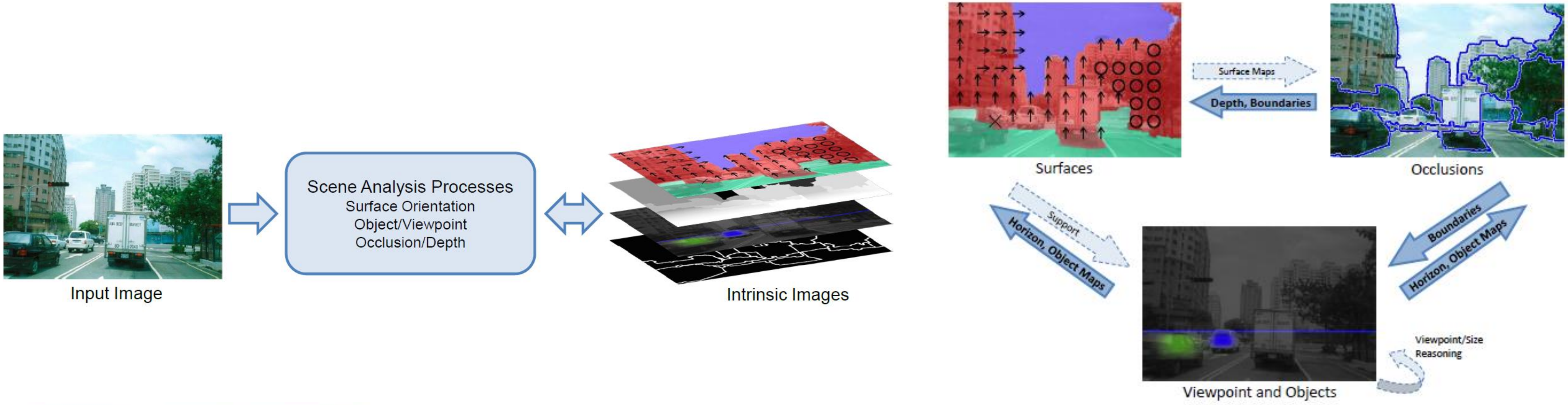


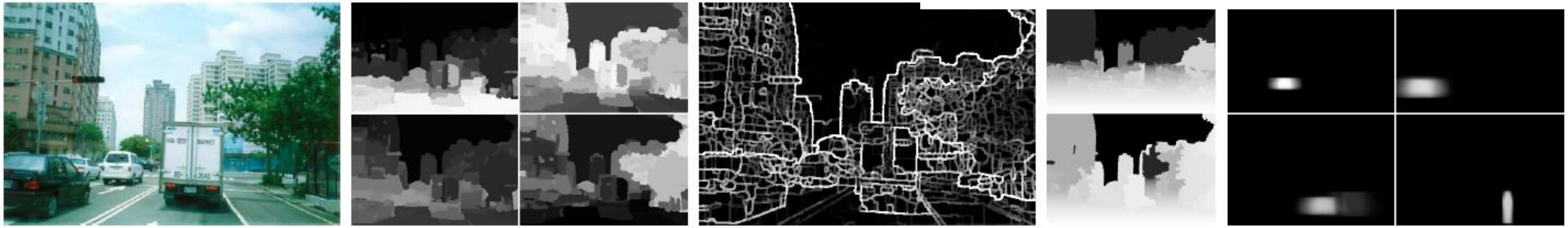
Depth (Min)



Depth (Max)

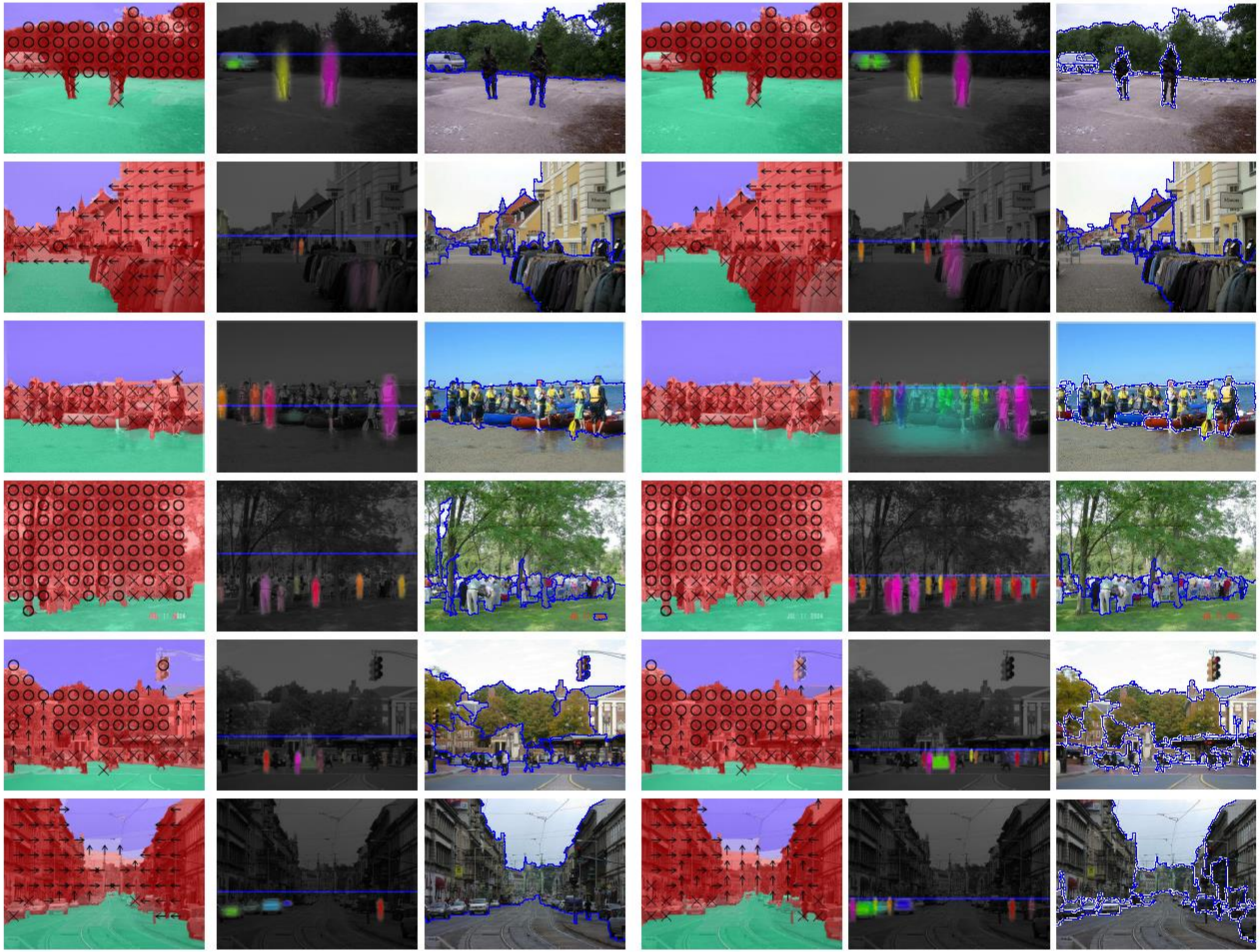
Early learning: 3D integration





(a) Input Image (b) Surfaces (c) Occlusion (d) Depth (e) Objects

[Hoiem et al. 2008]



Surfaces [**102**] Objects [**41**] Occlusions [**107**] Surfaces (joint) Objects (joint) Occlusions (joint)

Single view 3D is a subtle problem

- Depth
  - Humans are bad at absolute depth but can predict ordinal relationships and can function as if they know depth (throw, pick up etc.)
  - Depth informs more about distance than shape
  - Precision matters more for close objects
- Surface Normals
  - Humans are good at predicting normals
  - Normals describe shape
  - Normals are scale dependent
  - Precision matters more for close objects
- Occlusion Boundaries
  - Humans are good at predicting
  - Exterior boundaries tell us which things can move separately
  - Interior boundaries needed with normals to predict complex shapes

https://www.friendlyshade.com/product/medieval-brick-wall/

Deep learning for depth, surface, boundaries

Learning performance depends on classifier, optimization, loss, data – most recent work focuses on loss and data

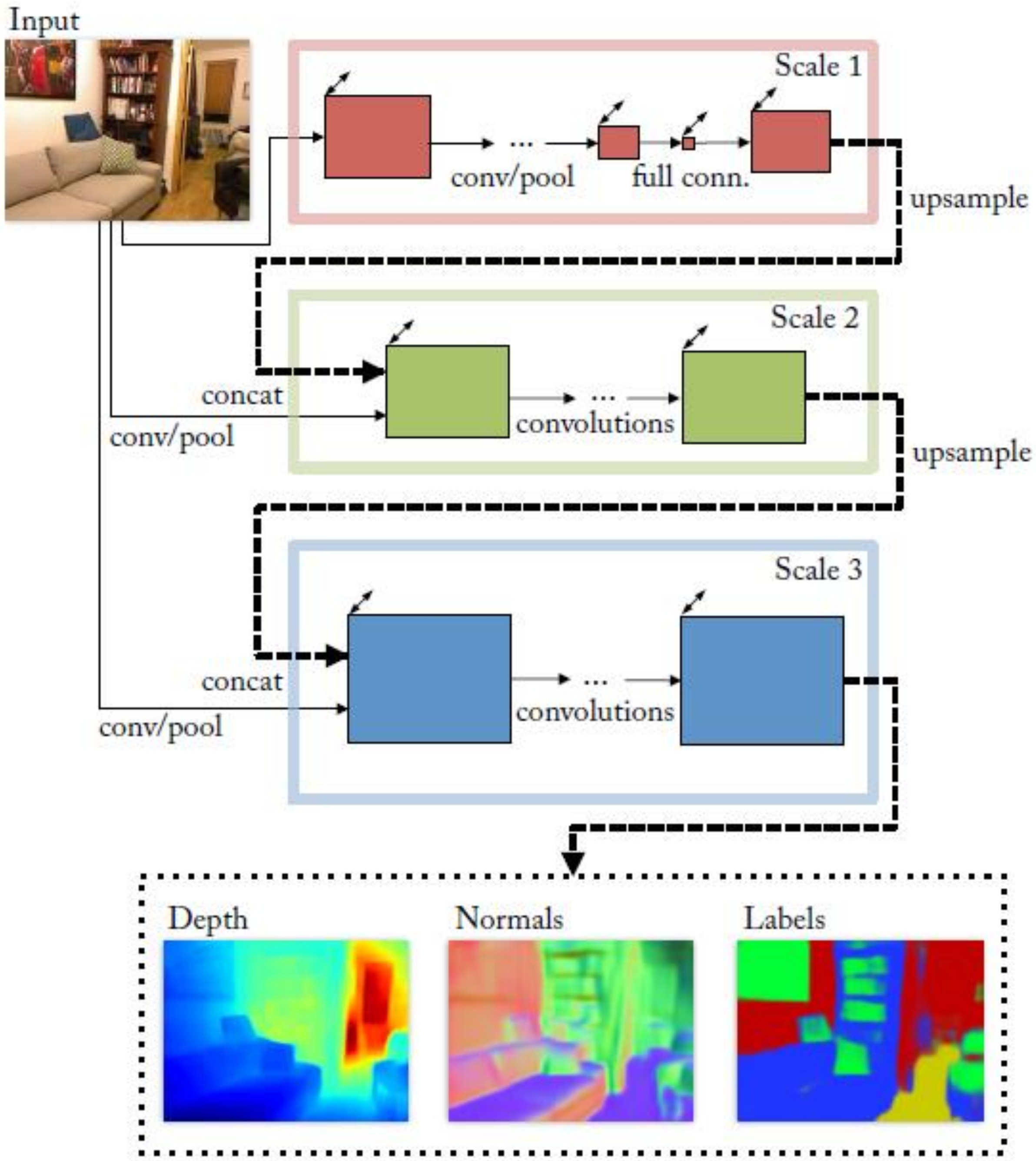
- Classifier form (architecture)
  - Most methods use something like a UNet
- **Loss**
  - Continuous objective, scale ambiguity
- **Data and augmentation**
  - Hard to get ground truth 3D data
- Optimization

Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture

ICCV 2015

David Eigen[1] Rob Fergus[1,2]
[1] Dept. of Computer Science, Courant Institute, New York University
[2] Facebook AI Research

- One architecture, 3 tasks: depth, normals, class labels
- Multiscale encoder, related to (contemporaneous) UNet
- Mostly weights are not shared between tasks (except depth/normal share scale 1)



