Learning and Inferring Depth from Monocular Images

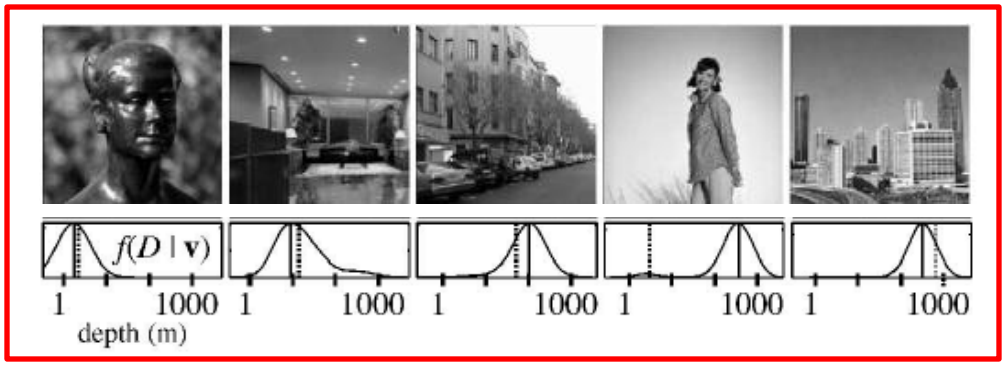
Jiyan Pan

April 1, 2009

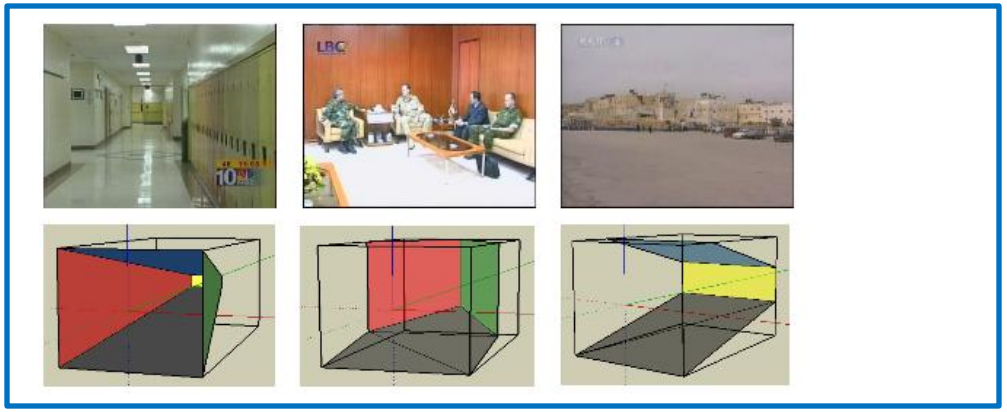
Problem

- Traditional ways of inferring depth
  - Binocular disparity
  - Structure from motion
  - Defocus
- Given a single monocular image, how to infer absolute depth?
  - Learn the relationship between image features and absolute depth

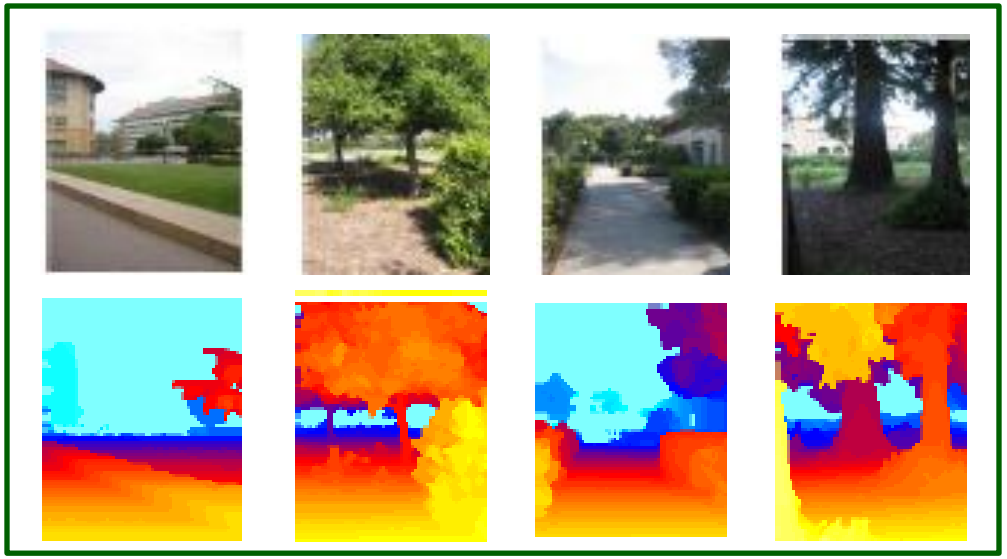
coarse



Mean depth
(Paper #1)



Geometric stage
(Paper #2)



Dense depth map
(Paper #3)

fine

Paper #1

- A. Torralba, A. Oliva., “Depth estimation from image structure,” PAMI, 24(9): 1-13, 2002

- Goal: estimate absolute *mean* depth of a scene based on scene structure

- This paper inspired the invention of GIST feature

Intuition

- Scenes at different depths have different spatial structures
  - Panoramic views:
    Uniform texture zones along horizontal layers
  - Mid-range urban environments:
    Dominant long horizontal and vertical contours and square patterns
  - Close-up views of objects:
    Large flat surfaces with no dominant orientation

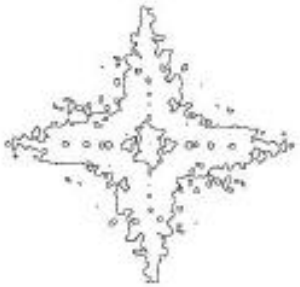
Image Structure Representation

- Global spectral signature
  - Defined as the amplitude spectrum of the entire image

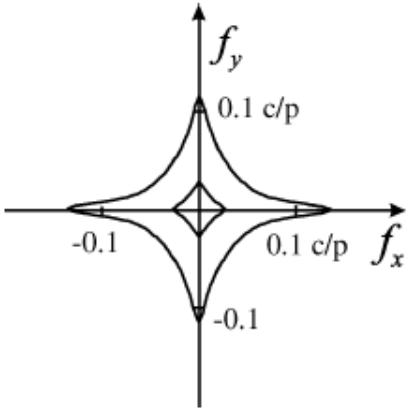
$$I(\mathbf{f}) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} i(\mathbf{x})\, h(\mathbf{x})\, e^{-j2\pi<\mathbf{f},\mathbf{x}>}, \quad A(\mathbf{f}) = |I(\mathbf{f})|$$

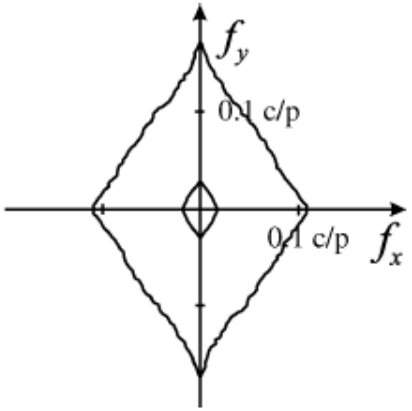
  - Man-made and natural scenes have disparate global spectral signatures

Specific examples:



On average:





Man-made

Natural

Image Structure Representation

- Local spectral signature
  - Defined as the magnitude of the output of Gabor wavelet filters

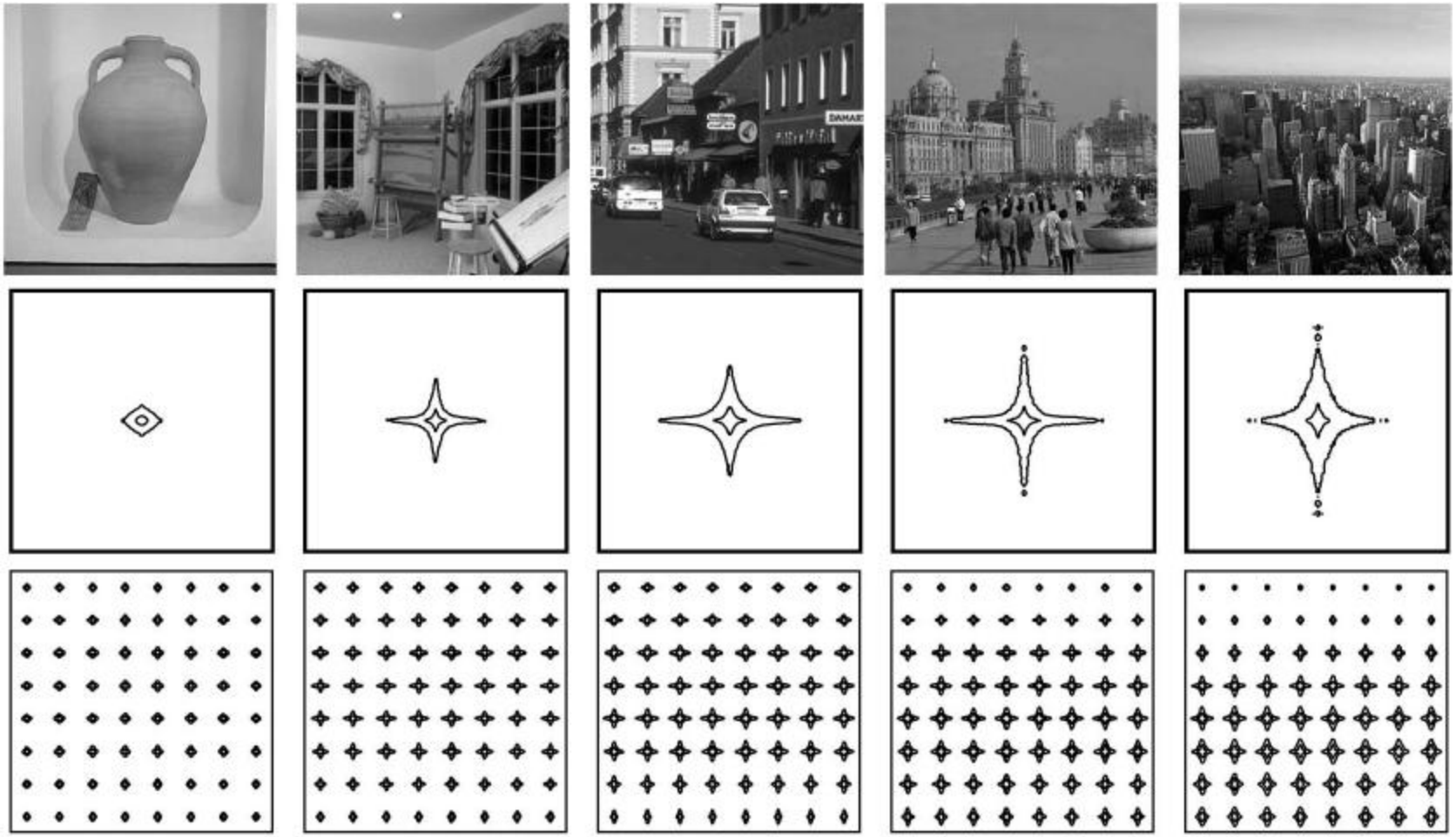
$$I(\mathbf{x}, k) = \sum_{x'} i(\mathbf{x}')\ h_k(\mathbf{x} - \mathbf{x}'), \qquad A(\mathbf{x}, k) = |I(\mathbf{x}, k)|$$

