ECS289 VISUAL RECOGNITION

Depth Map Prediction from a Single Image using a Multi-Scale Deep Network – NIPS 2014

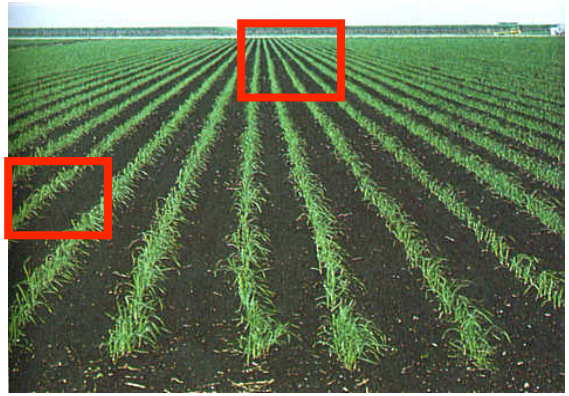
- D. Eigen, C. Puhrsch, and R. Fergus

Presenter Wei-Chih Chen(Michael)

2015/11/12

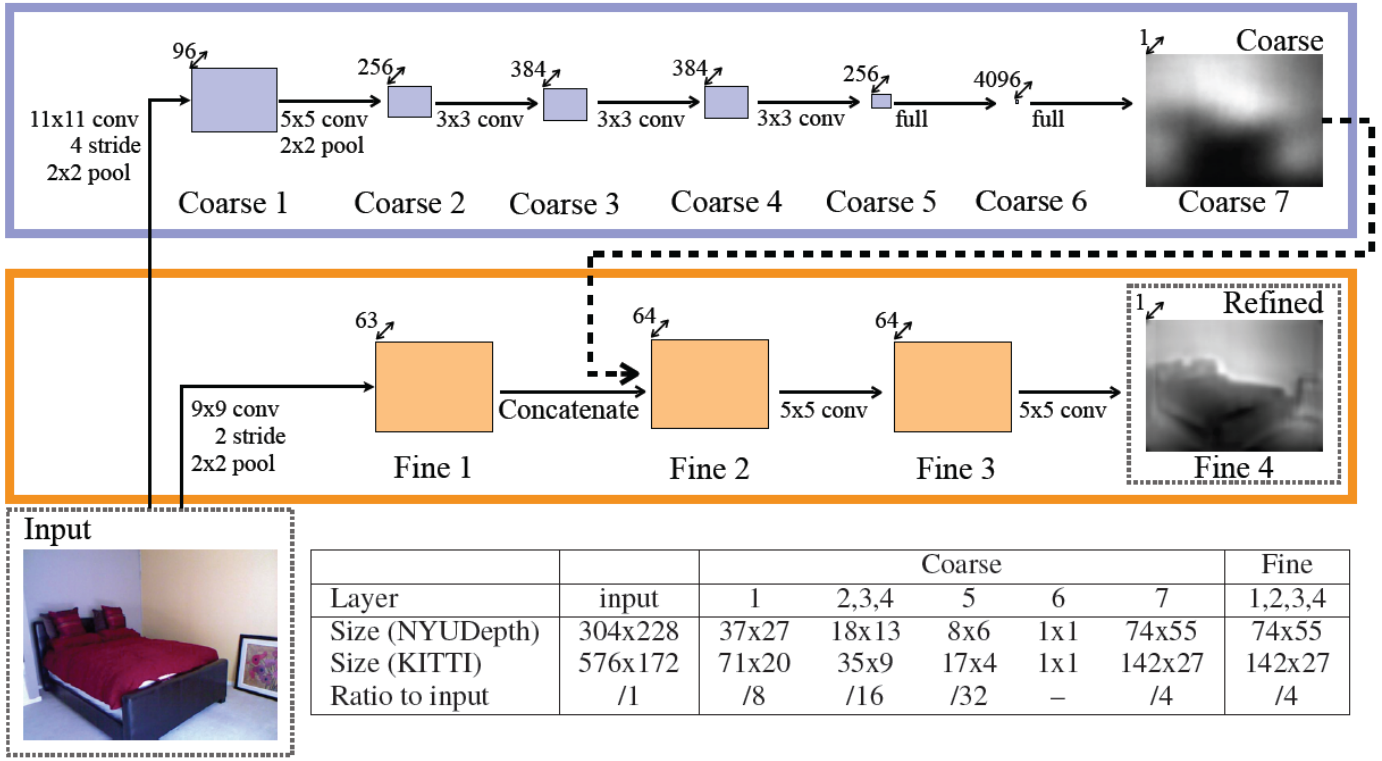
Introduction

- Estimate depth from a single image:
  - Line angles
  - Texture variations
  - Object sizes
  - Haze color(far)