Losses

https://arxiv.org/pdf/1406.2283.pdf

- Depth: scale-invariant log depth, gradient

D is log depth

$$d = D - D^*$$

$$L_{depth}(D, D^*) = \underbrace{\frac{1}{n}\sum_i d_i^2 - \frac{1}{2n^2}\left(\sum_i d_i\right)^2}_{\text{Scale-invariant squared log depth error (variance of log depth difference)}} + \underbrace{\frac{1}{n}\sum_i [(\nabla_x d_i)^2 + (\nabla_y d_i)^2]}_{\text{Squared gradient log depth error}}$$

- Normals: correlation

$$L_{normals}(N, N^*) = -\frac{1}{n}\sum_i N_i \cdot N_i^* = -\frac{1}{n} N \cdot N^*$$

- Class labels: cross-entropy

$$L_{semantic}(C, C^*) = -\frac{1}{n}\sum_i C_i^* \log(C_i)$$

Architecture / Training

- AlexNet and VGG backbones tested
- Optimize scales 1-2, then optimize scale 3
  - End-to-end would be done now
- Augmentation: scaling, in-plane rotation, translation, color, flips, contrast (w/ corresponding changes to depth/normal)

Results

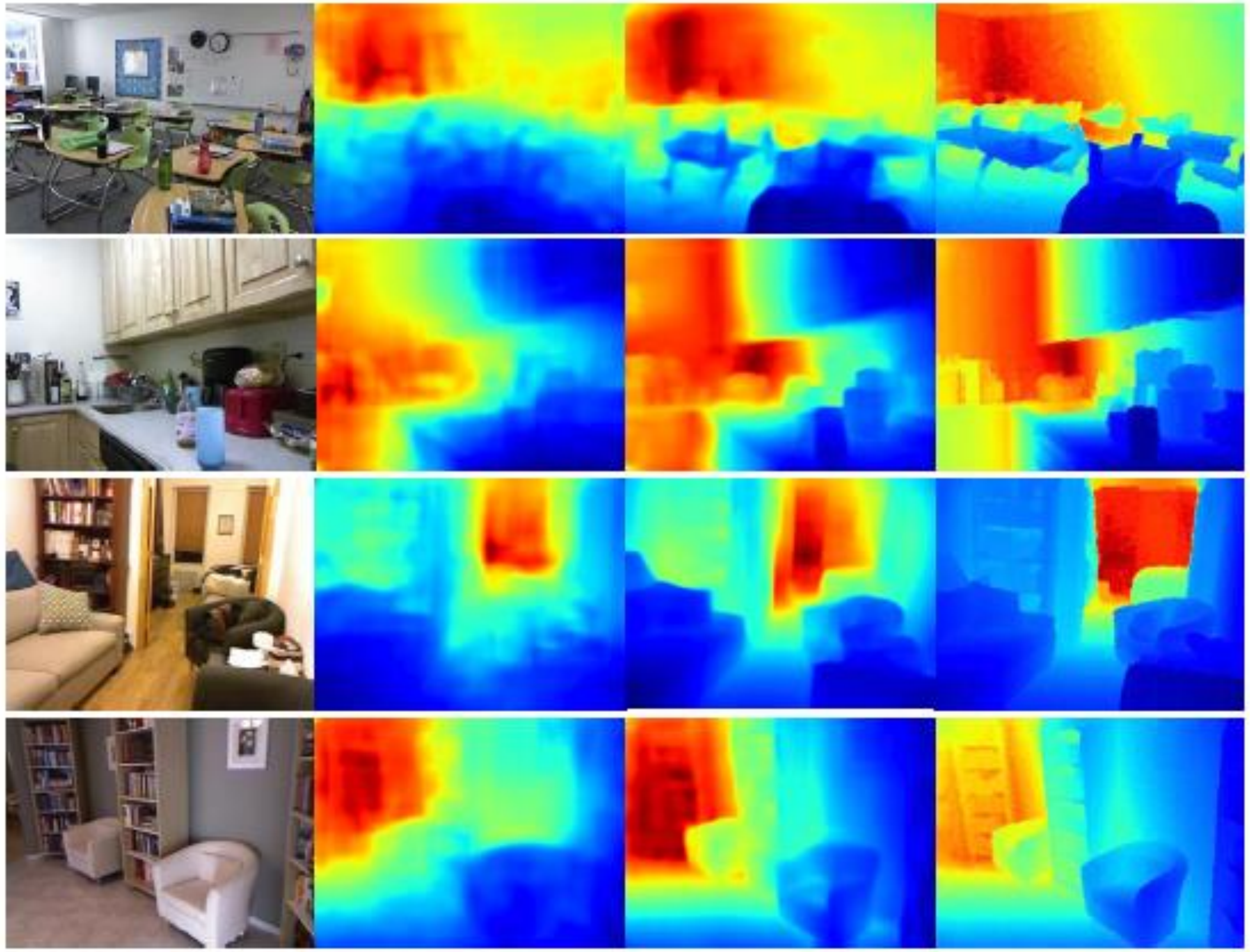


Figure 2. Example depth results. (a) RGB input; (b) result of [8]; (c) our result; (d) ground truth. Note the color range of each image is individually scaled.

NYU v2

Depth Prediction							
	Ladicky[20]	Karsch[18]	Baig [1]	Liu [23]	Eigen[8]	Ours(A)	Ours(VGG)
$\delta < 1.25$	0.542	–	0.597	0.614	0.614	0.697	**0.769**
$\delta < 1.25^2$	0.829	–	–	0.883	0.888	0.912	**0.950**
$\delta < 1.25^3$	0.940	–	–	0.971	0.972	0.977	**0.988**
abs rel	–	0.350	0.259	0.230	0.214	0.198	**0.158**
sqr rel	–	–	–	–	0.204	0.180	**0.121**
RMS(lin)	–	1.2	0.839	0.824	0.877	0.753	**0.641**
RMS(log)	–	–	–	–	0.283	0.255	**0.214**
sc-inv.	–	–	0.242	–	0.219	0.202	**0.171**

Table 1. Depth estimation measurements. Note higher is better for top rows of the table, while lower is better for the bottom section.

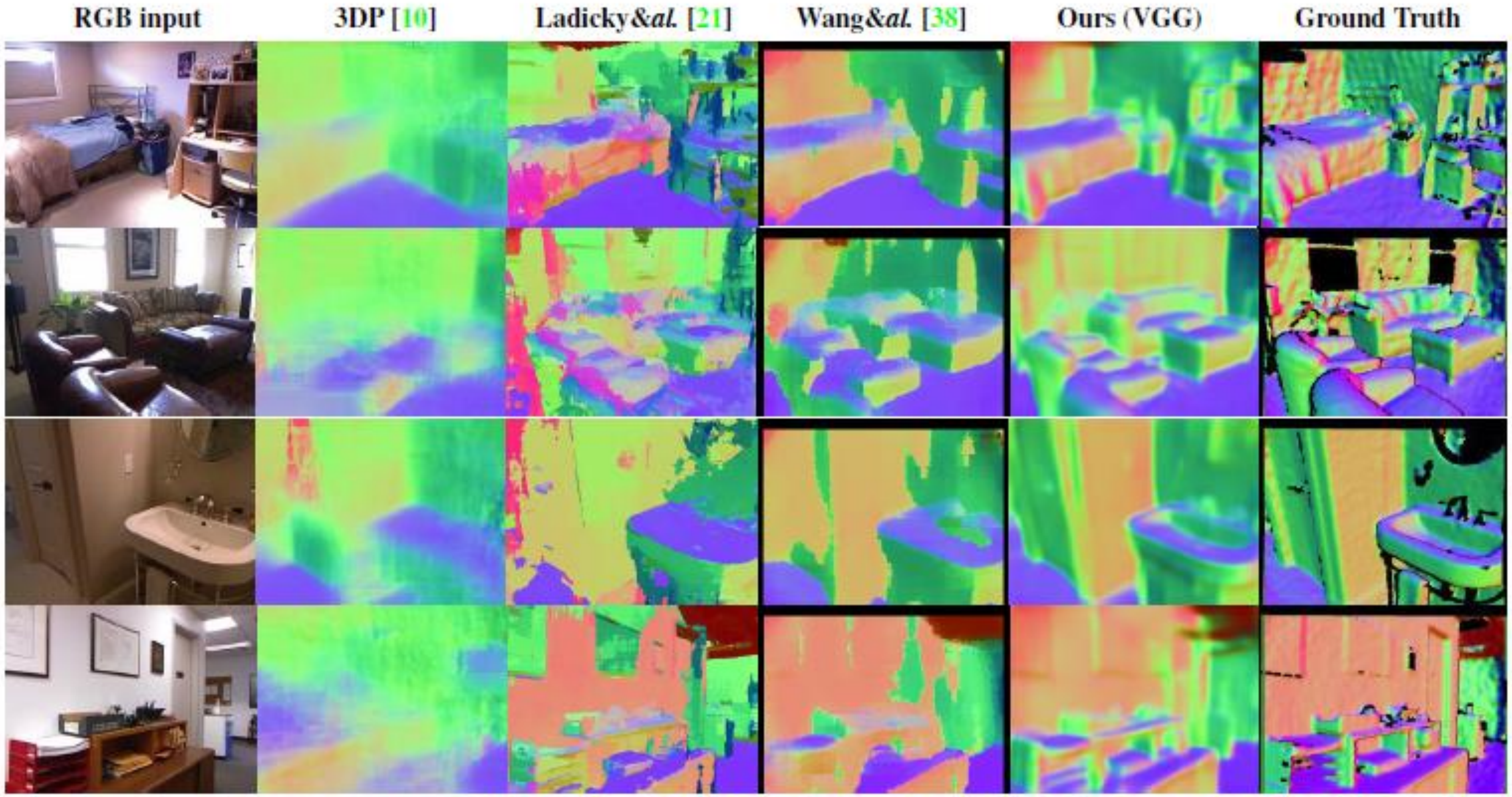


Figure 3. Comparison of surface normal maps.

NYU v2

Surface Normal Estimation (GT [21])					
	Angle Distance		Within t° Deg.		
	Mean	Median	11.25°	22.5°	30°
3DP [10]	35.3	31.2	16.4	36.6	48.2
Ladicky *&al.* [21]	33.5	23.1	27.5	49.0	58.7
Fouhey *&al.* [11]	35.2	17.9	40.5	54.1	58.9
Wang *&al.* [38]	26.9	14.8	42.0	61.2	68.2
Ours (AlexNet)	23.7	15.5	39.2	62.0	71.1
Ours (VGG)	**20.9**	**13.2**	**44.4**	**67.2**	**75.9**