  - Not only differs between man-made and natural scenes, but is spatially non-stationary as well

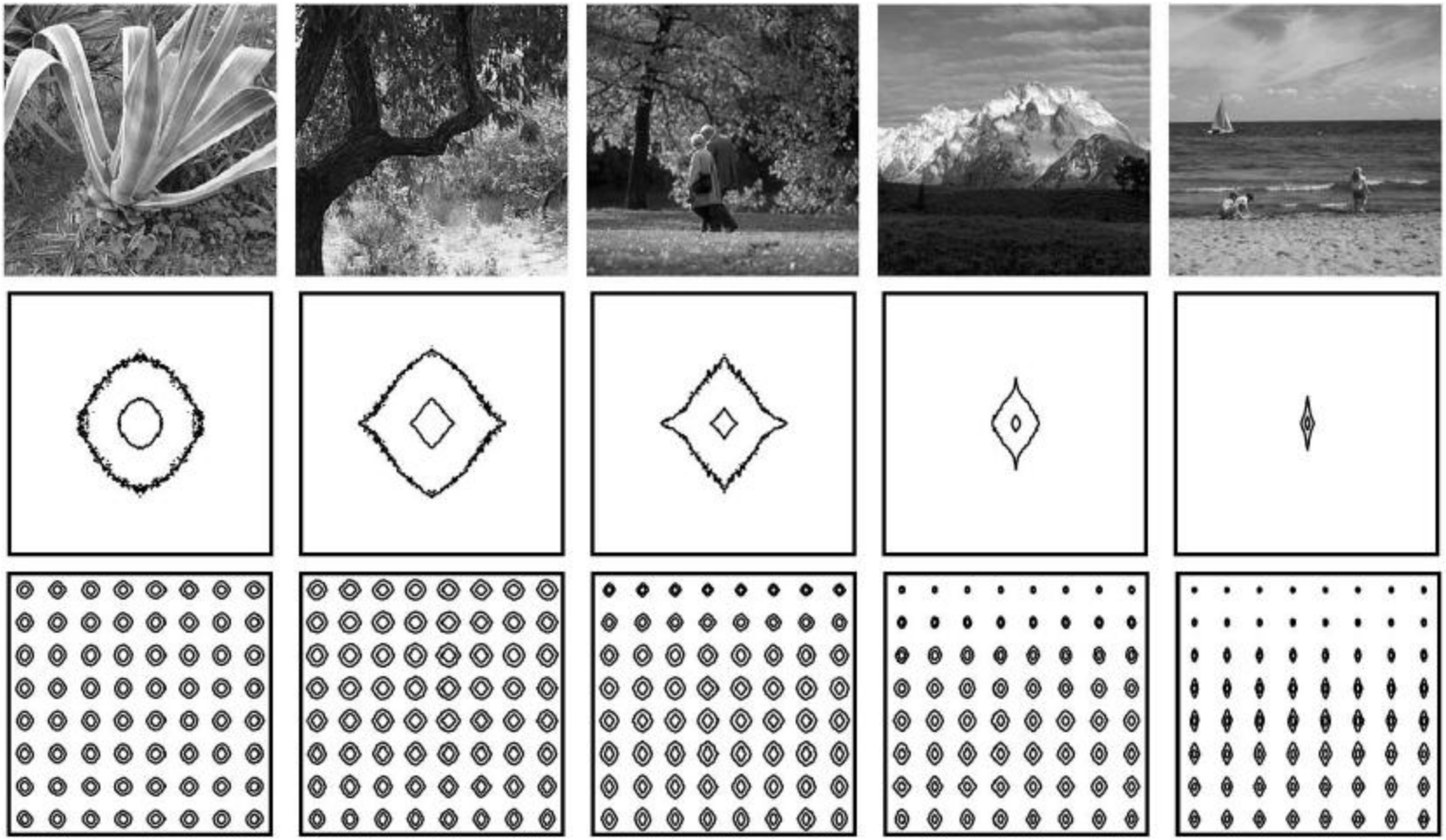
Spectral Signature vs. Mean Depth

- How do the global and local spectral signatures change with the mean depth of images?
  - The answer to this question determines if it is feasible to estimate mean depth using those spectral signatures

Man-made



Natural



Spectral Signature vs. Mean Depth

- Typical behavior of spectral signatures
  - An increase of global roughness w.r.t. depth for man-made structures
  - A decrease of global roughness w.r.t. depth for natural structures
  - Local spectral signature becomes increasingly non-stationary as depth gets large
  - More biased towards horizontal and vertical orientations as depth increases

Low-Dimensional Representation of Spectral Signature

- Global spectral signature
  - Amplitude spectrum is defined on a continuous 2-D frequency space
  - Discretize it by sampling at specific frequency magnitudes and orientations
  - Approximated by summing the energy of wavelet coefficients over the entire image at specific scales and orientations

$$A_k^2 = \sum_{\mathbf{x}} |I(\mathbf{x}, k)|^2$$

  - Termed as global energy, which encodes dominant orientations and scales in the image

Low-Dimensional Representation of Spectral Signature

- Global spectral signature (another variant)
  - Wavelet coefficients at different scales and orientations are correlated
  - Define the magnitude correlation

$$A_{i,j}^2 = \sum_{\mathbf{x}} |I(\mathbf{x}, i)||I(\mathbf{x}, j)|$$

  - Magnitude correlations encode degree of clutter of edges and shapes

Low-Dimensional Representation of Spectral Signature

- Local spectral signature
  - The original local spectral signature has the same spatial resolution as the image
  - Downsample it to contain only M^2 pixels

$$A_M^2(\mathbf{x}, k) = \left\{ |I(\mathbf{x}, k)|^2 \downarrow M \right\}$$

  - Termed as local energy, which encodes local scales and orientations

Low-Dimensional Representation of Spectral Signature

- Dimension of global energy: K
- Dimension of magnitude correlations: K^2
- Dimension of local energy: M^2K

- Perform PCA to reduce the feature dimension to L

Learning to Estimate Mean Depth from Feature Vector

- Regress the mean depth D on feature vector $\mathbf{v}$
- Need to approximate the regression function

$$E[D \mid \mathbf{v}] = \int_{-\infty}^{\infty} D\, f_{D|v}(D \mid \mathbf{v})\, dD$$

- Take a generative approach:
  - Given a feature vector $\mathbf{v}$, use Gaussian Bayes classifier to determine if the image belongs to man-made scene group or natural scene group

Learning to Estimate Mean Depth from Feature Vector

– Given each scene group, model the joint distribution of D and $\mathbf{v}$ as a two-level hierarchical mixture of Gaussians

$$f(D, \mathbf{v} \mid art) = \sum_{i=1}^{N_c} g(D \mid \mathbf{v}, c_i)\, g(\mathbf{v} \mid c_i) p(c_i)$$

where

$$g(\mathbf{v} \mid c_i) = \frac{\exp\left[-\frac{1}{2}(\mathbf{v} - \mu_i)^T \mathbf{X}_i^{-1} (\mathbf{v} - \mu_i)\right]}{(2\pi)^{L/2} |\mathbf{X}_i|^{1/2}}$$

$$g(D \mid \mathbf{v}, c_i) = \frac{\exp\left[-\left(D - a_i - \mathbf{v}^T \vec{b}_i\right)^2 / 2\sigma_i^2\right]}{\sqrt{2\pi}\sigma_i}$$

Learning to Estimate Mean Depth from Feature Vector

- Note that the mean of D given $\mathbf{v}$ and cluster is a linear function of $\mathbf{v}$, suggesting that the ML estimation of a_i and $\vec{b}_i$ is equivalent to LS linear regression

Learning to Estimate Mean Depth from Feature Vector

- Use EM algorithm to learn the mixture model
  - E-step
  - M-step

$$p^k(c_i \mid D_t, \mathbf{v}_t) = \frac{g^k(D_t \mid \mathbf{v}_t, c_i)\, g^k(\mathbf{v}_t \mid c_i)\, p^k(c_i)}{\sum_{i=1}^{N_c} g^k(D_t \mid \mathbf{v}_t, c_i)\, g^k(\mathbf{v}_t \mid c_i)\, p^k(c_i)}$$

$$p^{k+1}(c_i) = \frac{\sum_{t=1}^{N_t} p^k(c_i \mid D_t, \mathbf{v}_t)}{\sum_{i=1}^{N_c} \sum_{t=1}^{N_t} p^k(c_i \mid D_t, \mathbf{v}_t)}.$$

$$\mu_i^{k+1} = < \mathbf{v} >_i = \frac{\sum_{t=1}^{N_t} p^k(c_i \mid D_t, \mathbf{v}_t)\, \vec{v}_t}{\sum_{t=1}^{N_t} p^k(c_i \mid D_t, \mathbf{v}_t)},$$

$$\mathbf{X}_i^{k+1} = < (\mathbf{v} - \mu_i^{k+1})(\mathbf{v} - \mu_i^{k+1})^T >_i,$$

$$\mathbf{b}_i^{k+1} = \left(\mathbf{X}_i^{k+1}\right)^{-1} < D\,\mathbf{v} >_i,$$

$$a_i^{k+1} = < D - \mathbf{v}^T \mathbf{b}_i^{k+1} >_i,$$

$$\sigma_i^{k+1} = < (D - a_i^{k+1} - \mathbf{v}^T \mathbf{b}_i^{k+1})^2 >_i .$$

Learning to Estimate Mean Depth from Feature Vector

- Having learned the model,
  - The estimate of D is given by a mixture of linear regressions

$$\hat{D} = \frac{\sum_{i=1}^{N_c} (a_i + \mathbf{v}^T \vec{b_i})\, g(\mathbf{v} \mid c_i)\, p(c_i)}{\sum_{i=1}^{N_c} g(\mathbf{v} \mid c_i)\, p(c_i)}$$