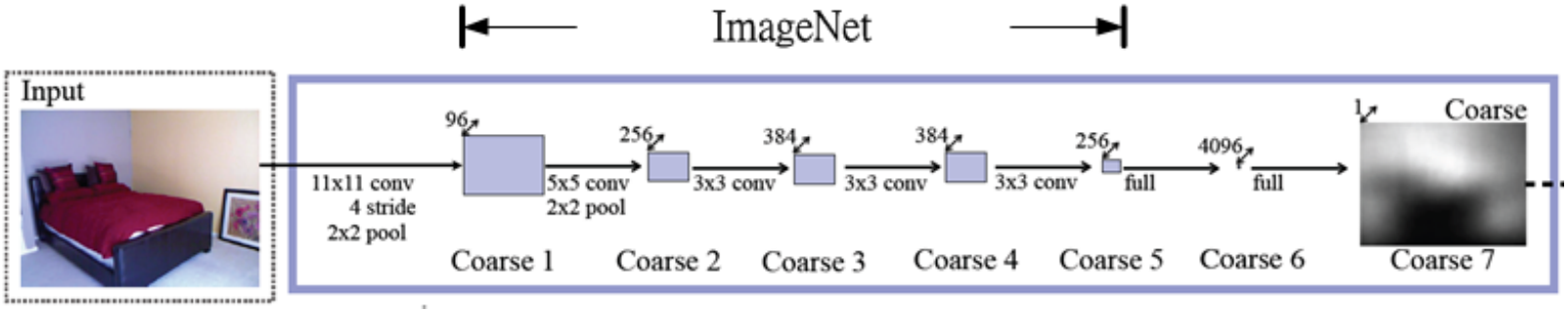
Model architecture

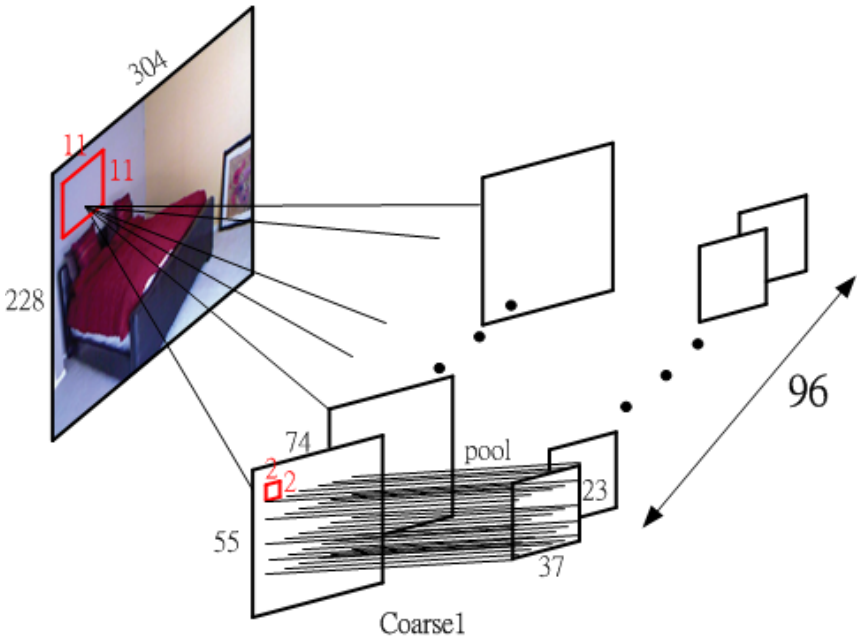
- Global Coarse-Scale Network-learn spatial information
- Local Fine-Scale Network-local refine

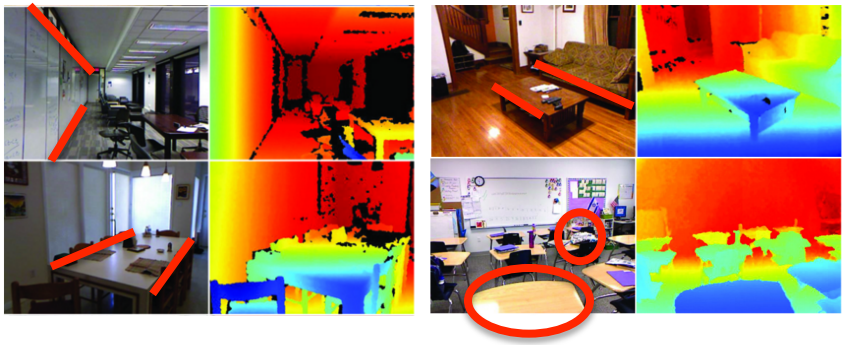


		Coarse					Fine
Layer	input	1	2,3,4	5	6	7	1,2,3,4
Size (NYUDepth)	304x228	37x27	18x13	8x6	1x1	74x55	74x55
Size (KITTI)	576x172	71x20	35x9	17x4	1x1	142x27	142x27
Ratio to input	/1	/8	/16	/32	–	/4	/4

Global Coarse-Scale Network(1/2)



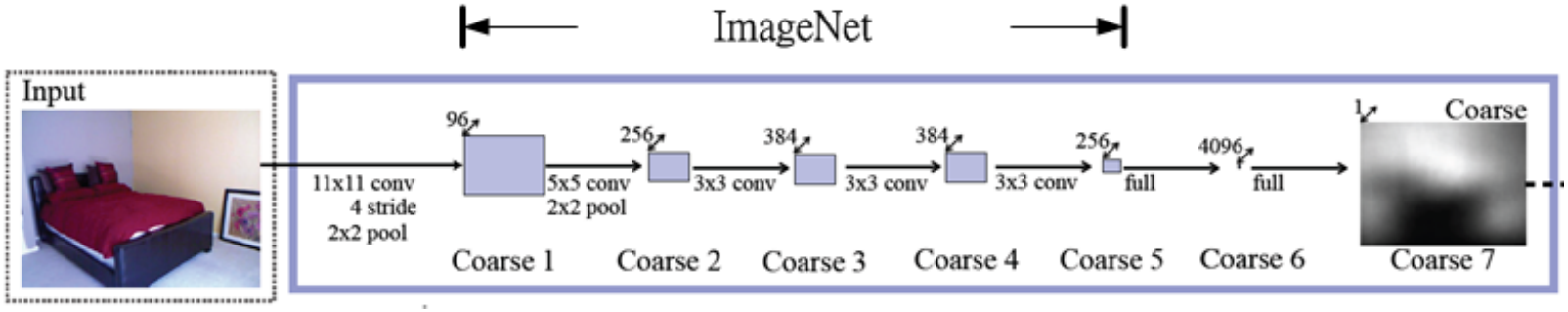


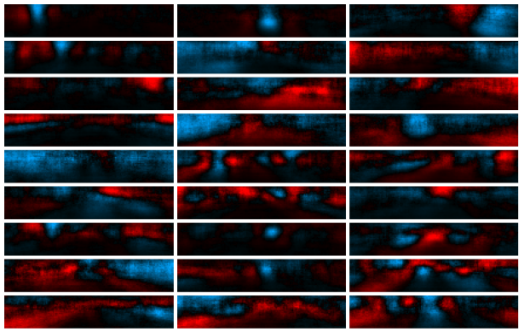


- Red: Positive(farther)
- Blue: Negative(closer)
- Record spatial features

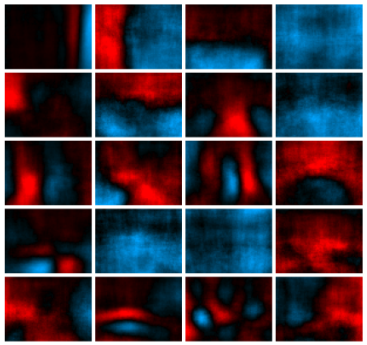
UCDAVIS

Global Coarse-Scale Network(2/2)