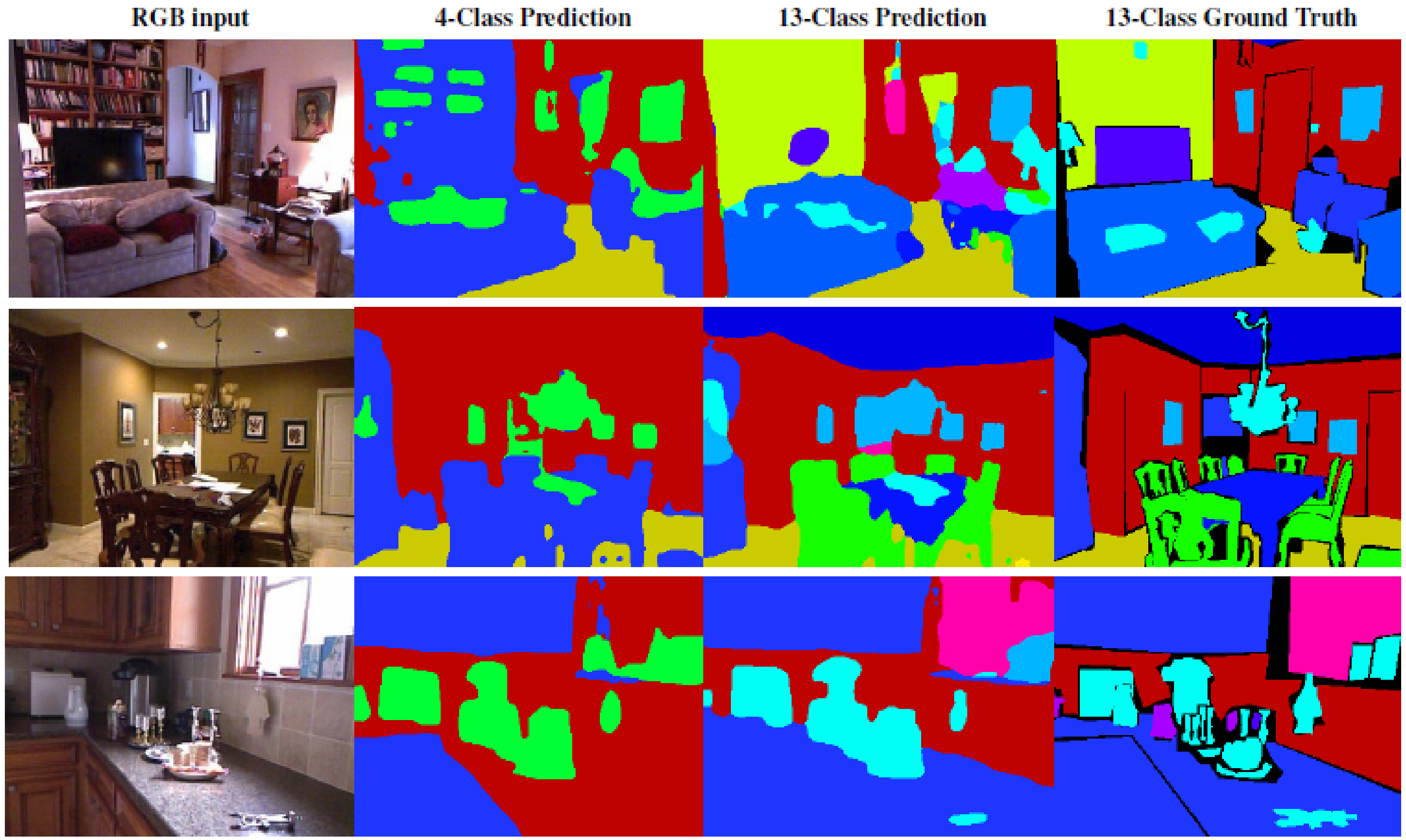


Figure 4. Example semantic labeling results for NYUDepth: (a) input image; (b) 4-class labeling result; (c) 13-class result; (d) 13-class ground truth.

NYU v2

4-Class Semantic Segmentation	Pixel	Class
Couprie &*al.* [6]	64.5	63.5
Khan &*al.* [15]	69.2	65.6
Stuckler &*al.* [33]	70.9	67.0
Mueller &*al.* [26]	72.3	71.9
Gupta &*al.* '13 [13]	78	–
Ours (AlexNet)	80.6	79.1
Ours (VGG)	**83.2**	**82.0**

13-Class Semantic	Pixel	Class
Couprie &*al.* [6]	52.4	36.2
Wang &*al.* [37]	–	42.2
Hermans &*al.* [17]	54.2	48.0
Khan &*al.* [15] *	58.3	45.1
Ours (AlexNet)	70.5	59.4
Ours (VGG)	**75.4**	**66.9**

40-Class Semantic Segmentation	Pix. Acc.	Per-Cls Acc.	Freq. Jaccard	Av. Jaccard
Gupta&*al.*'13 [13]	59.1	28.4	45.6	27.4
Gupta&*al.*'14 [14]	60.3	35.1	47.0	28.6
Long&*al.* [24]	65.4	**46.1**	49.5	34.0
Ours (AlexNet)	62.9	41.3	47.6	30.8
Ours (VGG)	**65.6**	45.1	**51.4**	**34.1**

Table 3. Semantic labeling on NYUDepth v2

* Khan&*al.* use a different overlapping label set.

Lessons learned

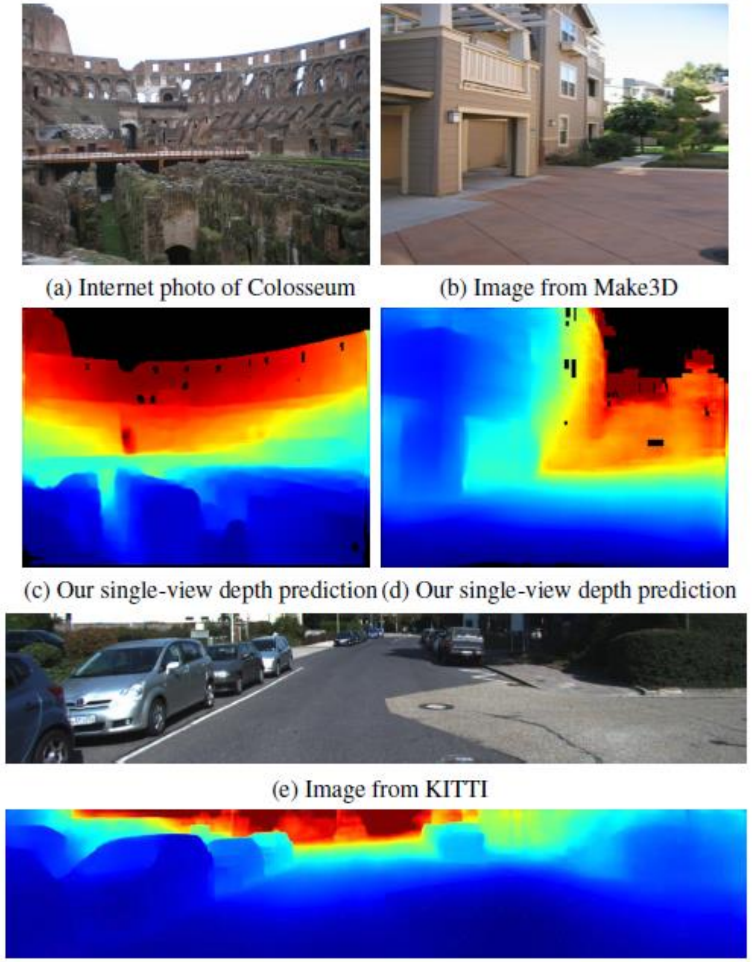
- Depth prediction, normal prediction, and semantic segmentation can be performed with similar architectures
- Multi-scale UNet-like architecture is effective
- Scale-invariant loss accounts for scale ambiguity of depth

MegaDepth: Learning Single-View Depth Prediction from Internet Photos

CVPR 2018

Zhengqi Li   Noah Snavely
Department of Computer Science & Cornell Tech, Cornell University

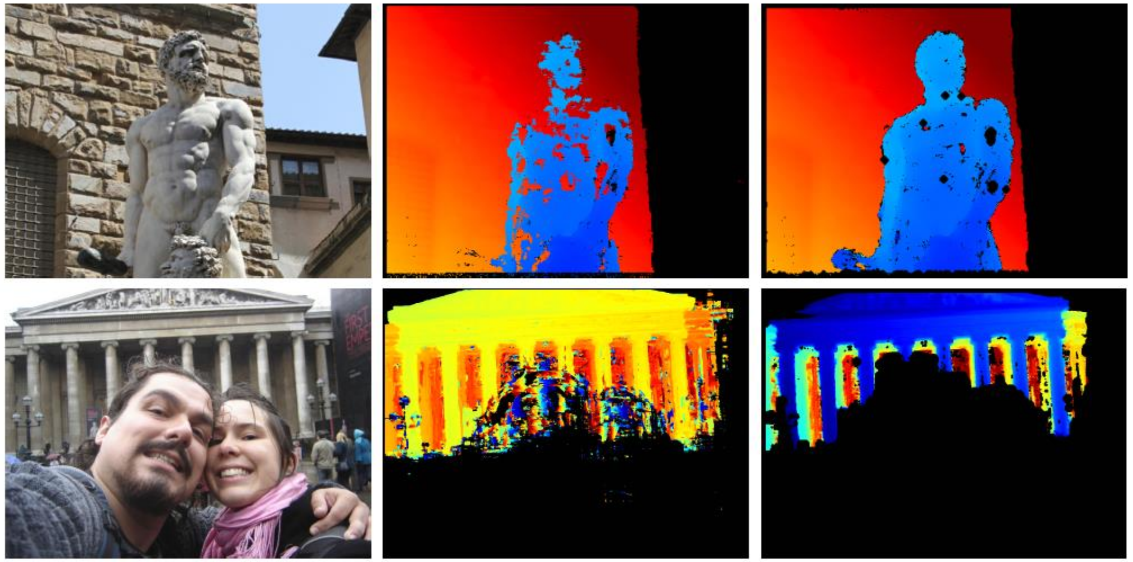
- Generate depth maps using MVS and semantic segmentation on internet photos
- Train with log depth and ordinal depth losses
- Test on scaled RMSE and ordinal depth, several datasets



(a) Internet photo of Colosseum (b) Image from Make3D
(c) Our single-view depth prediction (d) Our single-view depth prediction
(e) Image from KITTI
(f) Our single-view depth prediction

Creating training data

- COLMAP SfM+MVS
  - Modified to prune foreground less
- Semantic segmentation
  - PSPNet labels 150 categories
  - Discard foreground objects with <50% depth values
  - Discard sky depths
  - Enable foreground vs. background as ordinal prediction task
- Keep images with > 30% depth values (ignoring sky)
- Use ordinal depth labels (F/B) for others
- Dataset
  - 150K images processed
  - 100K depth images, 30K ordinal depth
  - Tanks&Temples also used for training

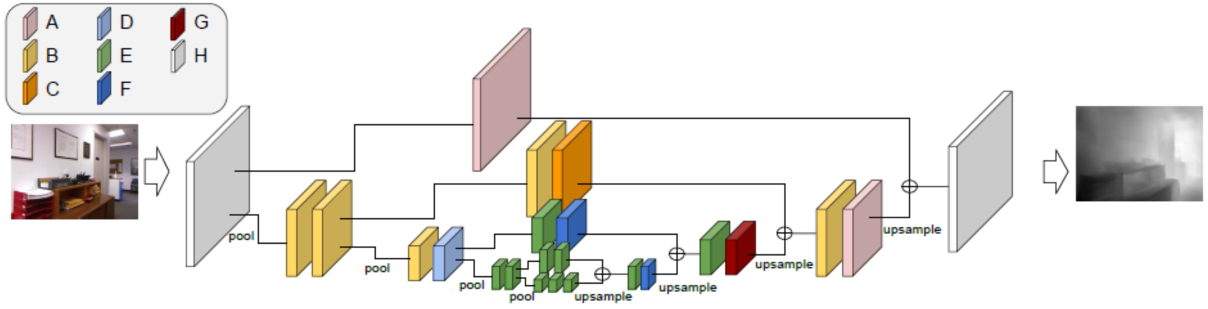


(a) Input photo (b) Raw depth (c) Refined depth

Figure 3: **Examples of automatic ordinal labeling.** Blue mask: foreground (F_{ord}) derived from semantic segmentation. Red mask: background (B_{ord}) derived from reconstructed depth.