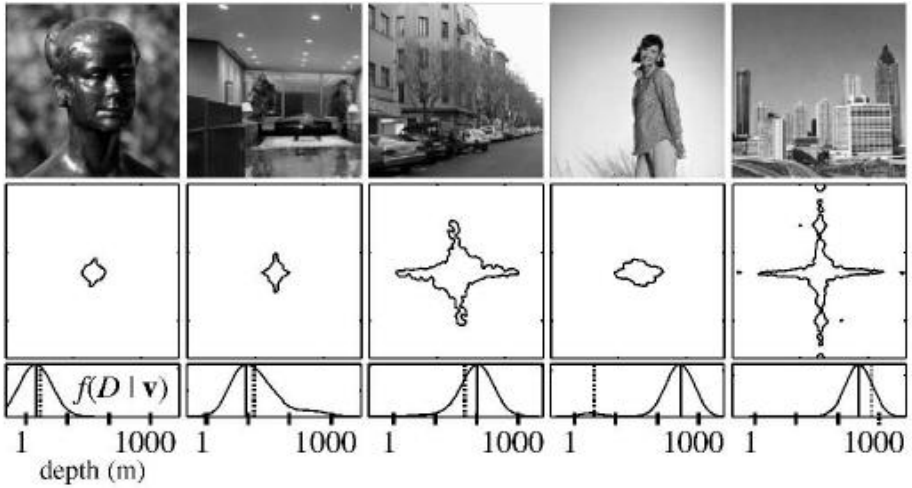
  - The confidence of the estimate is given by

$$\sigma_D^2 = E[(\hat{D} - D)^2 | \mathbf{v}] = \frac{\sum_{i=1}^{N_c} \sigma_i^2\, g(\mathbf{v} \mid c_i)\, p(c_i)}{\sum_{i=1}^{N_c} g(\mathbf{v} \mid c_i)\, p(c_i)}$$

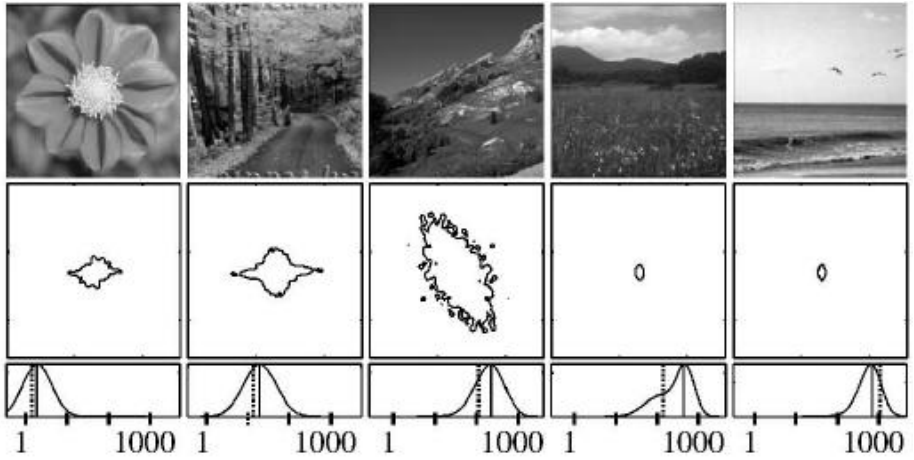
Experimental Results

- Ground truth generation
  - Divide the entire training data set into man-made scene group and natural scene group
  - Sort the images in each group according to their mean depths
  - Humans estimate the mean depths of a portion of the images in each group
  - Use a polynomial to fit the estimated mean depth as a function of the sorted rank

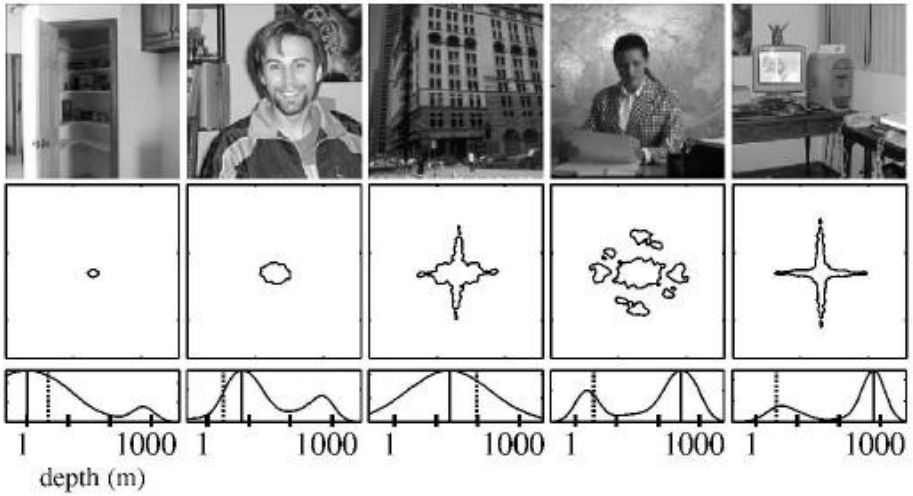
Using global energy features



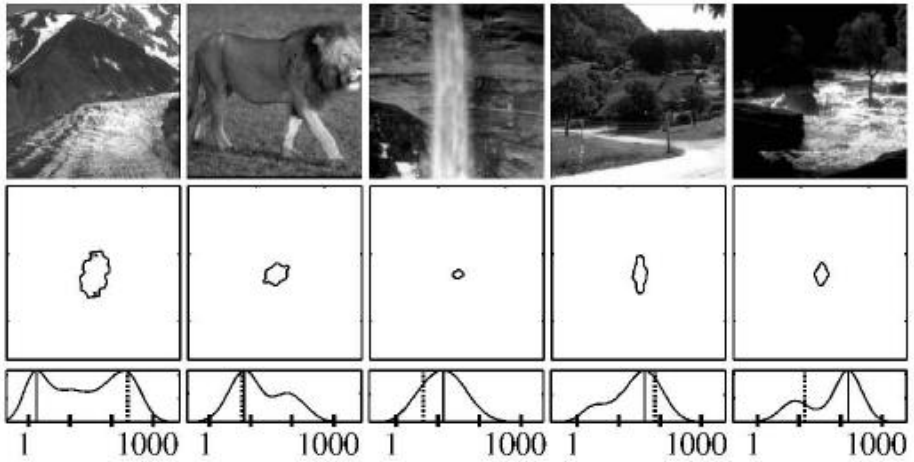
(a)



(b)

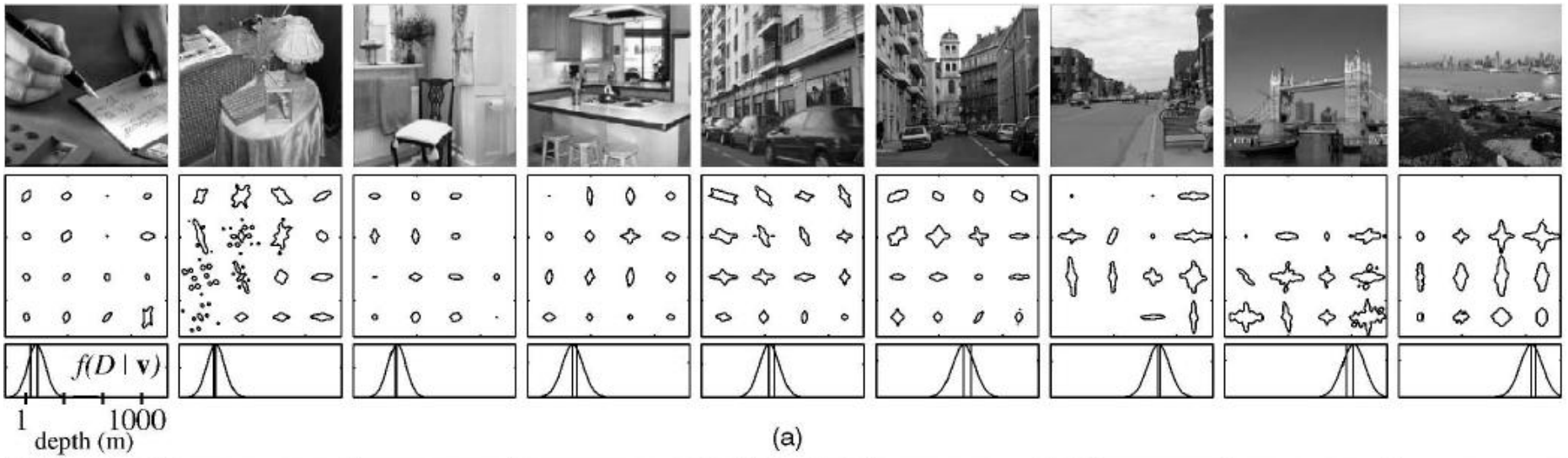


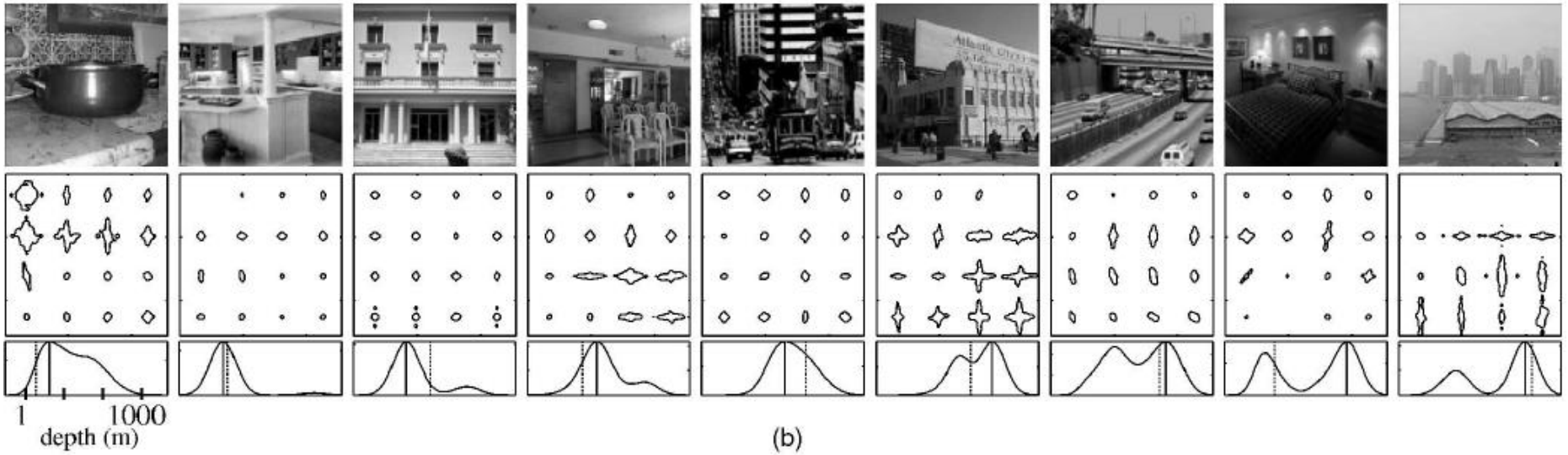
(c)