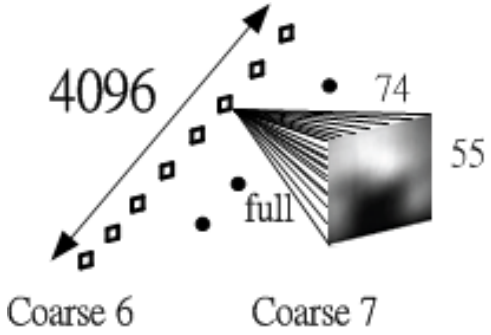




(a) KITTI

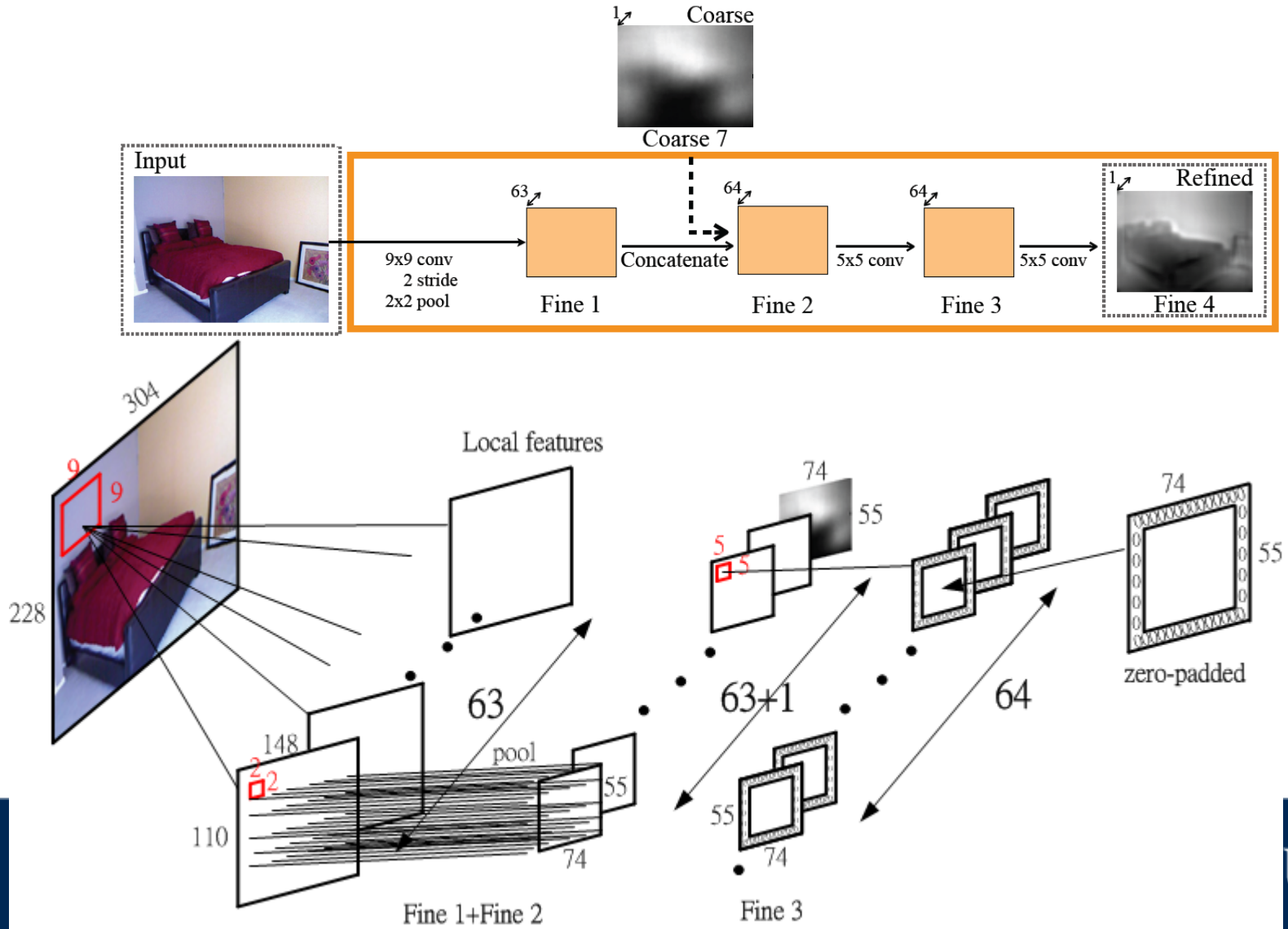


(b) NYUDepth



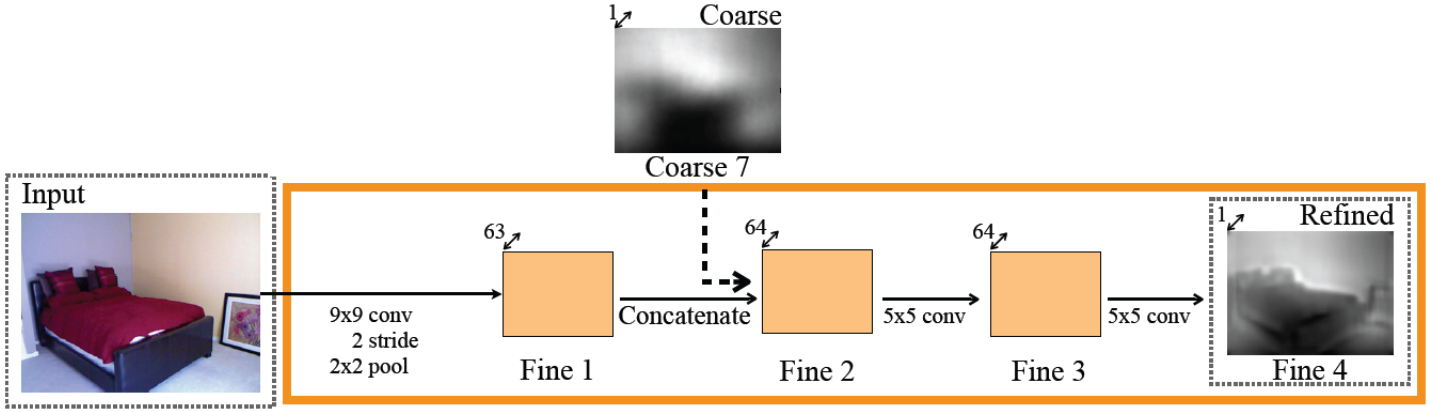
- All hidden layers: rectified linear units
- Layer 7 is linear(Softmax)
- Dropout: fully connected hidden layer 6

Local Fine-Scale Network(1/2)



UCDAVIS

Local Fine-Scale Network(2/2)

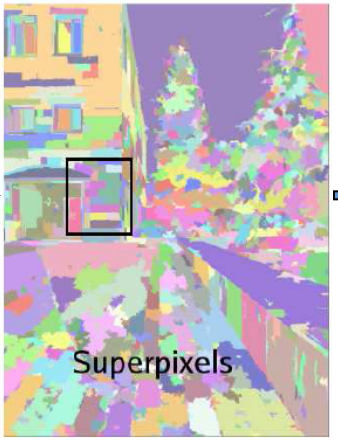


- Convolutional layers only
- All hidden layers: rectified linear units
- Fine 4 is linear
- Keep coarse-scale output fixed

Previous 2D->3D work: Make3D(1/8)

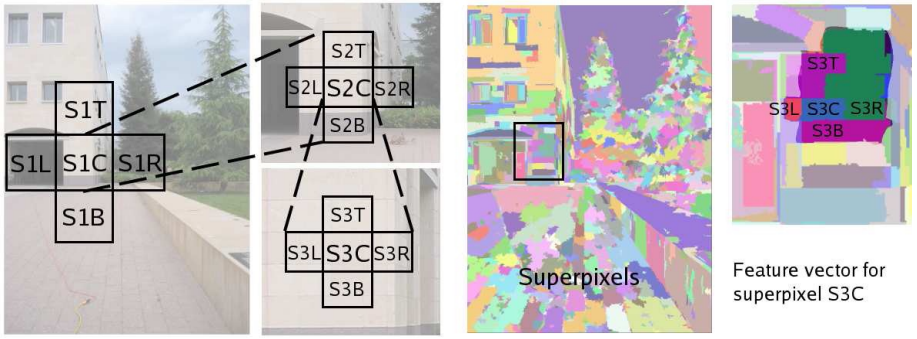
- Using superpixels to find absolute depth.
- Group into small homogenous regions by filters
- Find 3D location and orientation of these patch

Image



Superpixels

Filters



Feature vector for superpixel S3C

Appended at multiple scales and with neighbors.