Training

- Experimented with VGG, ResNet, Hourglass (like UNet)
  - Hourglass worked best
- Loss
  - Variance of log depth differences
  - L1 multiscale gradient
  - Ordinal depth



$$\mathcal{L}_{\text{si}} = \mathcal{L}_{\text{data}} + \alpha \mathcal{L}_{\text{grad}} + \beta \mathcal{L}_{\text{ord}}$$

$$\mathcal{L}_{\text{data}} = \frac{1}{n}\sum^{n}(R_i)^2 - \frac{1}{n^2}\left(\sum^{n} R_i\right)^2$$

$$R_i = L_i - L_i^*$$

$$\mathcal{L}_{\text{grad}} = \frac{1}{n}\sum_k \sum_i \left(|\nabla_x R_i^k| + |\nabla_y R_i^k|\right)$$

$$\mathcal{L}_{\text{ord}} = \begin{cases} \log\left(1 + \exp\left(P_{ij}\right)\right) & \text{if } P_{ij} \le \tau \\ \log\left(1 + \exp\left(\sqrt{P_{ij}}\right)\right) + c & \text{if } P_{ij} > \tau \end{cases}$$

$$P_{ij} = -r_{ij}^* \left(L_i - L_j\right)$$

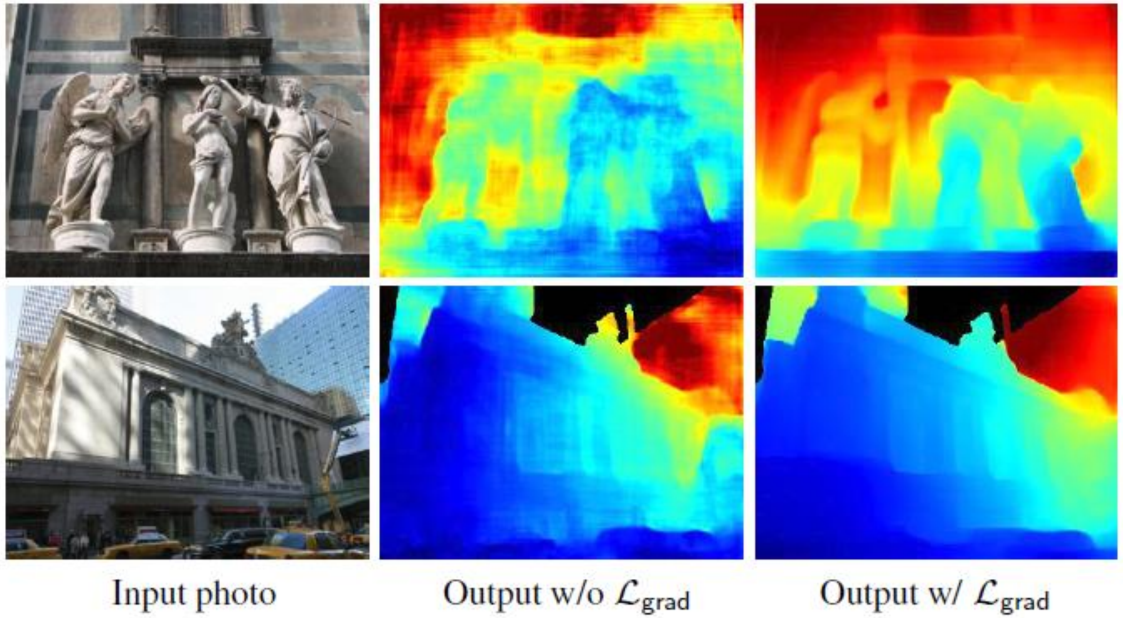


Figure 4: **Effect of $\mathcal{L}_{\text{grad}}$ term.** $\mathcal{L}_{\text{grad}}$ encourages predictions to match the ground truth depth gradient.

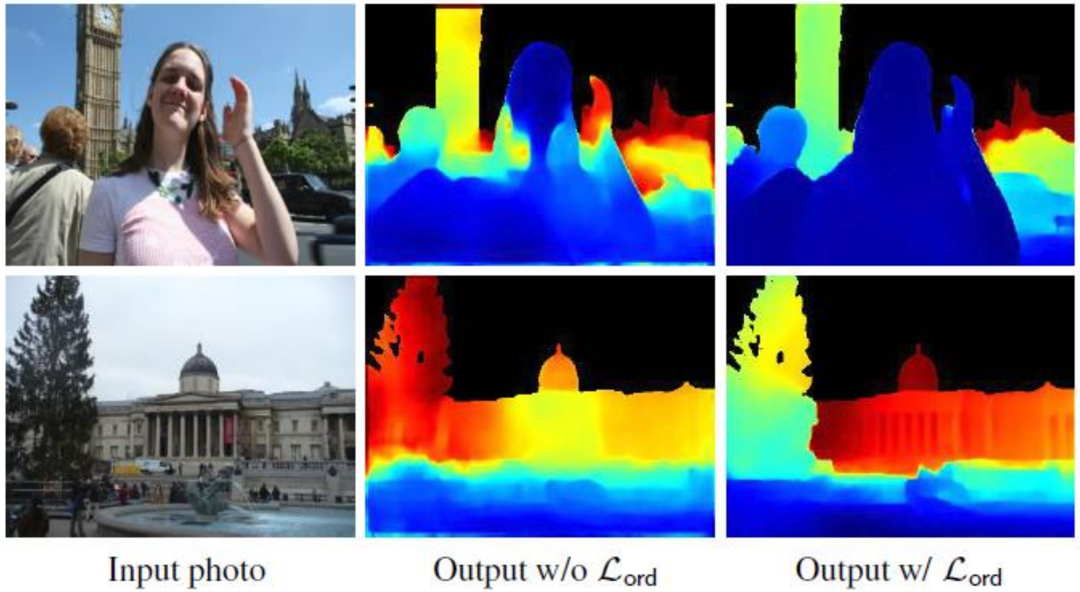


Figure 5: **Effect of $\mathcal{L}_{\text{ord}}$ term.** $\mathcal{L}_{\text{ord}}$ tends to corrects ordinal depth relations for hard-to-construct objects such as the person in the first row and the tree in the second row.

Network	si-RMSE	SDR$^=$%	SDR$^{\neq}$%	SDR%
VGG* [6]	0.116	31.28	28.63	29.78
VGG (full)	0.115	29.64	27.22	28.40
ResNet (full)	0.124	**27.32**	25.35	26.27
HG (full)	**0.104**	27.73	**24.36**	**25.82**

Table 1: **Results on the MD test set (places unseen during training) for several network architectures.** For VGG* we use the same loss and network architecture as in [6] for comparison to [6]. Lower is better.

Method	si-RMSE	SDR$^=$%	SDR$^{\neq}$%	SDR%
$\mathcal{L}_{\text{data}}$ only	0.148	33.20	30.65	31.75
$+\mathcal{L}_{\text{grad}}$	0.123	**26.17**	28.32	27.11
$+\mathcal{L}_{\text{grad}} + \mathcal{L}_{\text{ord}}$	**0.104**	27.73	**24.36**	**25.82**

Table 2: **Results on MD test set (places unseen during training) for different loss configurations.** Lower is better.

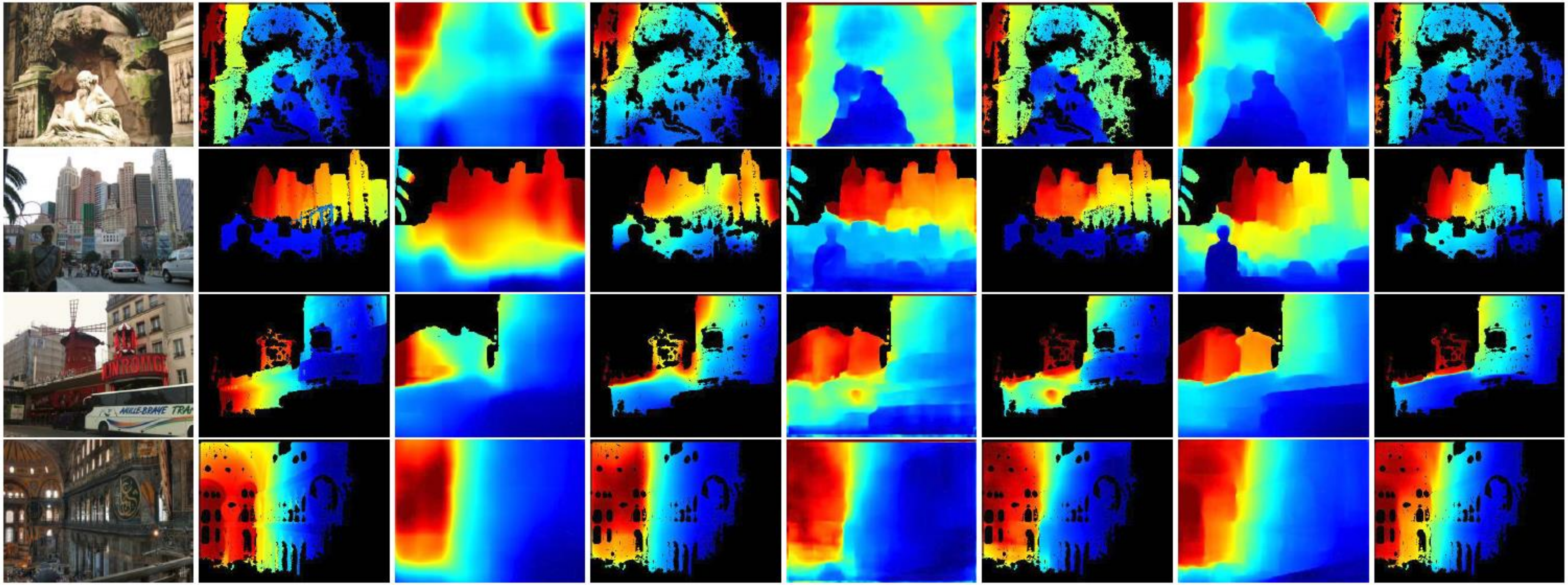
si-RMSE = scale-invariant RMSE of log depth
SDR = ordinal disagreement with sparse point pairs

$$\mathrm{SDR}(D, D^*) = \frac{1}{n} \sum_{i,j \in \mathcal{P}} \mathbb{1}\left(\mathrm{ord}(D_i, D_j) \neq \mathrm{ord}(D_i^*, D_j^*)\right)$$

$$\mathrm{ord}(D_i, D_j) = \begin{cases} 1 & \text{if } \frac{D_i}{D_j} > 1 + \delta \\ -1 & \text{if } \frac{D_i}{D_j} < 1 - \delta \\ 0 & \text{if } 1 - \delta \le \frac{D_i}{D_j} \le 1 + \delta \end{cases}$$

Test set	Error measure	Raw MD	Clean MD
Make3D	RMS	11.41	**5.322**
	Abs Rel	0.614	**0.364**
	log10	0.386	**0.152**
KITTI	RMS	12.15	**6.680**
	RMS(log)	0.582	**0.414**
	Abs Rel	0.433	**0.368**
	Sq Rel	3.927	**2.587**
DIW	WHDR%	31.32	**24.55**

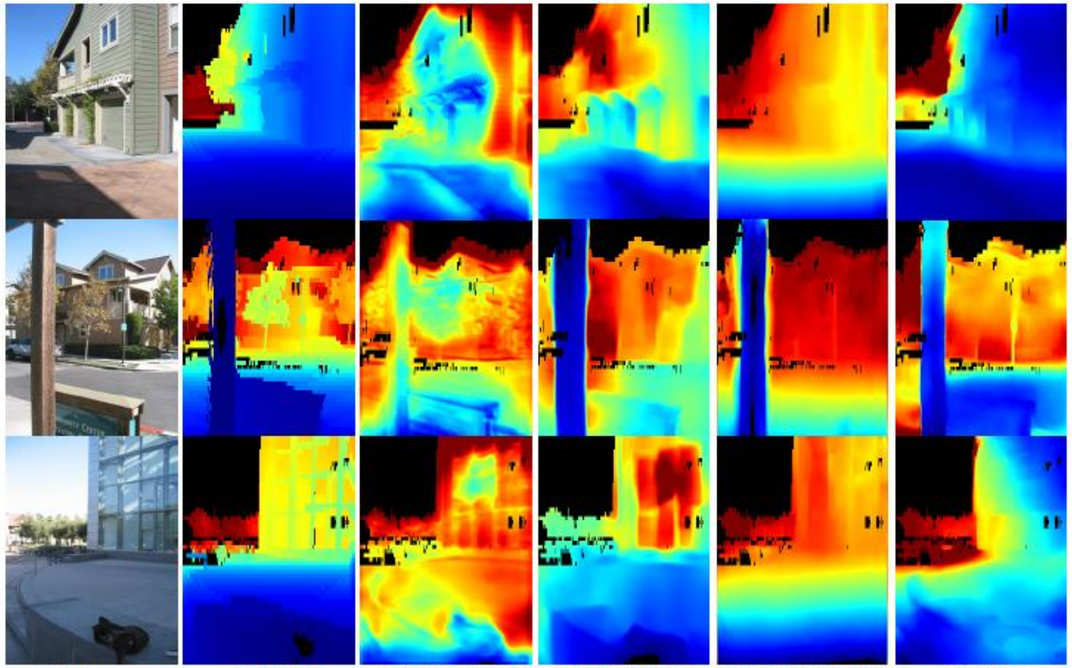
Table 3: **Results on three different test sets with and without our depth refinement methods**. *Raw MD* indicates raw depth data; *Clean MD* indicates depth data using our refinement methods. Lower is better for all error measures.