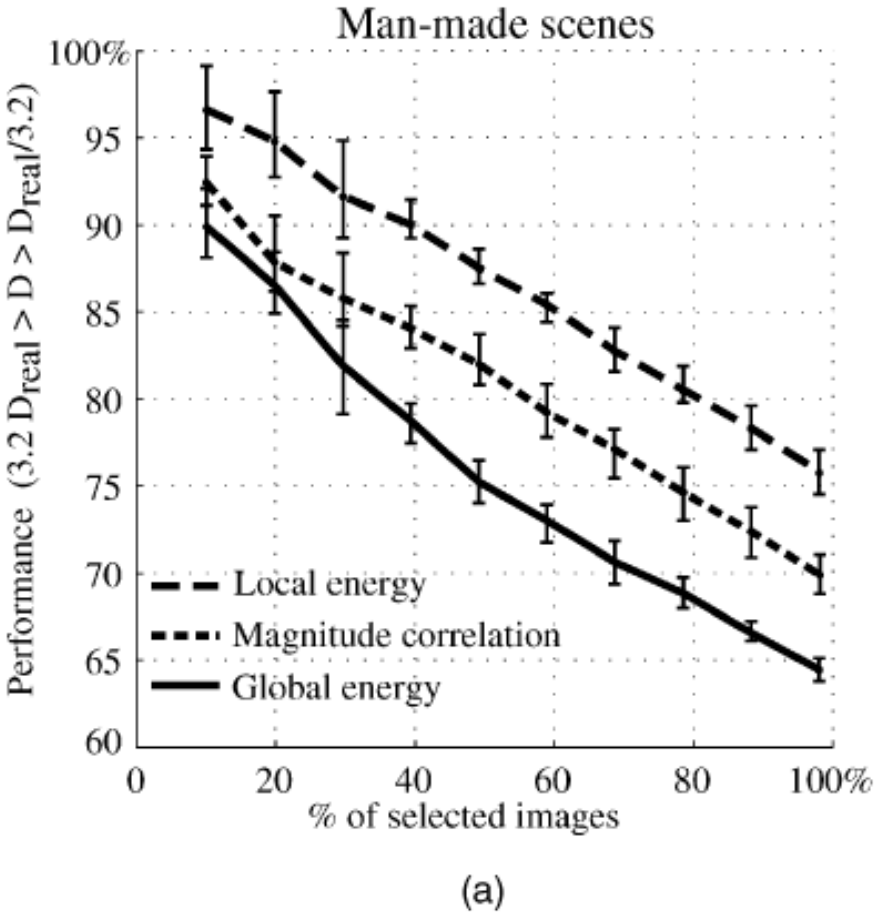
(d)

Using local energy features

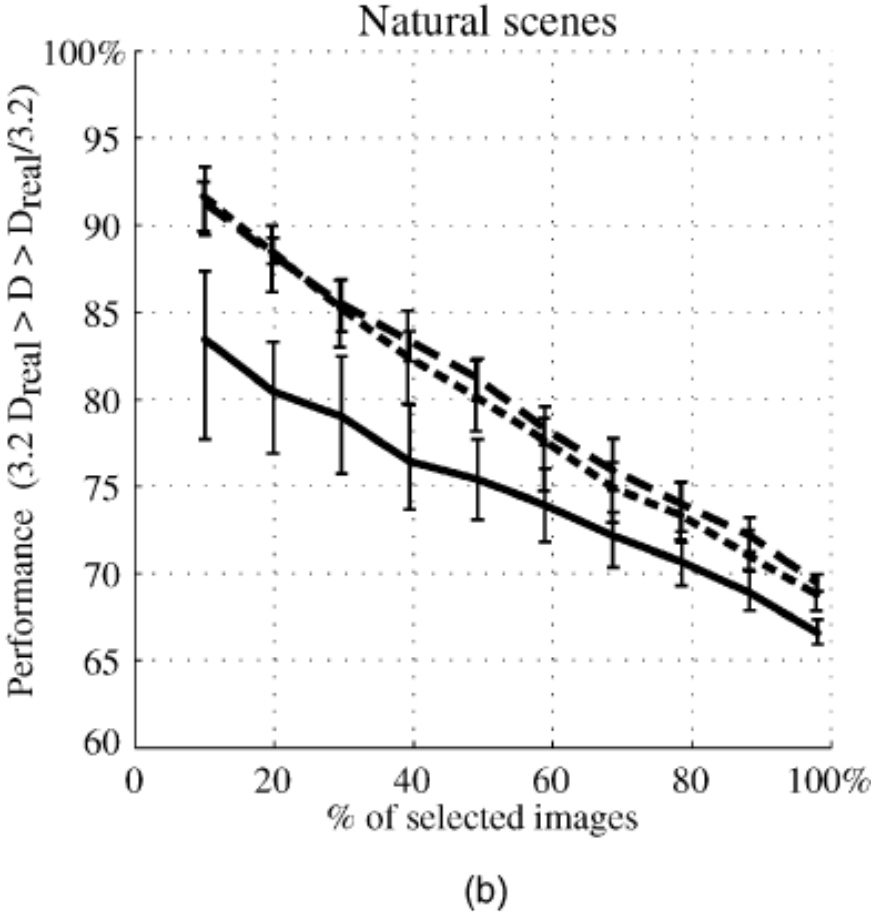




Comparison



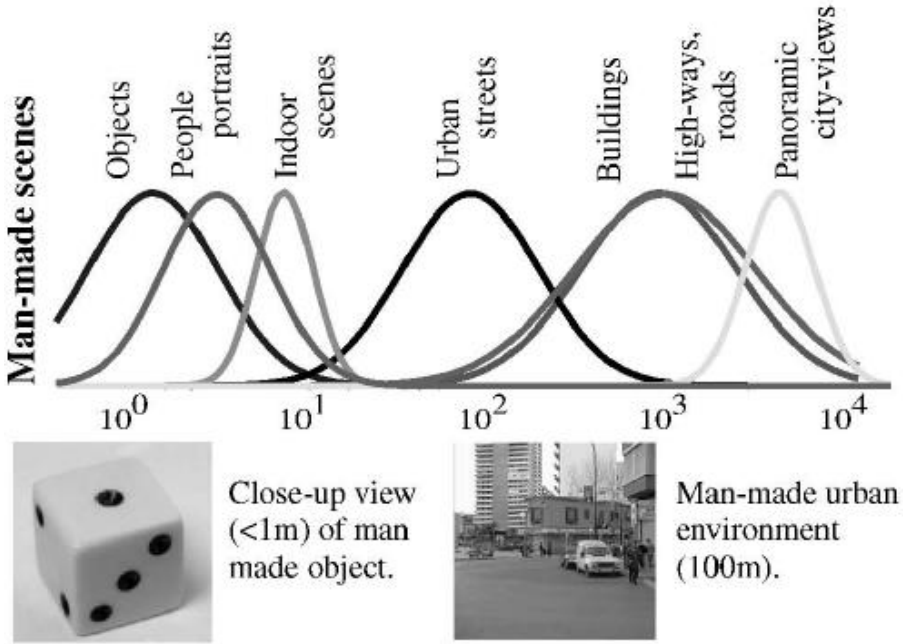
(a)

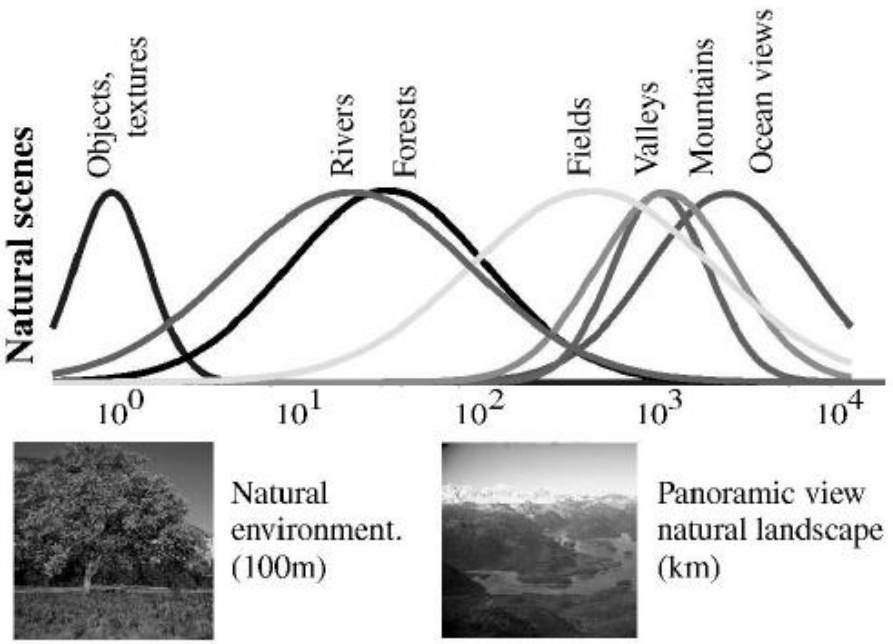


(b)

Application

- Scene category recognition





Application

- Scale selection

Paper #2

- V. Nedovic, A. W.M. Smeulders, A. Redert and J.M. Geusebroek, “Depth Information by Stage Classification,” ICCV, 2007

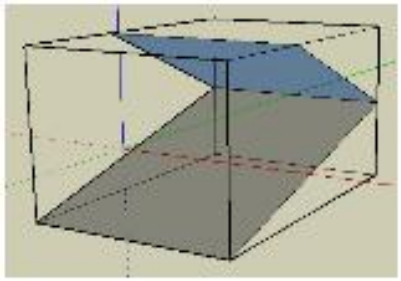
- Goal: Estimate the *geometric type* of the global scene in an image

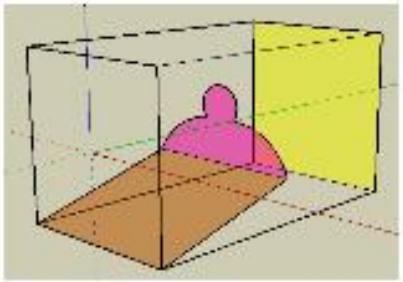
Intuition

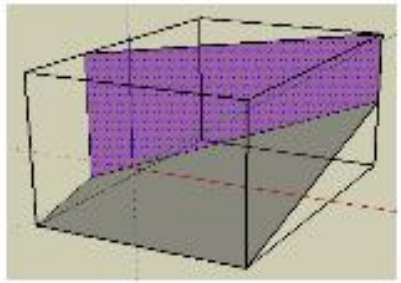
- Instead of directly regressing absolute depth, it may be helpful to classify the image into relatively few 3D scene geometries (stages) first
  - This type of scene information narrows down possible locations, scale, and identities of individual objects