[16] A. Saxena. TPAMI, 2008.

Cite from Ashutosh Saxena

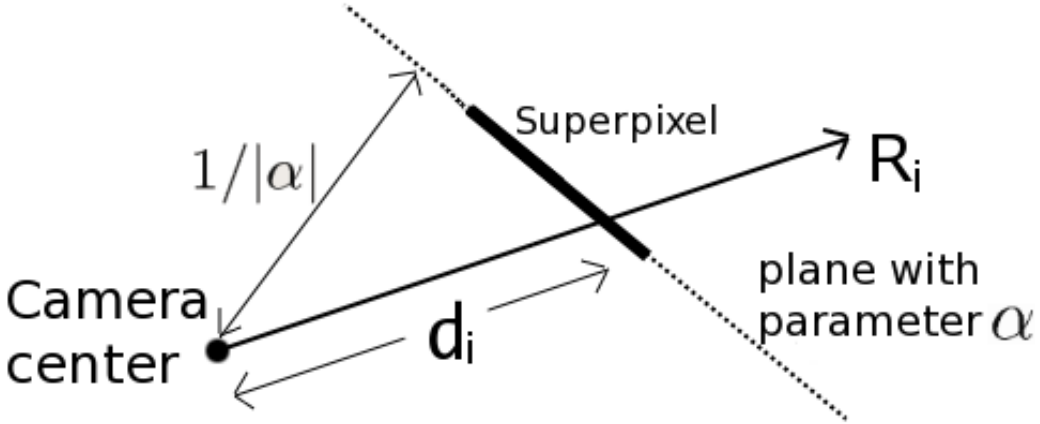
Markov Random Field (MRF) (2/8)

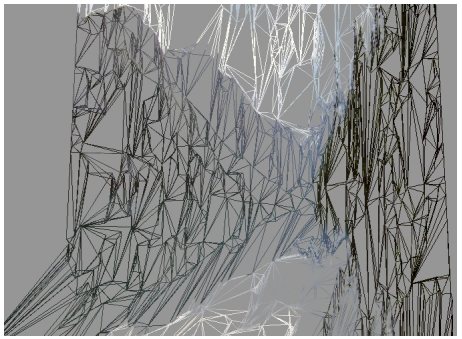
- Xi, Xj are superpixels
- MRF models the relations (shown by the edge Xij) between neighboring superpixels. This defines whether Xi and Xj are on the same plane.

Cite from Ashutosh Saxena

Plane Parameters (3/8)

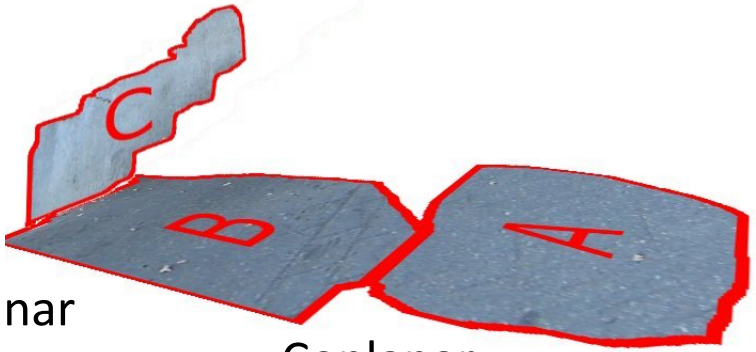
- View each superpixel is a plane
- Plane parameters $\alpha \in \mathbb{R}^3$
  (location, depth and orientation)
- Rays **R**, depth $d_i = 1/R_i^T \alpha$
- Want to model $P(\alpha \mid x; \theta)$; θ is MRF model





Coplanarity and Connectivity (4/8)