(a) Image (b) GT (c) VGG* (d) VGG* (M) (e) ResNet (f) ResNet (M) (g) HG (h) HG (M)

Training set	Method	RMS	Abs Rel	log10
Make3D	Karsch *et al.* [16]	9.20	0.355	0.127
	Liu *et al.* [24]	9.49	0.335	0.137
	Liu *et al.* [22]	8.60	0.314	0.119
	Li *et al.* [20]	7.19	0.278	0.092
	Laina *et al.* [19]	4.45	0.176	0.072
	Xu *et al.* [39]	4.38	0.184	0.065
NYU	Eigen *et al.* [6]	6.89	0.505	0.198
	Liu *et al.* [22]	7.20	0.669	0.212
	Laina *et al.* [19]	7.31	0.669	0.216
KITTI	Zhou *et al.* [43]	8.39	0.651	0.231
	Godard *et al.* [13]	9.88	0.525	0.319
DIW	Chen *et al.* [4]	7.25	0.550	0.200
MD	Ours	6.23	0.402	0.156
MD+Make3D	Ours	4.25	0.178	0.064

Table 4: **Results on Make3D for various training datasets and methods.** The first column indicates the training dataset. Errors for "Ours" are averaged over four models trained/validated on MD. Lower is better for all metrics.



(a) Image (b) GT (c) DIW [4] (d) NYU [6] (e) KITTI [13] (f) MD

Figure 7: **Depth predictions on Make3D.** The last four columns show results from the best models trained on non-Make3D datasets (final column is our result).

Training set	Method	RMS	RMS(log)	Abs Rel	Sq Rel
KITTI	Liu *et al.* [23]	6.52	0.275	0.202	1.614
	Eigen *et al.* [7]	6.31	0.282	0.203	1.548
	Zhou *et al.* [43]	6.86	0.283	0.208	1.768
	Godard *et al.* [13]	5.93	0.247	0.148	1.334
Make3D	Laina *et al.* [19]	8.68	0.422	0.339	3.136
	Liu *et al.* [22]	8.70	0.447	0.362	3.465
NYU	Eigen *et al.* [6]	10.37	0.510	0.521	5.016
	Liu *et al.* [22]	10.10	0.526	0.540	5.059
	Laina *et al.* [19]	10.07	0.527	0.515	5.049
CS	Zhou *et al.* [43]	7.58	0.334	0.267	2.686
DIW	Chen *et al.* [4]	7.12	0.474	0.393	3.260
MD	Ours	6.68	0.414	0.368	2.587
MD+KITTI	Ours	5.25	0.229	0.139	1.325

Table 5: **Results on the KITTI test set for various training datasets and approaches.** Columns are as in Table 4.

Training set	Method	WHDR%
DIW	Chen *et al.* [4]	22.14
KITTI	Zhou *et al.* [43]	31.24
	Godard *et al.* [13]	30.52
NYU	Eigen *et al.* [6]	25.70
	Laina *et al.* [19]	45.30
	Liu *et al.* [22]	28.27
Make3D	Laina *et al.* [19]	31.65
	Liu *et al.* [22]	29.58
MD	Ours	24.55

Table 6: **Results on the DIW test set for various training datasets and approaches.** Columns are as in Table 4.

Lessons Learned

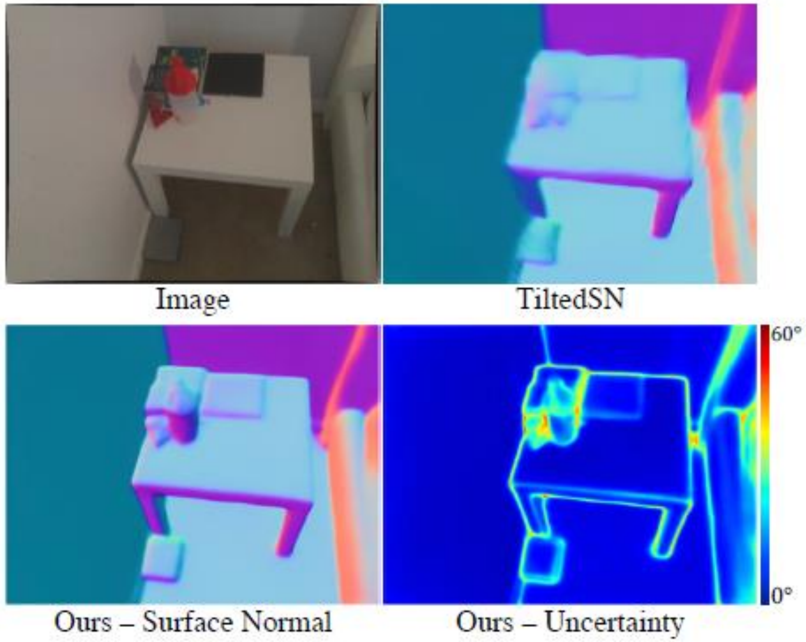
- SfM/MVS can be used to effectively create single-view depth training sets
- Such training generalizes to other datasets
- Combination of losses for scale-invariant depth, log depth gradients, and ordinal depth is effective

Estimating and Exploiting the Aleatoric Uncertainty in Surface Normal Estimation

ICCV 2021

Gwangbin Bae Ignas Budvytis Roberto Cipolla
University of Cambridge
{gb585,ib255,rc10001}@cam.ac.uk

- Takes into account uncertainty of ground truth normal and improves detail
  - Predict probability distribution of normals, with loss as function of uncertainty
  - Coarse-to-fine, focus on uncertain pixels in upsampling



- Baseline: minimize negative log likelihood of Gaussian-like distribution on unit sphere (von Mises-Fisher dist)
  - Corresponds to minimizing L2 distance weighted by uncertainty

$$\mathcal{L} = -\frac{1}{N}\sum_i \log p_i(\mathbf{n}_i^{\mathrm{gt}} | \boldsymbol{\theta}_i(\mathcal{I}, \mathbf{W}))$$

$$p_{\mathrm{vonMF},i}(\mathbf{n}_i | \boldsymbol{\mu}_i, \kappa_i) = \frac{\kappa_i \exp(\kappa_i \boldsymbol{\mu}_i^T \mathbf{n}_i)}{4\pi \sinh \kappa_i}$$

$$\mathcal{L}_{\mathrm{vonMF},i} = -\log \kappa_i + \log \sinh \kappa_i - \kappa_i \boldsymbol{\mu}_i^T \mathbf{n}_i^{\mathrm{gt}}$$

Uncertainty cost

Uncertainty weight

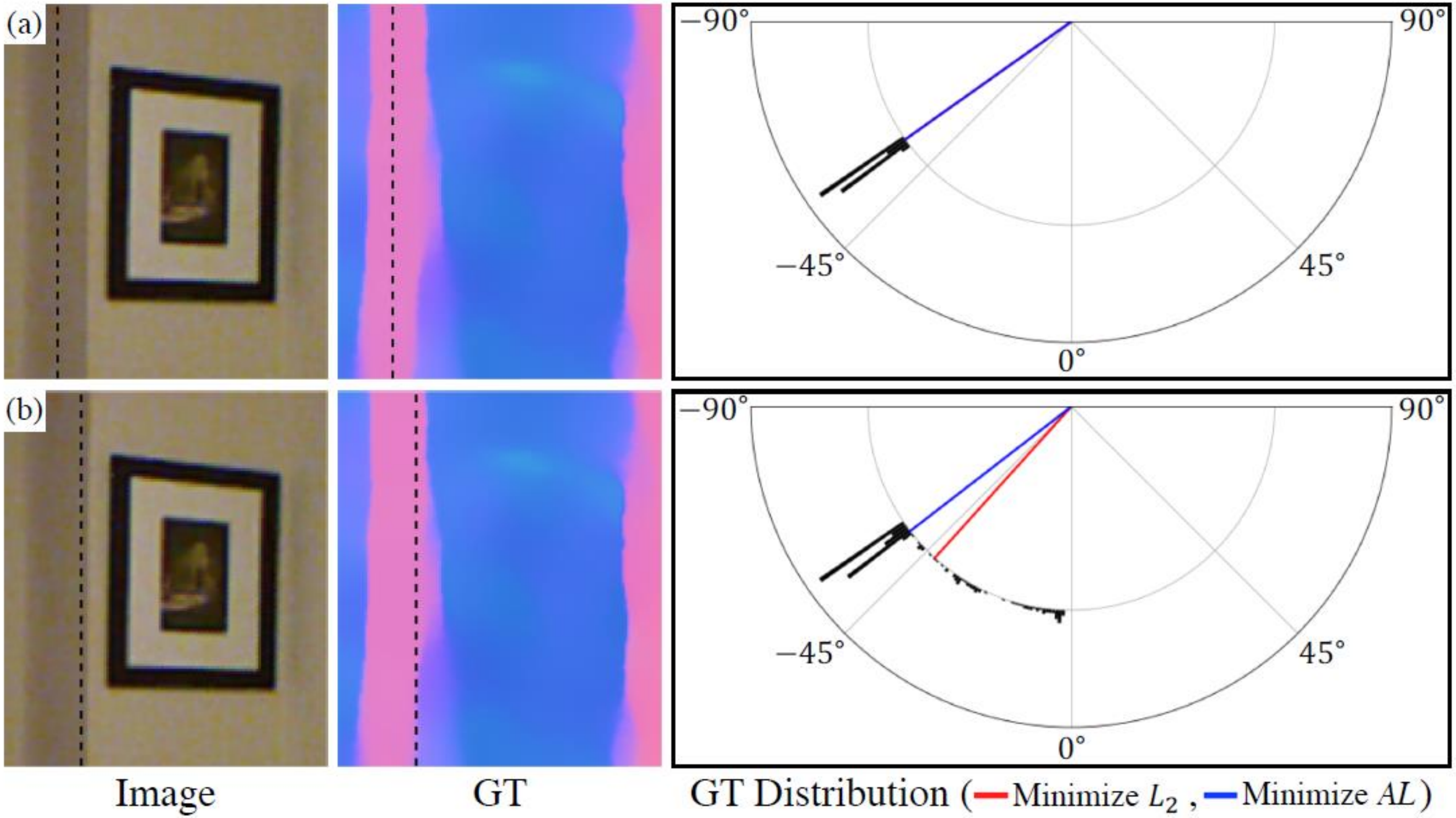
correlation

- Proposed: minimize angle between gt and prediction

$$p_{\mathrm{AngMF},i}(\mathbf{n}_i | \boldsymbol{\mu}_i, \kappa_i) = \frac{(\kappa_i^2 + 1)\exp(-\kappa_i \cos^{-1} \boldsymbol{\mu}_i^T \mathbf{n}_i)}{2\pi(1 + \exp(-\kappa_i \pi))} \tag{4}$$

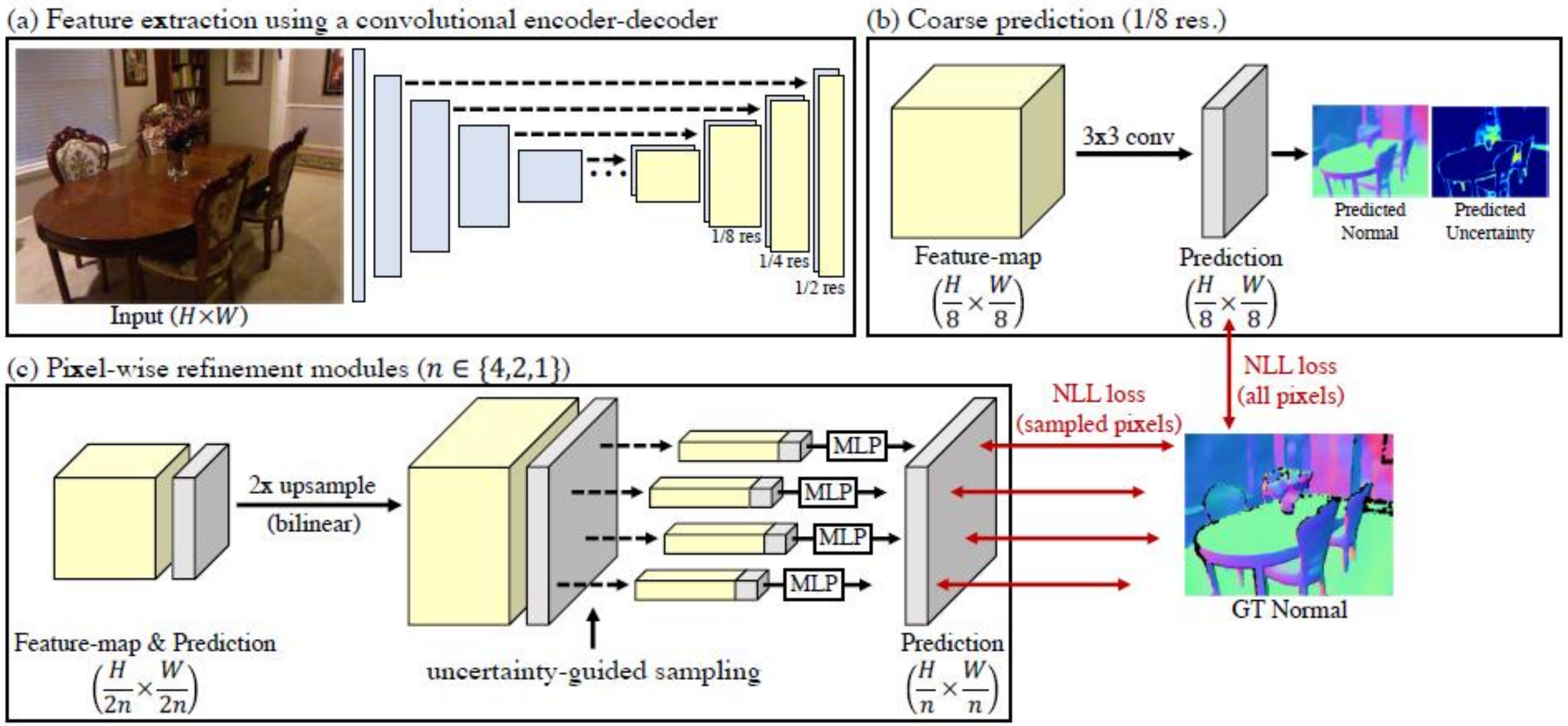
and $$\mathcal{L}_{\mathrm{AngMF},i} = -\log(\kappa_i^2 + 1) + \log(1 + \exp(-\kappa_i \pi)) + \kappa_i \cos^{-1} \boldsymbol{\mu}_i^T \mathbf{n}_i^{\mathrm{gt}}. \tag{5}$$