LBC

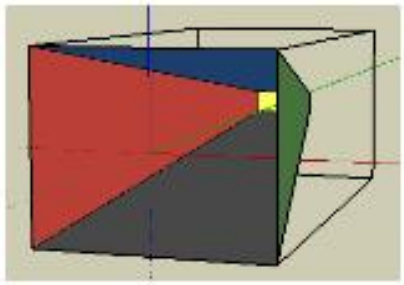

LBC

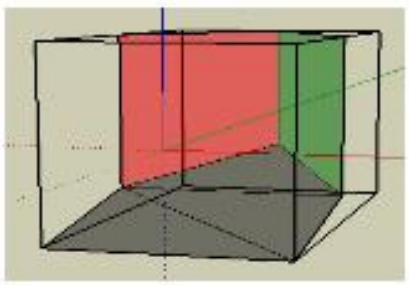


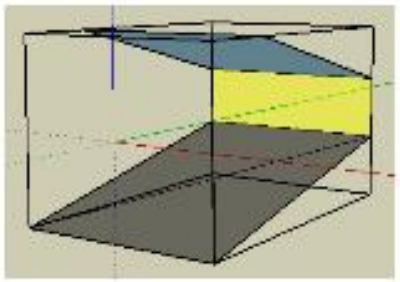





LBC



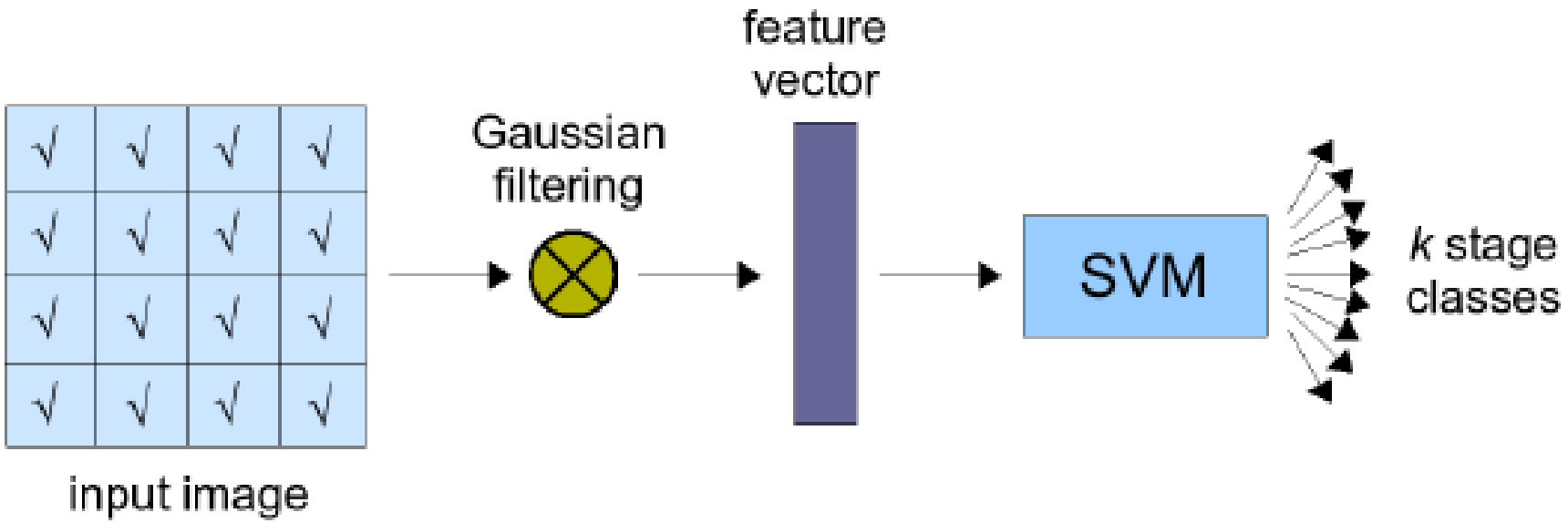




Intuition

- Image gradient has different distributions as depth increases
  - Makes it possible to use image gradient features to perform stage classification

Classification



Experimental Results

- Dataset
  - Keyframes of the 2006 TRECVID video benchmark dataset
  - Annotate 1241 TRECVID keyframes into one of the 15 stage categories
  - For each category, half for training and half for testing

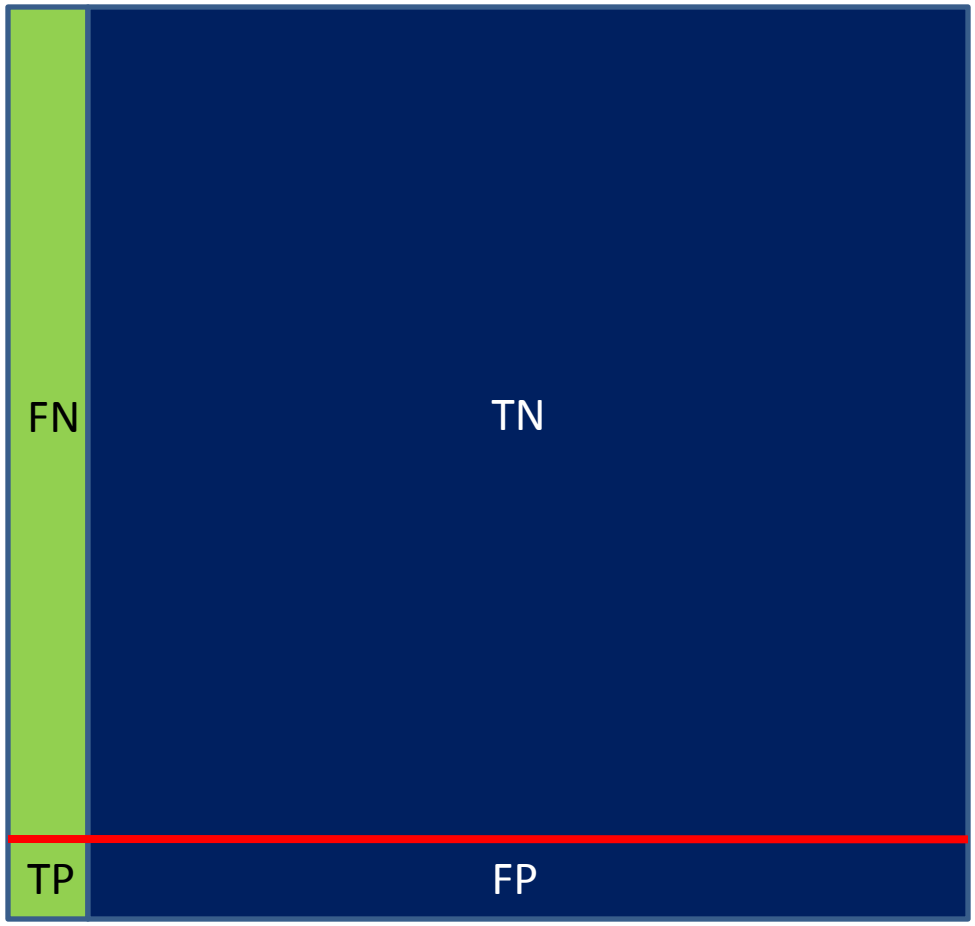
Experimental Results

- Evaluation
  - Correct rate = $\frac{\text{true positives + true negatives}}{\text{total number of images}}$
  - For a class that occupies p% of the dataset, the correct rate of random guess is

$$p/K + (1-p)(K-1)/K,$$

    where K is the total number of classes
  - When p is small, this measure is overly optimistic

Experimental Results



$$\frac{TP + TN}{TP + TN + FP + FN}$$

is pretty large!