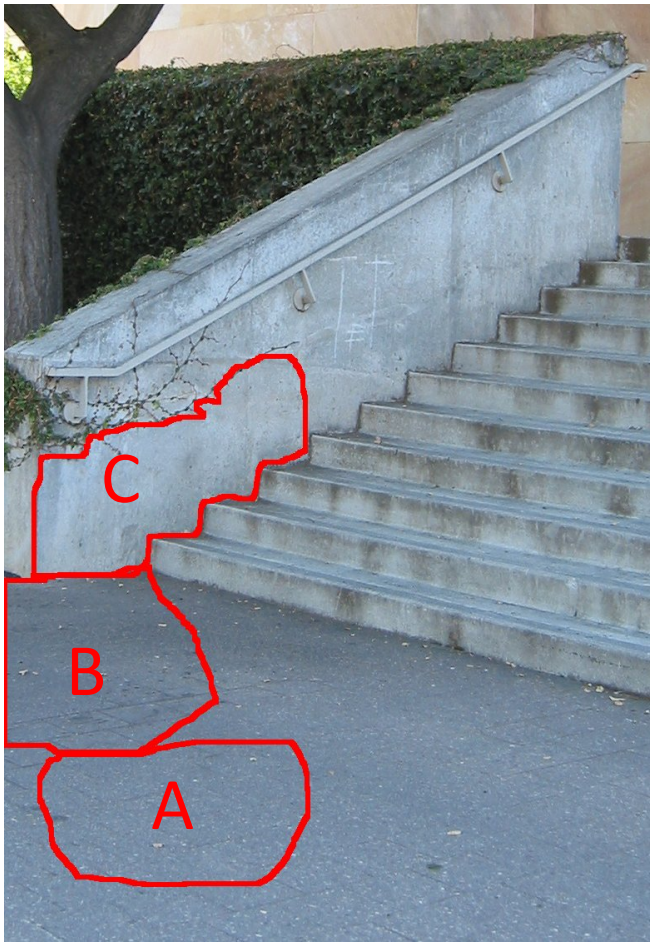
α_C is different plane



Not Coplanar

Coplanar

α_B and α_A are same plane



Cite from Ashutosh Saxena

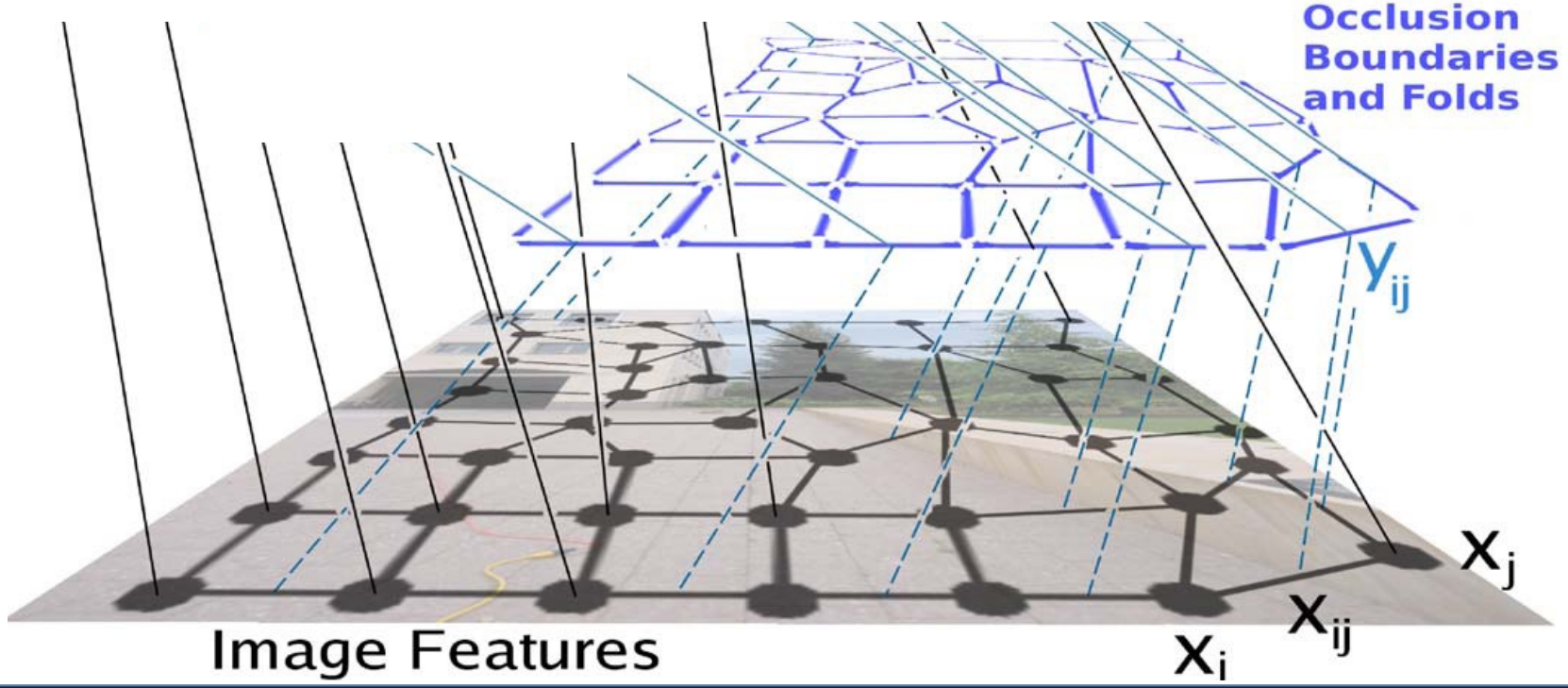
Occlusion Boundary / Fold (5/8)

Image

y: occlusion boundary or fold.
yij=0 indicates an boundary(black)

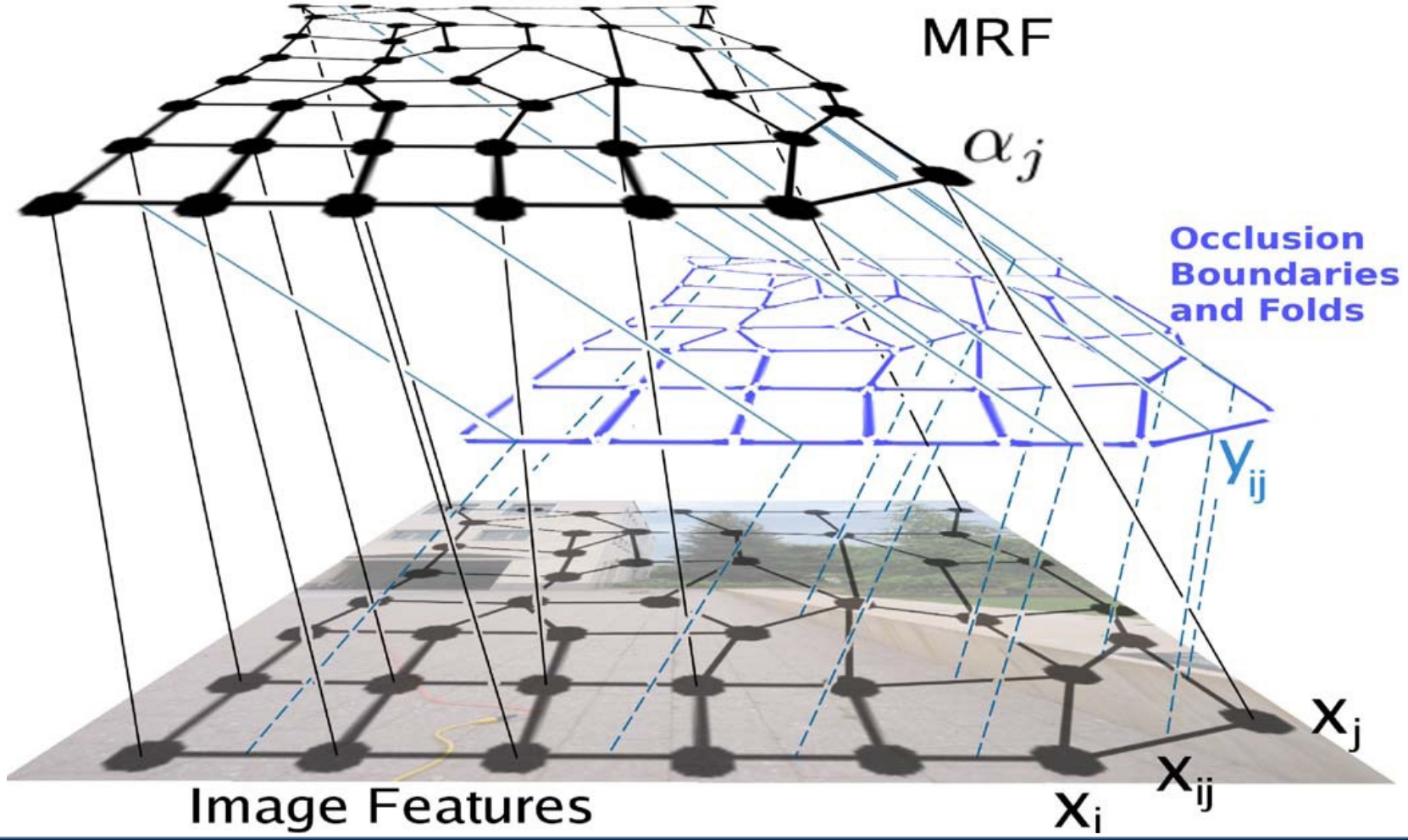
- Learn occlusion boundary/folds using features
- As 3D-model with true edges for MRF
- Learn Xij by yij

MRF Model (6/8)



Cite from Ashutosh Saxena

MRF Model (7/8)



Cite from Ashutosh Saxena

MRF Model (8/8)

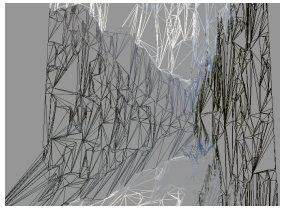
$$P(\alpha|X,\nu,y,R;\theta) = \frac{1}{Z}\prod_i f_1(\alpha_i|X_i,\nu_i,R_i;\theta)$$

feature depth

$$\prod_{i,j} f_2(\alpha_i,\alpha_j|y_{ij},R_i,R_j)$$

Coplaner or not

α: Plane parameters
X: Image features
y: Occlusion boundary/fold
R: Rays from the camera
v: Confidence in features