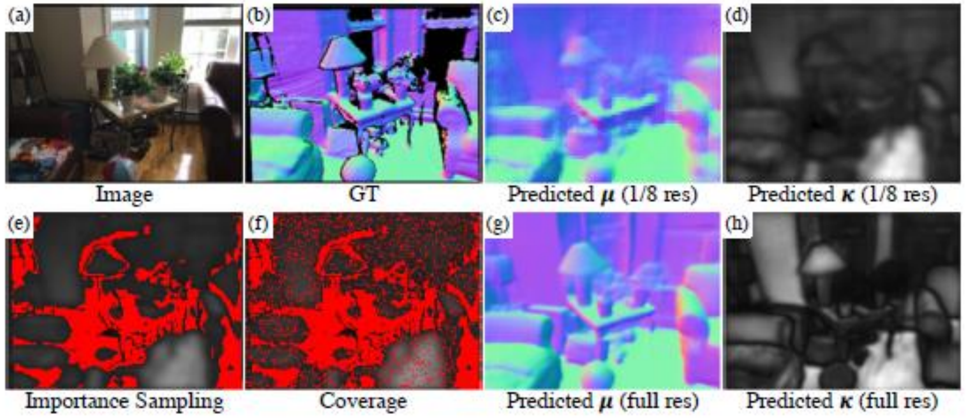
Minimizing angular loss is more robust



Training with focused refinement

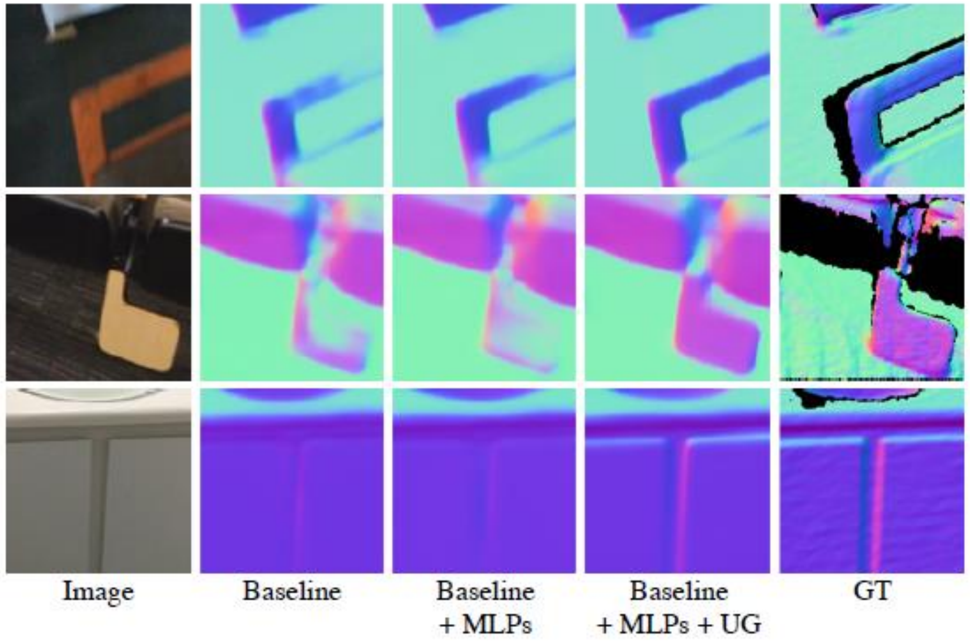
- Coarse to fine
- Network predicts normal and uncertainty
- Training focuses on pixels with high uncertainty
  - Prevent network only focusing on low-uncertainty planar regions





Architecture	Loss fn.	mean	median	rmse	5.0°	7.5°	11.25°	22.5°	30°
baseline (convolutional encoder-decoder with skip connections [2])	L_2	13.53	7.22	**21.16**	35.10	51.44	65.08	82.38	87.83
	NLL-vonMF	14.10	7.19	22.14	36.20	51.46	64.09	80.80	86.34
	AL	13.45	6.70	21.78	38.65	54.04	66.73	82.46	87.53
	NLL-AngMF	13.82	6.60	22.47	39.69	54.30	65.97	81.64	86.71
baseline + pixel-wise MLPs	*NLL-AngMF*	13.59	6.53	22.23	39.92	54.79	67.03	82.18	87.06
baseline + pixel-wise MLPs + uncertainty-guided sampling	*NLL-AngMF*	**13.17**	**6.48**	21.57	**40.09**	**55.19**	**67.62**	**83.10**	**87.97**

Table 1. (top) The baseline network is trained with different loss functions. The proposed *NLL-AngMF* shows higher accuracy than *NLL-vonMF*, except for RMSE. *NLL-AngMF* and *NLL-vonMF* are *AL* and L_2 with learned attenuation, respectively. As the training is biased to low-uncertainty pixels, the median error decreases, while RMSE increases. (bottom) The bias in training is solved by the proposed decoder modules. Both the pixel-wise MLPs and the uncertainty-guided sampling lead to improvement in all metrics.



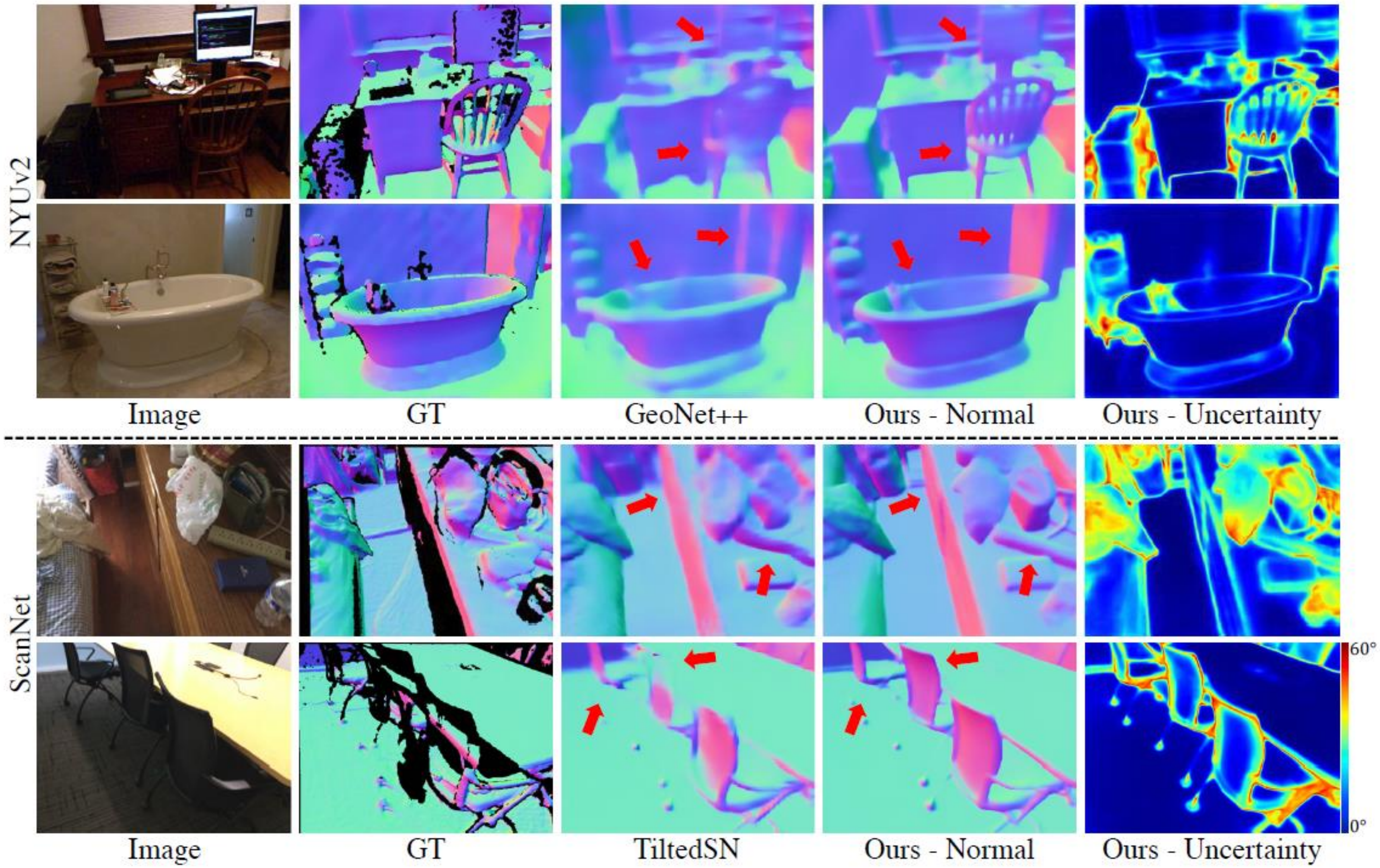


Figure 7. Qualitative comparison against GeoNet++ [32] and TiltedSN [6]. The predictions made by our method show clearer object boundaries and preserve the fine-details of the scene geometry (see the regions pointed by the red arrows). The estimated uncertainty is high near object boundaries and on small structures. More examples are provided in the supplementary material.

Method	Train	mean	median	rmse	11.25°	22.5°	30°
Ladicky et al. [22]		33.5	23.1	-	27.5	49.0	58.7
Fouhey et al. [10]		35.2	17.9	-	40.5	54.1	58.9
Deep3D [39]		26.9	14.8	-	42.0	61.2	68.2
Eigen et al. [7]		20.9	13.2	-	44.4	67.2	75.9
SkipNet [1]		19.8	12.0	28.2	47.9	70.0	77.8
SURGE [37]	N	20.6	12.2	-	47.3	68.9	76.6
GeoNet [31]		19.0	11.8	26.9	48.4	71.5	79.5
PAP [42]		18.6	11.7	25.5	48.8	72.2	79.8
GeoNet++ [32]		18.5	11.2	26.7	50.2	73.2	80.7
Ours	N	**14.9**	**7.5**	**23.5**	**62.2**	**79.3**	**85.2**
FrameNet[18]		18.6	11.0	26.8	50.7	72.0	79.5
VPLNet[38]	S	18.0	9.8	-	54.3	73.8	80.7
TiltedSN[6]		16.1	**8.1**	25.1	**59.8**	77.4	83.4
Ours	S	**16.0**	8.4	**24.7**	59.0	**77.5**	**83.7**

Table 3. Surface normal accuracy on NYUv2 [33]. The proposed method shows state-of-the-art performance. (top) The networks are trained on NYUv2. (bottom) The networks are trained on ScanNet [4] and tested on NYUv2 without fine-tuning.

Method	mean	median	rmse	11.25°	22.5°	30°
FrameNet[18]	14.7	7.7	22.8	62.5	80.1	85.8
VPLNet[38]	13.8	6.7	-	66.3	81.8	87.0
TiltedSN[6]	12.6	6.0	21.1	69.3	83.9	88.6
Ours	**11.8**	**5.7**	**20.0**	**71.1**	**85.4**	**89.8**

Table 4. Surface normal accuracy on ScanNet [4]. Our method outperforms other methods across all metrics.