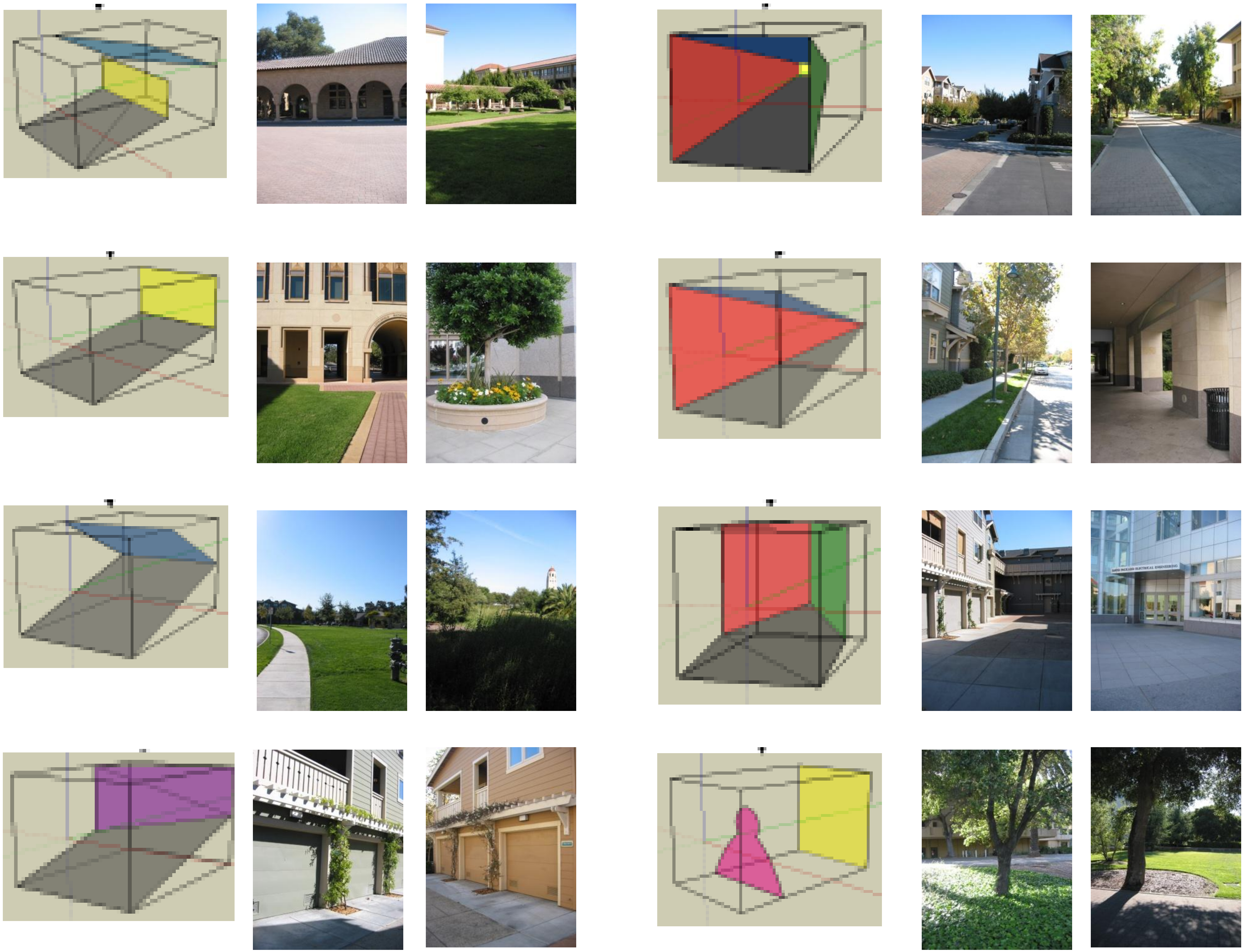
Experimental Results

- Lower level of hierarchy – 12 classes

class	name	% in dataset	% correct	**chance**
1	sky+bkg+gnd	6.3%	16.7%	86.42%
2	gnd+bkg	7.1%	8.2%	85.75%
3	sky+gnd	8.7%	60.7%	84.42%
4	gnd	7.4%	44.7%	85.50%
5	gnd+diagBkg	10.75%	26.9%	82.71%
6	diagBkg	6.4%	14.3%	86.33%
7	box	5.5%	8.1%	87.08%
8	1 side-wall	9%	13.6%	84.17%
9	corner	10.75%	34.3%	82.71%
10	tab+pers+bkg	7.4%	48%	85.50%
11	pers+bkg	13.1%	42.5%	80.75%
12	no depth	7.4%	22.4%	85.50%
			AVG: 28.4%	84.74%

Experimental Results

- My experiments
  - Stanford Range Image Dataset
  - Classify 271 images into 8 stage categories
  - For each stage, 2/3 used for training and 1/3 for testing

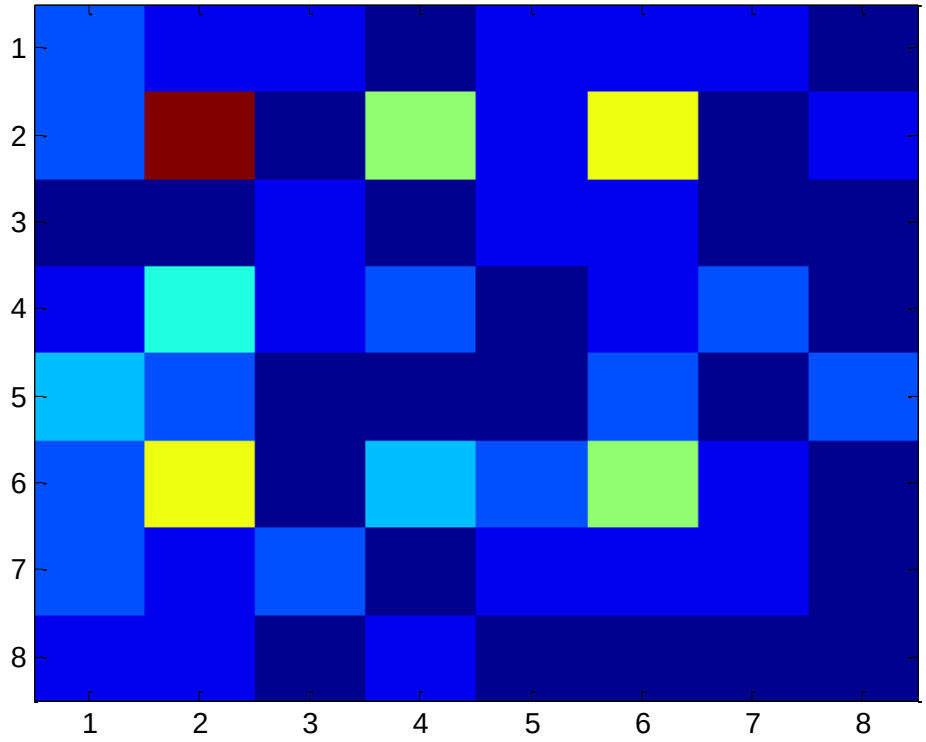


Experimental Results

- Confusion matrix

Overall accuracy = $\frac{\text{\# all TP}}{\text{\# all images}}$

Overall accuracy: 24.71%



Recall

28.57%

40.00%

33.33%

18.18%

0.00%

26.32%

12.50%

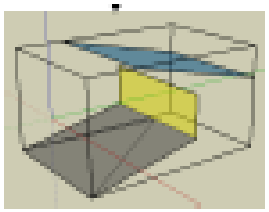
0.00%

Chance: 12.50%

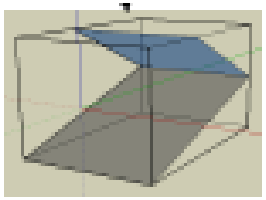
Experimental Results

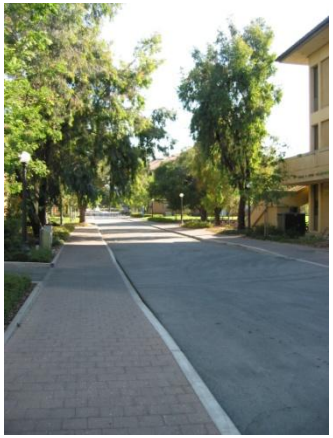
- Some misclassified examples

should be

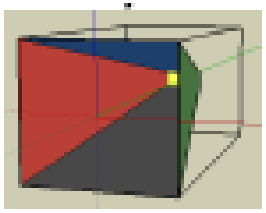


classified as

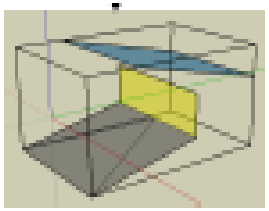




should be



classified as



Experimental Results

- What if we use multi-scale features?
- For each image, I extracted features from 6 levels of the corresponding pyramid, and performed PCA to keep the feature dimension the same as in the paper (64)
- Result: the overall accuracy increases to 31.76%

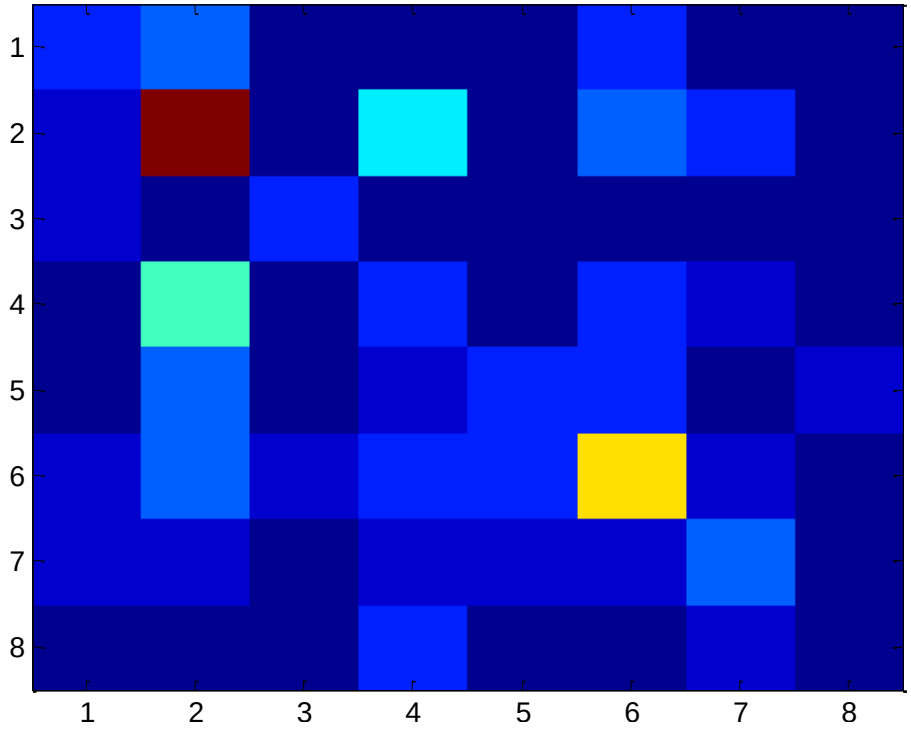
Experimental Results

- What if we use Gist features?
- The Gist features also undergo PCA to keep its dimension to be 64
- Still using multi-class SVM
- Result: the overall accuracy further increases to 40.00%

Experimental Results

- Confusion matrix

Overall accuracy: 40.00%



Recall

28.57%

56.00%

66.67%

18.18%

22.22%

47.37%

37.50%

0.00%

Chance: 12.50%

Experimental Results

- What is we use replace multi-class SVM with simple k-NN?
- The accuracy turns out to be better! (42.35%)
- Fancy stuffs are not necessarily good

Paper #3

- A. Saxena, S. H. Chung, A. Y. Ng, “Learning Depth from Single Monocular Images,” NIPS, 2005

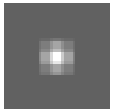
- Goal: recover *absolute depth value* for each small patch of a single monocular image

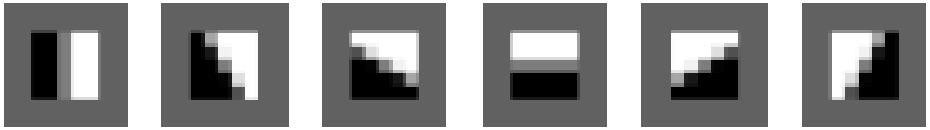
Intuition

- Texture intensity, edge directions and haze look different at different depths
  - They can be extracted as cues to infer depth
- Inferring depth from the cues of an individual patch is not reliable
  - Incorporate the cues of nearby patches
  - Use MRF to enforce constraints

Extracting Features for Absolute Depth

- For each patch
  - Apply a filter bank on the intensity channel to extract texture energy

  - Apply a low-pass filter  on the two color channels to capture haze.
  - Apply another filter bank on the intensity channel to extract edge directions