Depths / Plane Parameters

Occlusion boundaries and folds

Image Features

$$\alpha^* = argmax_\alpha \log P(\alpha|X,v,y,R;\theta_r)$$

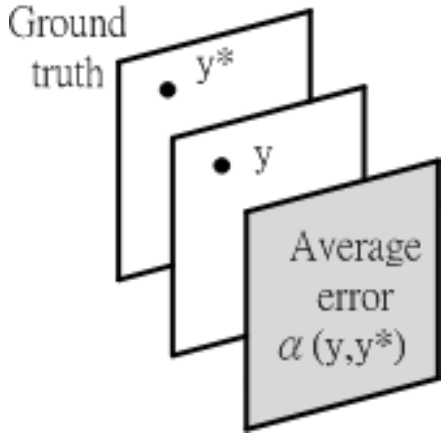
Cite from Ashutosh Saxena

Scale-Invariant Error

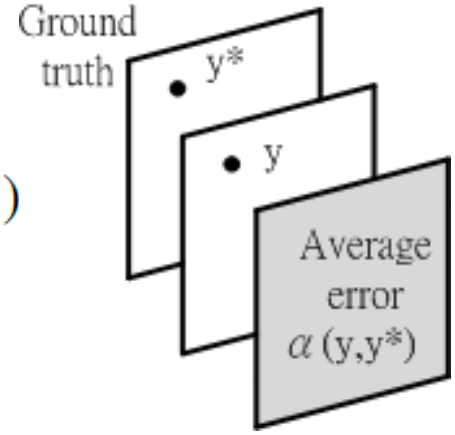
- Make3D uses elementwise to get each plane but average error is still high(0.41 and 0.33 error on RMSE).
- Not only minimize distance between y and y* in each pixels, but also minimize average error at the same time.

$$D(y, y^*) = \frac{1}{2n}\sum_{i=1}^{n}(\log y_i - \log y_i^* + \alpha(y, y^*))^2, \quad (1)$$

where $\alpha(y, y^*) = \frac{1}{n}\sum_i(\log y_i^* - \log y_i)$



Scale-Invariant Error



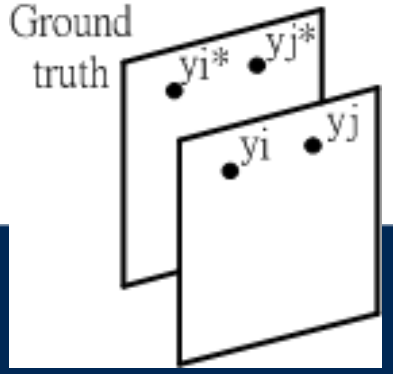
$$D(y, y^*) = \frac{1}{2n}\sum_{i=1}^{n}(\log y_i - \log y_i^* + \alpha(y, y^*))^2, \quad (1)$$

where $\alpha(y, y^*) = \frac{1}{n}\sum_i(\log y_i^* - \log y_i)$

$$D(y, y^*) = \frac{1}{2n^2}\sum_{i,j}\left((\log y_i - \log y_j) - (\log y_i^* - \log y_j^*)\right)^2 \quad (2)$$

$$= \frac{1}{n}\sum_i d_i^2 - \frac{1}{n^2}\sum_{i,j} d_i d_j = \frac{1}{n}\sum_i d_i^2 - \frac{1}{n^2}\left(\sum_i d_i\right)^2 \quad (3)$$

where $d_i = \log y_i - \log y_i^*$



UCDAVIS

Scale-Invariant Error

- Not only decrease error pixelwise but also error between other pixels.
- Every pixels share the same error. For any prediction y, e $^{\alpha}$ is the scale that best aligns it to the ground truth. All scalar multiples of y have the same error, hence the scale invariance.

$$D(y, y^*) = \frac{1}{n}\sum_i d_i^2 - \frac{1}{n^2}\sum_{i,j} d_i d_j = \frac{1}{n}\sum_i d_i^2 - \frac{1}{n^2}\left(\sum_i d_i\right)^2 \quad (3)$$

L2 error

Credits mistake if they are predict wrong in the same direction.

Experiments

- Loss function: $L(y, y^*) = \frac{1}{n}\sum_i d_i^2 - \frac{\lambda}{n^2}\left(\sum_i d_i\right)^2$
- Data Augmentation: scale, rotation, translation, color, flips

- **NYU Depth v2** (By Microsoft Kinect camera)
  - 249 scenes for training, 215 for testing,
  - Coarse network 2M using SGD with batches of size 32,
  - Fine network for 1.5M samples.
- **KITTI** (By extra LIDAR scanner)
  - 28 scenes for training, 28 scenes for testing,
  - Coarse network 1.5M, fine network for 1M samples.

Experiments

Threshold: % of y_i s.t. $\max(\frac{y_i}{y_i^*}, \frac{y_i^*}{y_i}) = \delta < thr$

Abs Relative difference: $\frac{1}{|T|}\sum_{y\in T}|y - y^*|/y^*$

Squared Relative difference: $\frac{1}{|T|}\sum_{y\in T}||y - y^*||^2/y^*$

RMSE (linear): $\sqrt{\frac{1}{|T|}\sum_{y\in T}||y_i - y_i^*||^2}$

RMSE (log): $\sqrt{\frac{1}{|T|}\sum_{y\in T}||\log y_i - \log y_i^*||^2}$

RMSE (log, scale-invariant): The error Eqn. 1

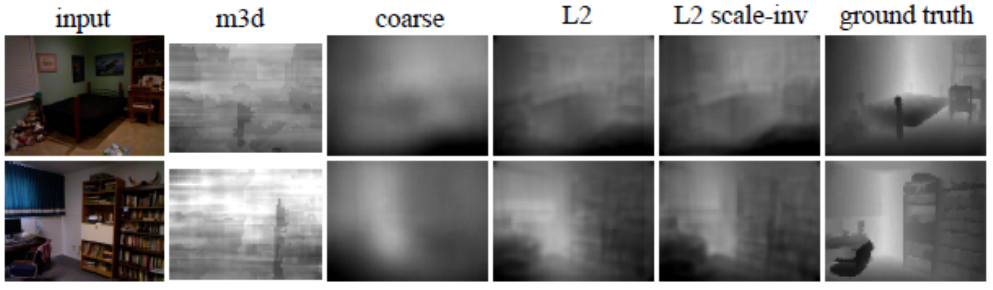
	Mean	Make3D	Ladicky*&al*	Karsch*&al*	Coarse	Coarse + Fine	
threshold $\delta < 1.25$	0.418	0.447	0.542	–	**0.618**	0.611	higher
threshold $\delta < 1.25^2$	0.711	0.745	0.829	–	**0.891**	0.887	is
threshold $\delta < 1.25^3$	0.874	0.897	0.940	–	0.969	**0.971**	better
abs relative difference	0.408	0.349	–	0.350	0.228	**0.215**	
sqr relative difference	0.581	0.492	–	–	0.223	**0.212**	lower
RMSE (linear)	1.244	1.214	–	1.2	**0.871**	0.907	is
RMSE (log)	0.430	0.409	–	–	**0.283**	0.285	better
RMSE (log, scale inv.)	0.304	0.325	–	–	0.221	**0.219**	