Lessons learned

- Minimizing angular error is better than minimizing correlation/L2 for surface normal prediction

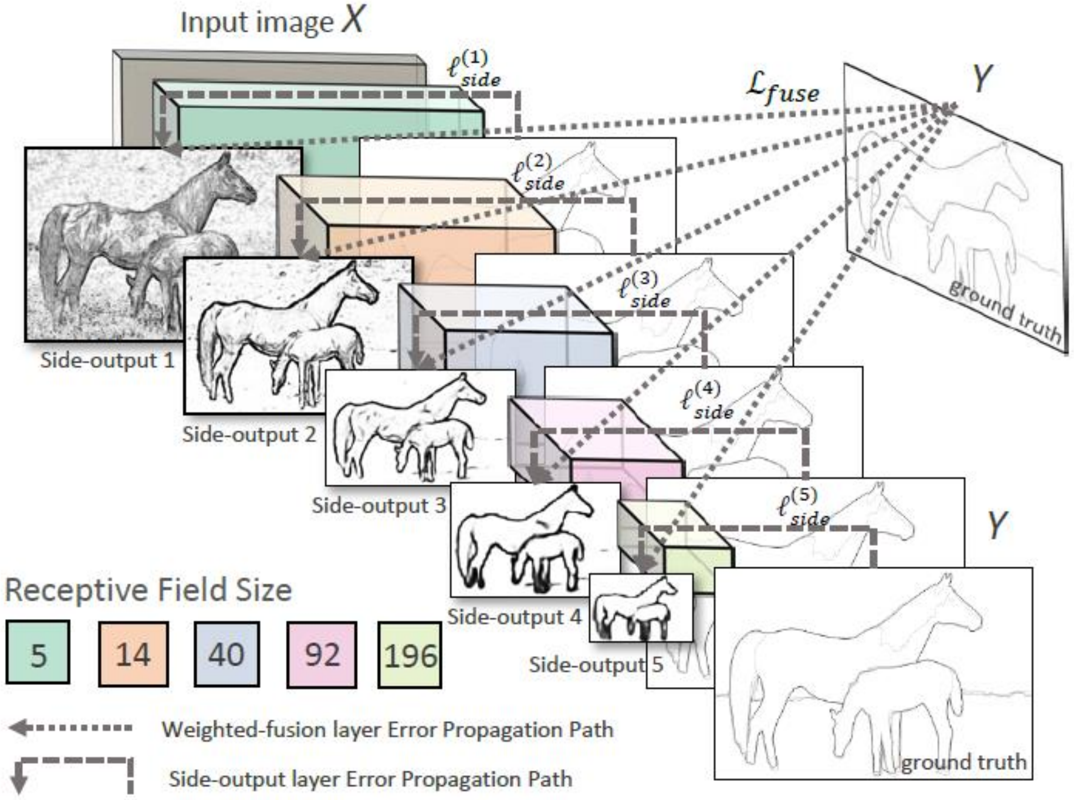
- Accounting for prediction/gt uncertainty and focusing refinement on less certain pixels is helpful

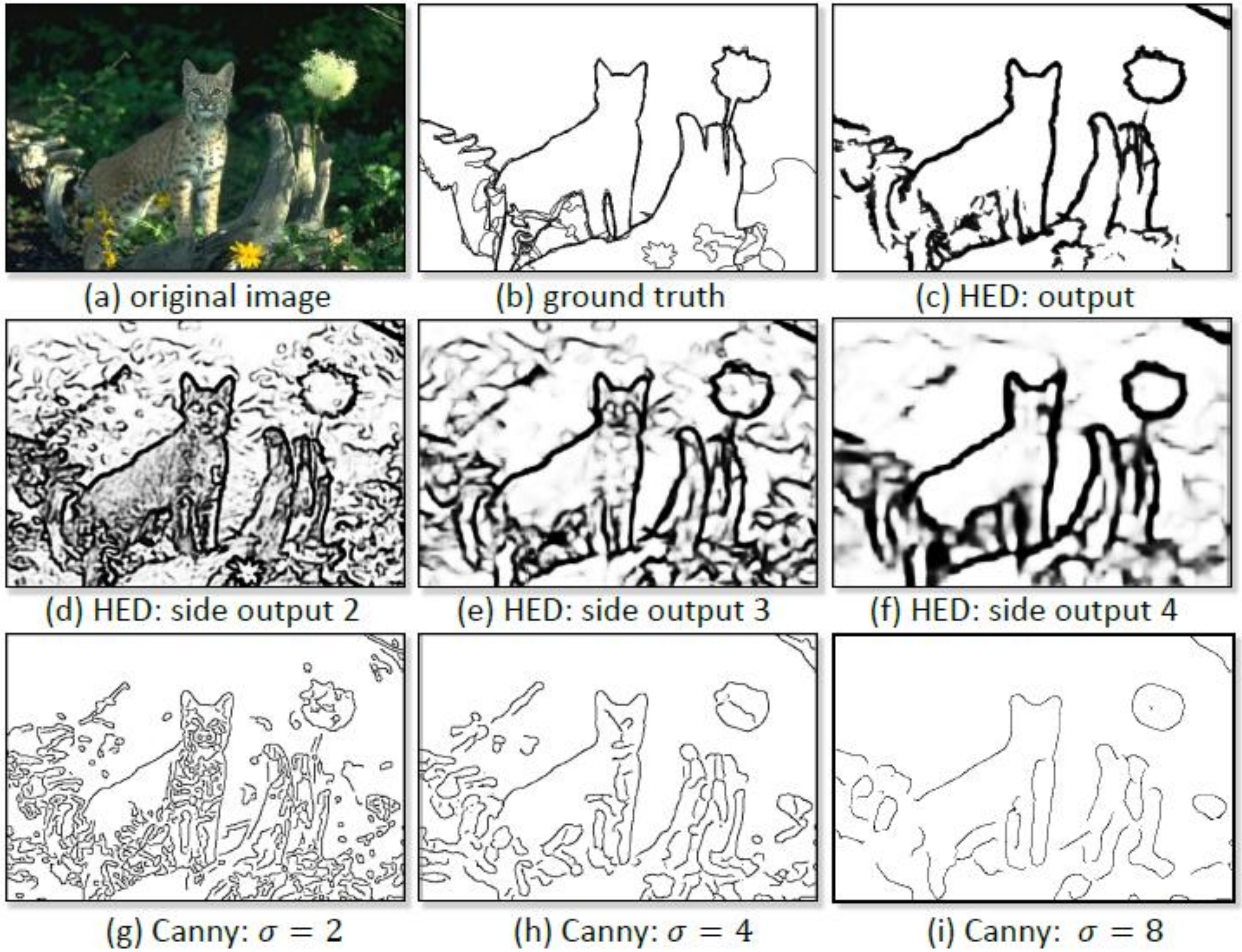
Holistically-Nested Edge Detection

ICCV 2015

Saining Xie
Dept. of CSE and Dept. of CogSci
University of California, San Diego

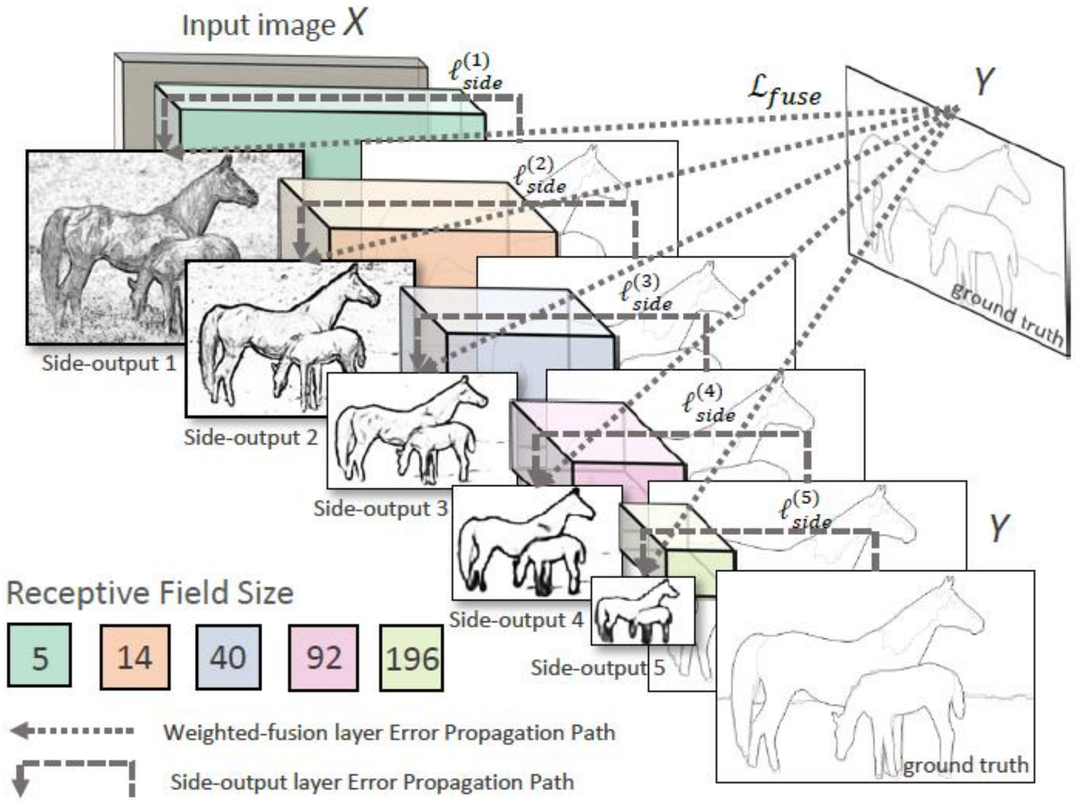
Zhuowen Tu
Dept. of CogSci and Dept. of CSE
University of California, San Diego





Approach

- Output boundary pixels at each scale
- Balance loss for positive and negative pixels
- Minimize loss of each scale and of fused prediction
  - Per scale loss helps prediction on fine boundaries



$$\ell^{(m)}_{\text{side}}(\mathbf{W}, \mathbf{w}^{(m)}) = -\beta \sum_{j \in Y_+} \log \Pr(y_j = 1 | X; \mathbf{W}, \mathbf{w}^{(m)}) - (1-\beta) \sum_{j \in Y_-} \log \Pr(y_j = 0 | X; \mathbf{W}, \mathbf{w}^{(m)}) \quad (2)$$

Results (qual from blog)