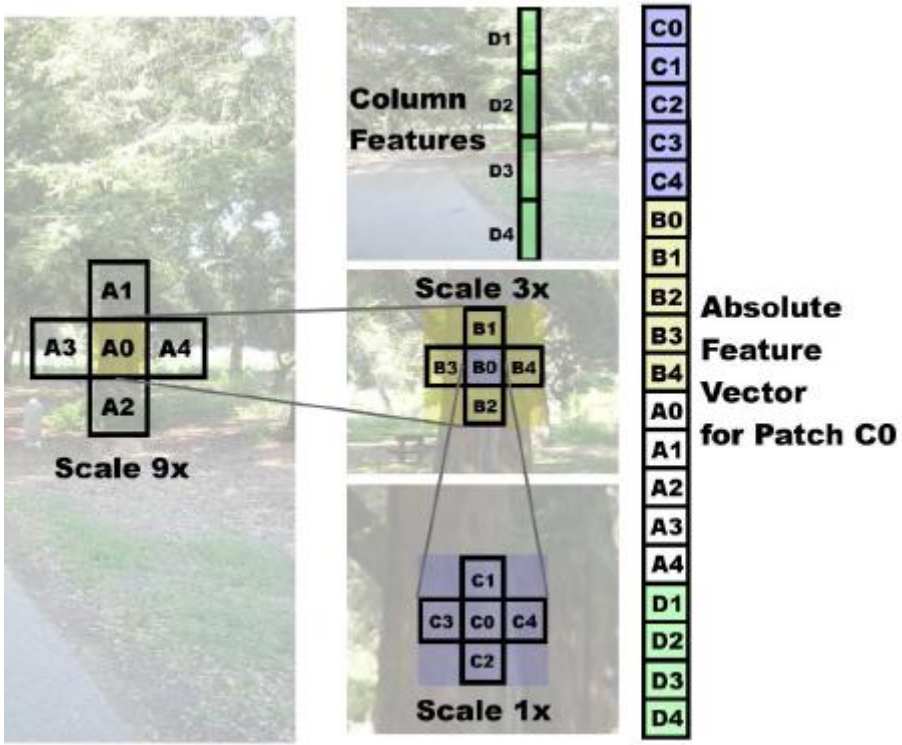
Extracting Features for Absolute Depth

- The absolute and squared filter outputs are summed over all the pixels within the patch
  - Each filter yields two values
  - 17 filters in total
  - Initial feature vector of dimension 34

$$E_i(n) = \sum_{(x,y)\in \text{patch}(i)} |I(x,y) * F_n(x,y)|^k, \quad k = \{1,2\}$$

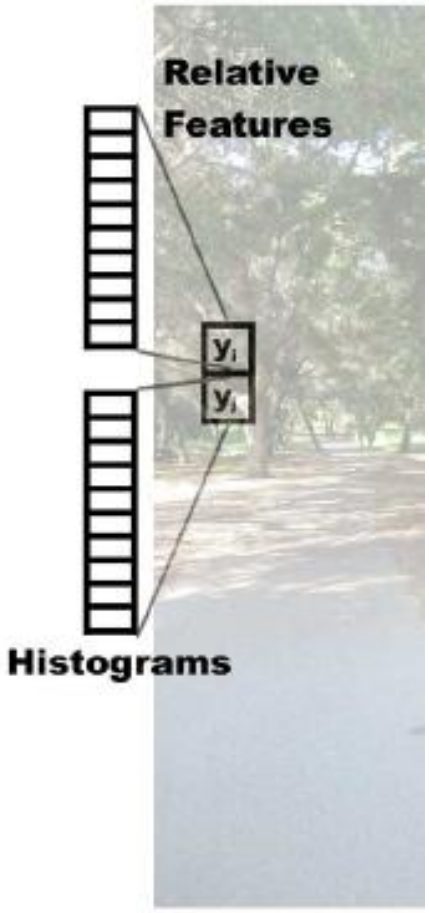
Extracting Features for Absolute Depth

- To incorporate contextual information
  - Cues from the four immediate neighboring patches are included
  - The process is repeated at three scales
  - Cues from the column the patch lies in
  - Final feature vector: 19*34=646 dimensional



Extracting Features for Relative Depth

- These features are used to estimate how different the depths of two adjacent patches can be
- Measure the difference of the statistics of the two patches
  - Uses the difference of the concatenated histograms of the 17 filter outputs
  - 10 bins for each histogram, yielding a dimension of 170



Gaussian MRF Model

- The model

$$P(d|X;\theta,\sigma) = \frac{1}{Z}\exp\left(-\sum_{i=1}^{M}\frac{(d_i(1)-x_i^T\theta_r)^2}{2\sigma_{1r}^2} - \sum_{s=1}^{3}\sum_{i=1}^{M}\sum_{j\in N_s(i)}\frac{(d_i(s)-d_j(s))^2}{2\sigma_{2rs}^2}\right)$$

  - Unary potential

$$-\sum_{i=1}^{M}\frac{(d_i(1)-x_i^T\theta_r)^2}{2\sigma_{1r}^2}$$

    - Linear predictor
    - Uncertainty reflected by σ_{1r}^2, which depends on the absolute depth feature x_i of the patch
    - Different rows have different model parameters (camera is mounted horizontally)

Gaussian MRF Model

- The model
  - Pairwise potential

$$-\sum_{s=1}^{3}\sum_{i=1}^{M}\sum_{j\in N_s(i)}\frac{(d_i(s)-d_j(s))^2}{2\sigma_{2rs}^2}$$

  - Encourage smooth depth map
  - Strength of smoothness constraint controlled by σ_{2rs}^2 , which is determined by the appearance difference of neighboring patches (captured by relative depth feature)
  - Different rows have different model parameters

Gaussian MRF Model

- The model
  - Pairwise potential

$$-\sum_{s=1}^{3}\sum_{i=1}^{M}\sum_{j\in N_s(i)}\frac{(d_i(s)-d_j(s))^2}{2\sigma_{2rs}^2}$$

  - Multi-scale model
    - At the smaller scale

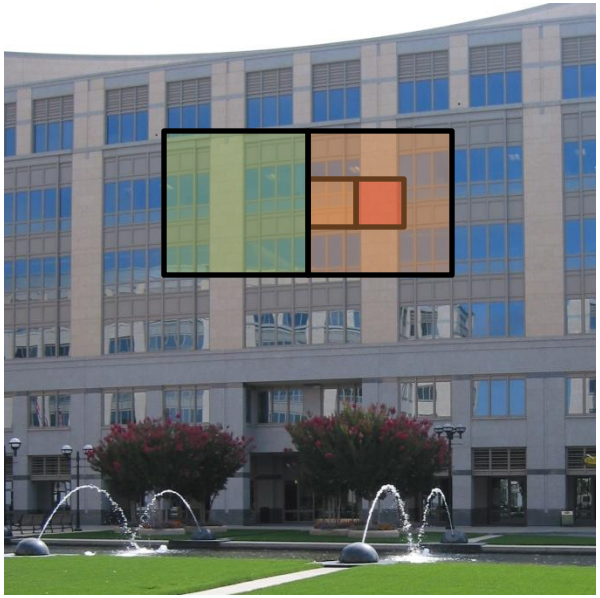
      Appearance very different

      Allows for large discontinuity in depth

    - At the larger scale

      Appearance similar

      Strong constraint on depth continuity



Gaussian MRF Model

- Learning parameters
  - Unary parameters

$$-\sum_{i=1}^{M}\frac{(d_i(1)-x_i^T\theta_r)^2}{2\sigma_{1r}^2}$$

    - θ_r : linear least square problem
    - $\sigma_{1r}^2 = v_r^T x_i$, choose v_r to fit the expected value of $(d_i(r)-\theta_r^T x_i)^2$, s.t. $v_r \geq 0$
  - Pairwise parameters

$$-\sum_{s=1}^{3}\sum_{i=1}^{M}\sum_{j\in N_s(i)}\frac{(d_i(s)-d_j(s))^2}{2\sigma_{2rs}^2}$$

    - $\sigma_{2rs}^2 = u_{rs}^T|y_{ijs}|$, choose u_{rs} to fit the expected value of $(d_i(s)-d_j(s))^2$, s.t. $u_{rs} \geq 0$

Laplacian MRF Model

- The model

$$P(d|X;\theta,\lambda) = \frac{1}{Z}\exp\left(-\sum_{i=1}^{M}\frac{|d_i(1)-x_i^T\theta_r|}{\lambda_{1r}} - \sum_{s=1}^{3}\sum_{i=1}^{M}\sum_{j\in N_s(i)}\frac{|d_i(s)-d_j(s)|}{\lambda_{2rs}}\right)$$

  - Replace square with absolute value
  - Replace variance parameters with spread parameters
- Learning parameters
  - Choose θ_r to minimize $\sum_{i=1}^{M}|d_i(1)-x_i^T\theta_r|$
  - Fit λ_{1r} to the expected value of $|d_i(1)-x_i^T\theta_r|$
  - Fit λ_{2rs} to the expected value of $|d_i(s)-d_j(s)|$

Depth Inference in Test Image

- Gaussian model
  - The log-likelihood is quadratic in d, the MAP estimate is found in closed form
- Laplacian model
  - Use linear programming to find MAP estimate

Experimental Results

- Ground truth depth maps collected using a 3D laser scanner
- 425 image+depthmap pairs
- 75% for training, 25% for testing
- Transform all the depths to a log scale before training

Experimental Results

Table 1: Effect of multiscale and column features on accuracy. The average absolute errors (RMS errors gave similar results) are on a log scale (base 10). H_1 and H_2 represent summary statistics for $k = 1, 2$. S_1, S_2 and S_3 represent the 3 scales. C represents the column features. Baseline is trained with only the bias term (no features).

FEATURE	ALL	FOREST	CAMPUS	INDOOR
BASELINE	.295	.283	.343	.228
GAUSSIAN (S_1,S_2,S_3, H_1,H_2,*no neighbors*)	.162	.159	.166	.165
GAUSSIAN (S_1, H_1,H_2)	.171	.164	.189	.173
GAUSSIAN (S_1,S_2, H_1,H_2)	.155	.151	.164	.157
GAUSSIAN (S_1, S_2,S_3, H_1,H_2)	.144	.144	.143	.144
GAUSSIAN (S_1,S_2,S_3, C, H_1)	.139	.140	.141	.122
GAUSSIAN (S_1,S_2,S_3, C, H_1,H_2)	.133	.135	.132	.124
LAPLACIAN	.132	.133	.142	.084

- Multiscale and column features help a lot
- Pairwise terms further improves performance

Experimental Results

Table 1: Effect of multiscale and column features on accuracy. The average absolute errors (RMS errors gave similar results) are on a log scale (base 10). H_1 and H_2 represent summary statistics for $k = 1, 2$. S_1, S_2 and S_3 represent the 3 scales. C represents the column features. Baseline is trained with only the bias term (no features).