Table 1: Comparison on the NYUDepth dataset

Experiments

	Mean	Make3D	Coarse	Coarse + Fine	
threshold $\delta < 1.25$	0.556	0.601	0.679	**0.692**	higher
threshold $\delta < 1.25^2$	0.752	0.820	0.897	**0.899**	is
threshold $\delta < 1.25^3$	0.870	0.926	**0.967**	**0.967**	better
abs relative difference	0.412	0.280	0.194	**0.190**	
sqr relative difference	5.712	3.012	1.531	**1.515**	lower
RMSE (linear)	9.635	8.734	7.216	**7.156**	is
RMSE (log)	0.444	0.361	0.273	**0.270**	better
RMSE (log, scale inv.)	0.359	0.327	0.248	**0.246**	

Table 2: Comparison on the KITTI dataset.



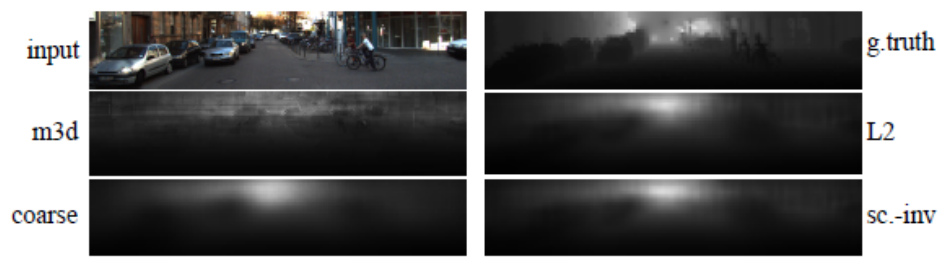
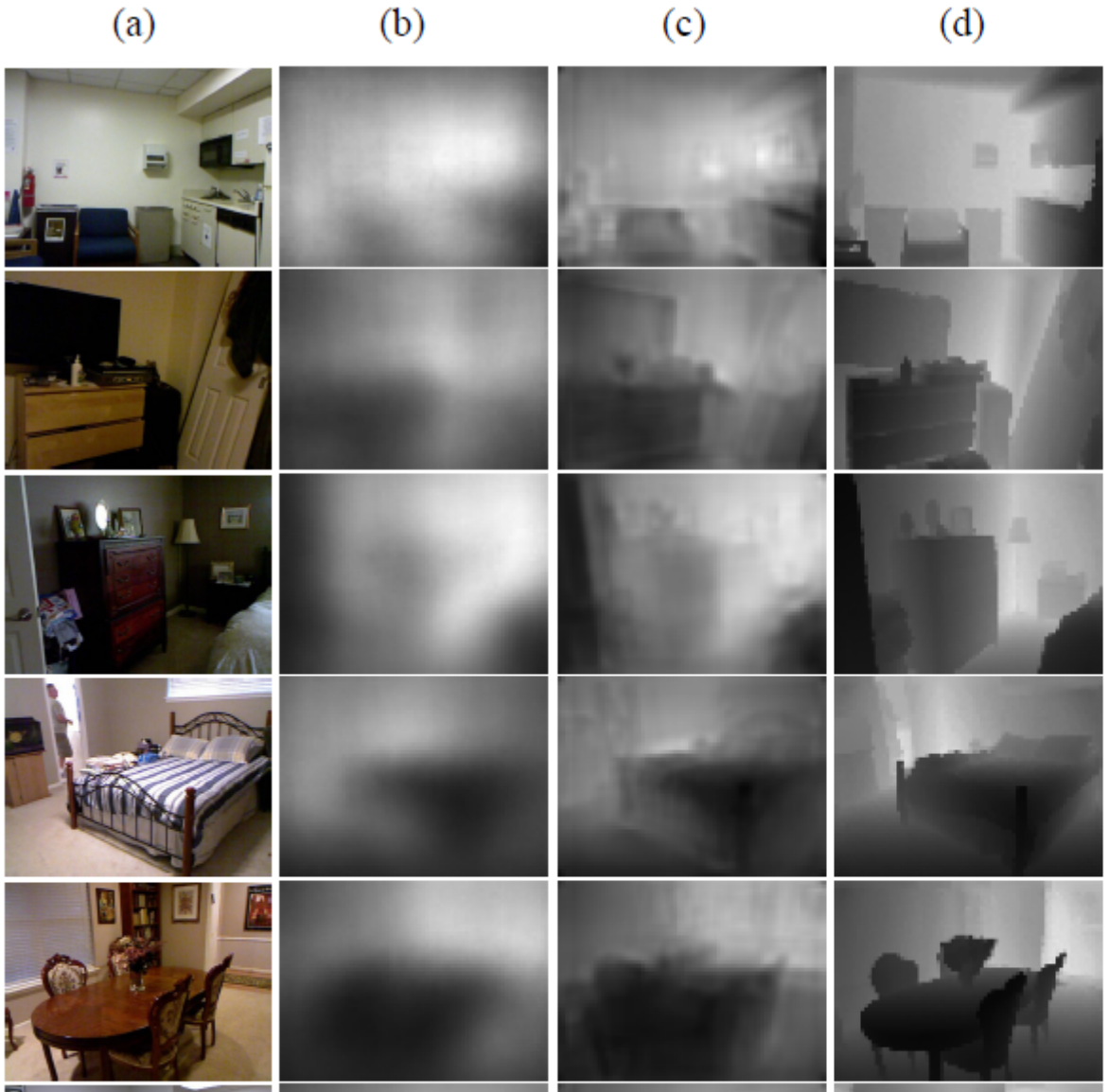
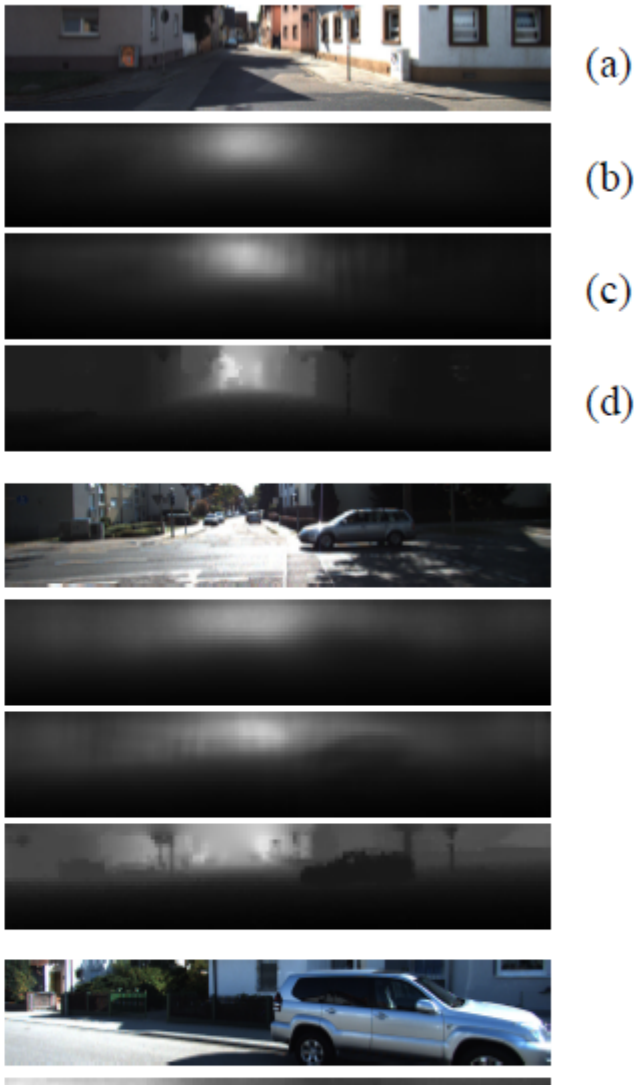