- Applicable to multiple datasets, e.g. BSDS500 and NYUv2

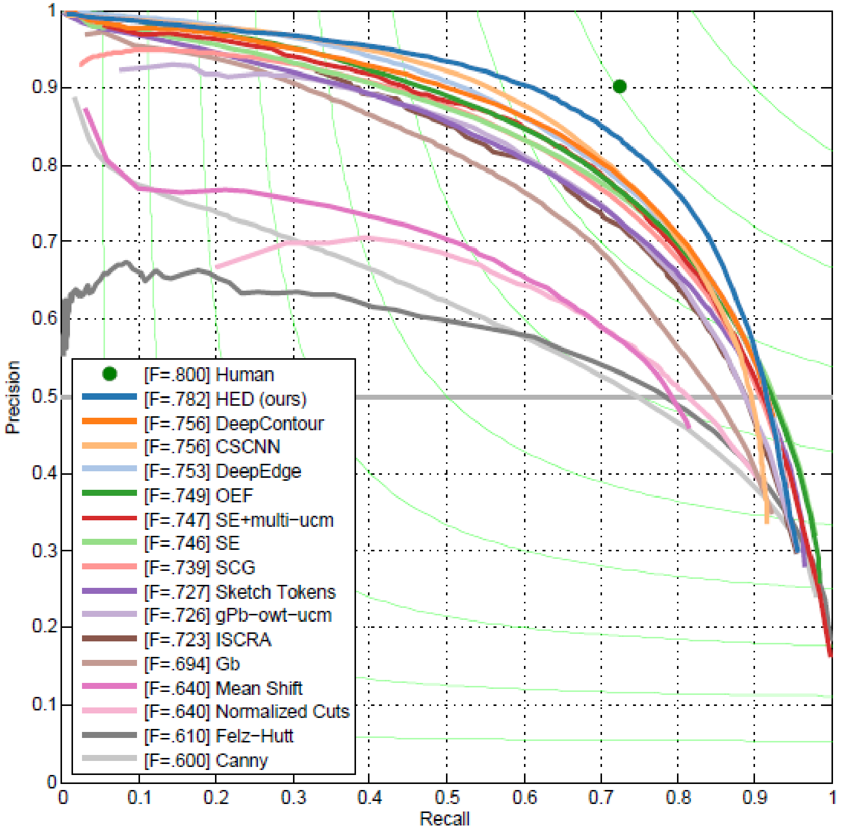
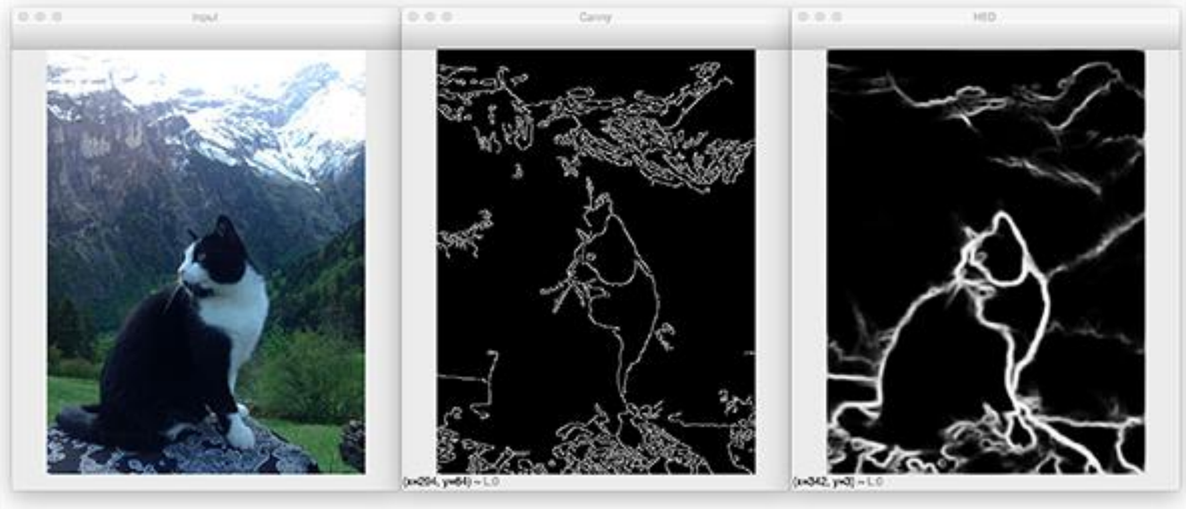
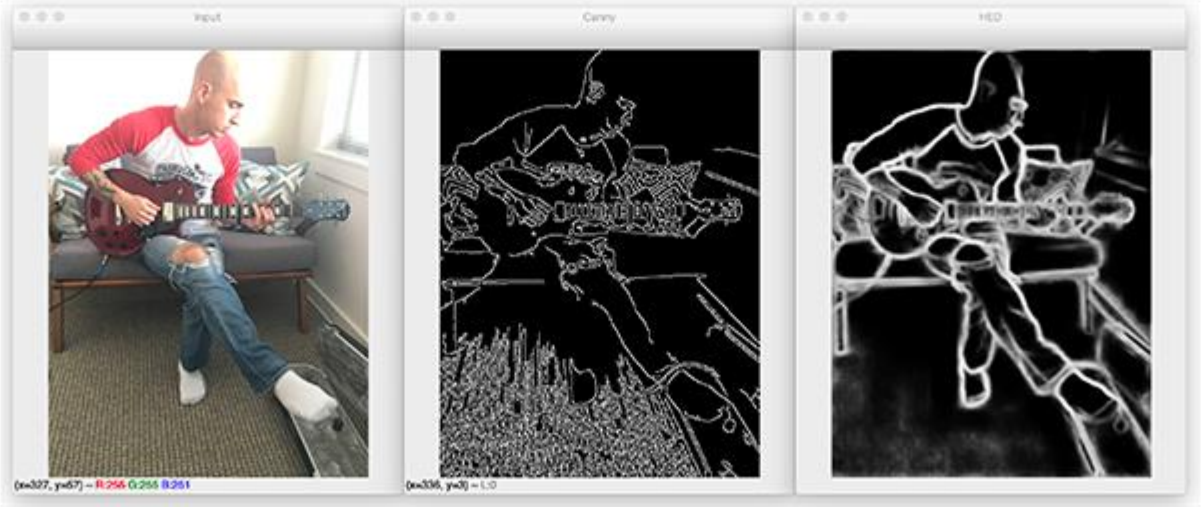


Figure 5. Results on the BSDS500 dataset. Our proposed HED framework achieves the best result (ODS=.782). Compared to several recent CNN-based edge detectors, our approach is also orders of magnitude faster. See Table 4 for a detailed discussion.





https://www.pyimagesearch.com/2019/03/04/holistically-nested-edge-detection-with-opencv-and-deep-learning/

https://www.pyimagesearch.com/2019/03/04/holistically-nested-edge-detection-with-opencv-and-deep-learning/

DOOBNet: Deep Object Occlusion Boundary Detection from an Image

ACCV 2018

Guoxia Wang[1], Xiaohui Liang[1], and Frederick W. B. Li[2]

[1]Beihang University, [2]University of Durham

- Goal: Predict object boundary with figure/ground
- Proposes tunable weighting on positive/negative loss
- ResNet50 encoder/decoder

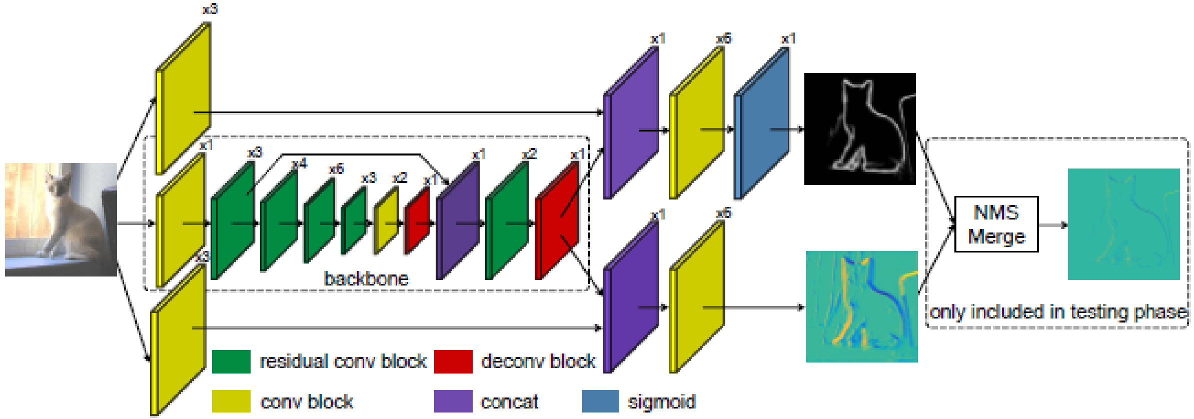
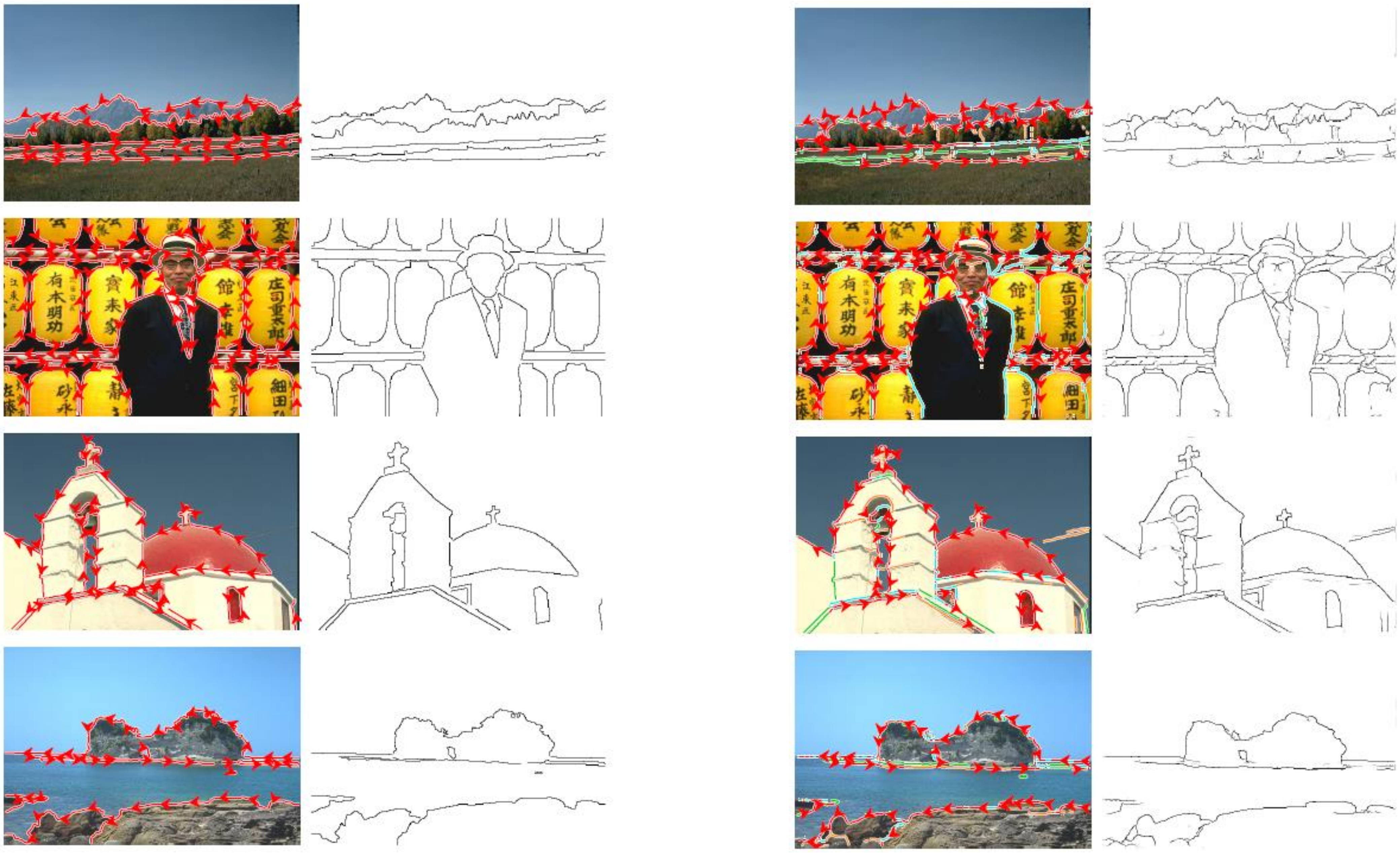


Fig. 4. DOOBNet Architecture.

GT

Prediction



GT

Prediction

Lessons learned

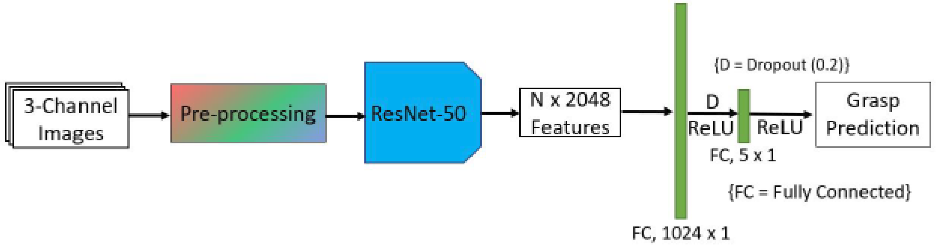
- Coarse-to-fine approach and UNet-style encoder/decoders are effective for edge prediction
- Balancing loss of positive and negative examples is critical since edges are sparse
- Relatively little recent work in this area, and edge detection may be seen as an implicit part of other problems now

What is it for?

- Computational photography
  - Selective blur
  - Relighting
- Photo tour, novel view synthesis
- Navigation, grasping, interaction
- Captioning, visual relationships?

Research ideas

- Main improvements likely through better use of self-supervised data/training
- How should depth/normal/boundary impact other vision tasks?
- Is it most useful when you want something to perform many tasks, including actions?
  - Taskonomy shows surface normal prediction is one of the best pre-training tasks
- Use of single-view depth/normal/boundary for grasping and manipulation



Kumra et al. IROS 2017

Summary

- Depth, normal, boundary prediction can be solved with similar architectures
- Works in past few years focus on losses and acquisition of training samples
- Biggest open question: Are explicit geometry representations needed or helpful for downstream tasks?