Figure 3: Qualitative comparison of Make3D, our method trained with l_2 loss ($\lambda = 0$), and our method trained with both l_2 and scale-invariant loss ($\lambda = 0.5$).

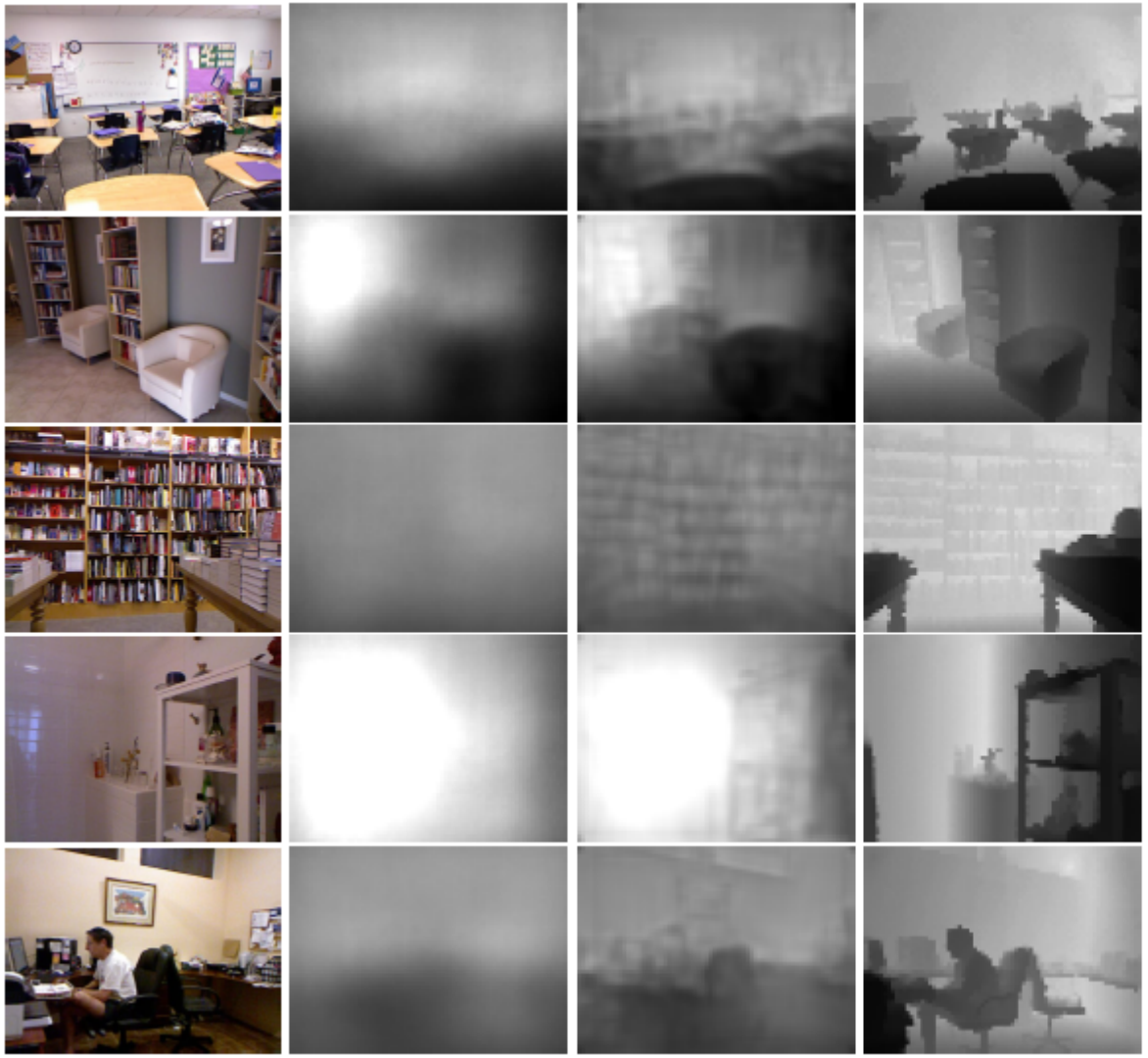
Experiments

(a) input, (b) output of coarse network,
(c) refined output of fine network, (d) ground truth.

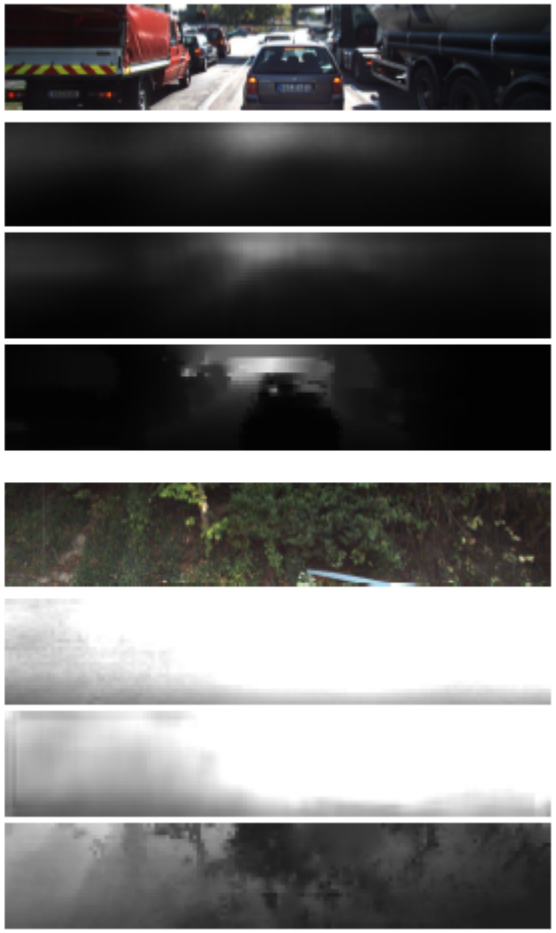




Experiments



- uncorrected alignment issues between the depth map and input in the training data



Conclusion

- Coarse network -> global depth
- Fine network -> local depth
- Scale-invariant error -> reduce relevant error