FEATURE	ALL	FOREST	CAMPUS	INDOOR
BASELINE	.295	.283	.343	.228
GAUSSIAN (S_1,S_2,S_3, H_1,H_2,*no neighbors*)	.162	.159	.166	.165
GAUSSIAN (S_1, H_1,H_2)	.171	.164	.189	.173
GAUSSIAN (S_1,S_2, H_1,H_2)	.155	.151	.164	.157
GAUSSIAN (S_1, S_2,S_3, H_1,H_2)	.144	.144	.143	.144
GAUSSIAN (S_1,S_2,S_3, C, H_1)	.139	.140	.141	.122
GAUSSIAN (S_1,S_2,S_3, C, H_1,H_2)	.133	.135	.132	.124
LAPLACIAN	.132	.133	.142	.084

- Laplacian model outperforms Gaussian model
  - $(d_i - d_j)$ empirically appears Laplacian
  - Heavier tail more robust to outliers and errors
  - Laplacian tends to model sharp transitions better

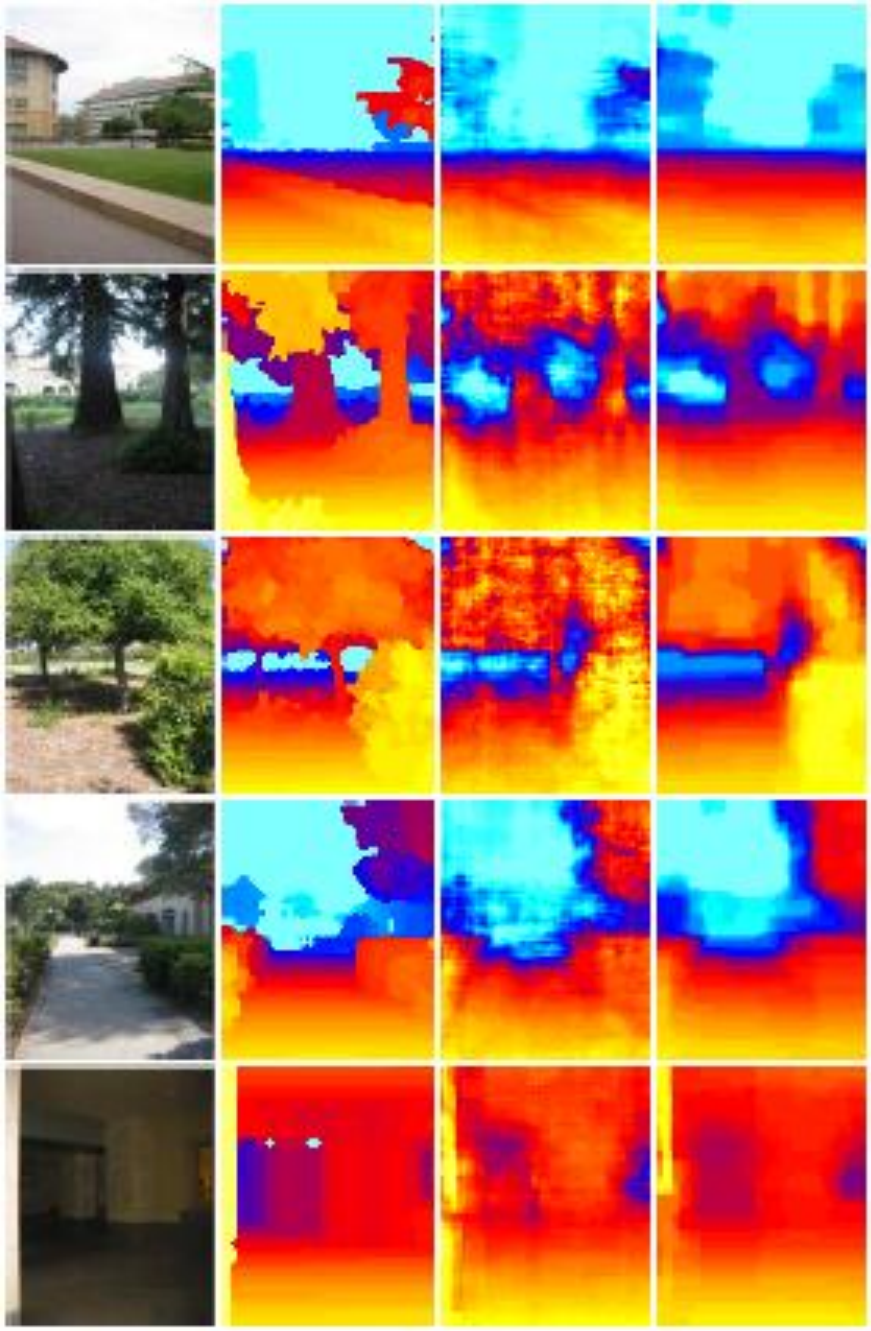


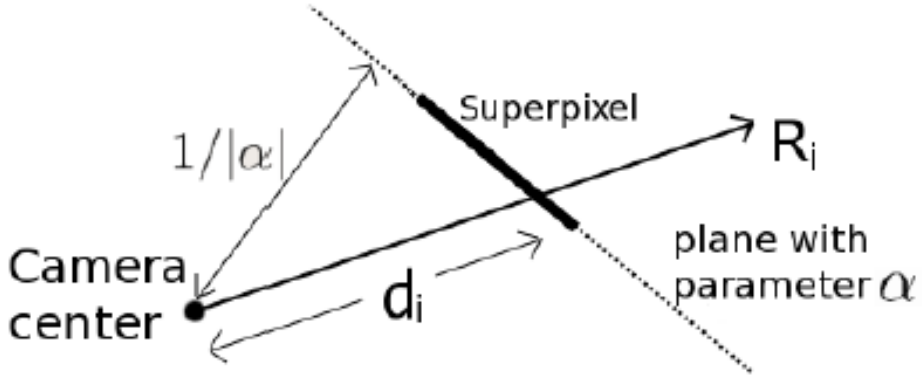
Figure 3: Results for a varied set of environments, showing original image (column 1), ground truth depthmap (column 2), predicted depthmap by Gaussian model (column 3), predicted depthmap by Laplacian model (column 4). (**Best viewed in color**)

Further Extension…

- Not only do we want to estimate the depth map, but we want to infer the orientations of surfaces as well
- Follow-up paper:
  A. Saxena, M. Sun, A. Y. Ng, "Make3D: Learning 3-D Scene Structure from a Single Still Image," PAMI, 2008

Further Extension…

- How to relate surface orientation with depth?



  - Plane equation: $\alpha^T q = 1$, where α represents both the 3D location and orientation of the surface
  - R_i : unit vector from the camera center to a point lying on the plane. This vector is known.
  - The coordinate of the point is $R_i\, d_i$
  - Therefore, $d_i = 1/R_i^T \alpha$

Further Extension…

- Modification of the model
  - Replace image patches with superpixels
  - For each superpixel, extract the features mentioned in the previous paper, plus shape and location features
  - In unary potential, relative error is used

$$\frac{\hat{d}_{i,s_i} - d_{i,s_i}}{d_{i,s_i}} = \frac{1}{d_{i,s_i}}(\hat{d}_{i,s_i}) - 1 = R_{i,s_i}^T \alpha_i (x_{i,s_i}^T \theta_r) - 1$$

$$- \sum_i \sum_{s_i=1}^{S_i} \nu_{i,s_i} \left| R_{i,s_i}^T \alpha_i (x_{i,s_i}^T \theta_r) - 1 \right|$$

  - ν_{i,s_i} is the confidence of the linear predictor, which is learned as another linear function of the superpixel features

Further Extension…

- Modification of the model
  - In pairwise potential, a pair of surfaces is penalized for deviating from
    - Being connected $-y_{ij}|(R_{i,s_i}^T\alpha_i - R_{j,s_j}^T\alpha_j)\hat{d}|$
    - Being co-planar $-y_{ij}|(R_{j,s_j''}^T\alpha_i - R_{j,s_j''}^T\alpha_j)\hat{d}_{s_j''}|$
    - Being co-linear $-y_{ij}|(R_{j,s_j}^T\alpha_i - R_{j,s_j}^T\alpha_j)\hat{d}|$
  - Strength of penalty is different under different conditions
    - When there is evidence that two superpixels are separated by an occlusion boundary, the first two penalties are alleviated accordingly
    - When there is evidence that two superpixels lie on a common straight long line, the third penalty is imposed accordingly

Further Extension…

- Quantitative comparison

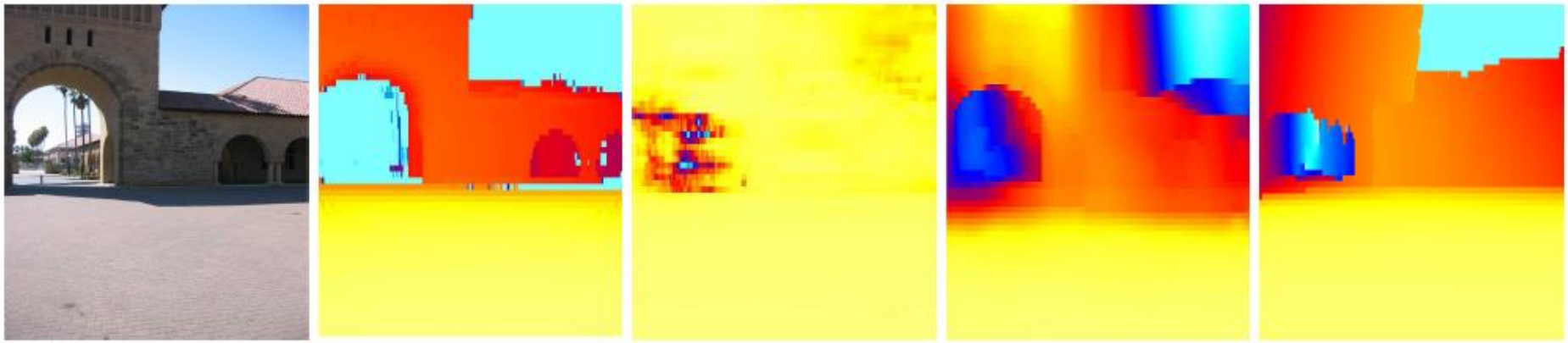
RESULTS: QUANTITATIVE COMPARISON OF VARIOUS METHODS.

METHOD	CORRECT (%)	% PLANES CORRECT	$\log_{10}$	REL
SCN	NA	NA	0.198	0.530
HEH	33.1%	50.3%	0.320	1.423
BASELINE-1	0%	NA	0.300	0.698
NO PRIORS	0%	NA	0.170	0.447
POINT-WISE MRF	23%	NA	**0.149**	0.458
BASELINE-2	0%	0%	0.334	0.516
NO PRIORS	0%	0%	0.205	0.392
CO-PLANAR	45.7%	57.1%	0.191	0.373
PP-MRF	**64.9%**	**71.2%**	0.187	**0.370**

  - Spatial support is important
  - Geometric constraints help, especially qualitatively

Further Extension…

- Qualitative comparison



(a) (b) (c) (d) (e)

Fig. 11. (a) Original Image, (b) Ground truth depthmap, (c) Depth from image features only, (d) Point-wise MRF, (e) Plane parameter MRF. (*Best viewed in color.*)

Further Extension…

- More impressive results

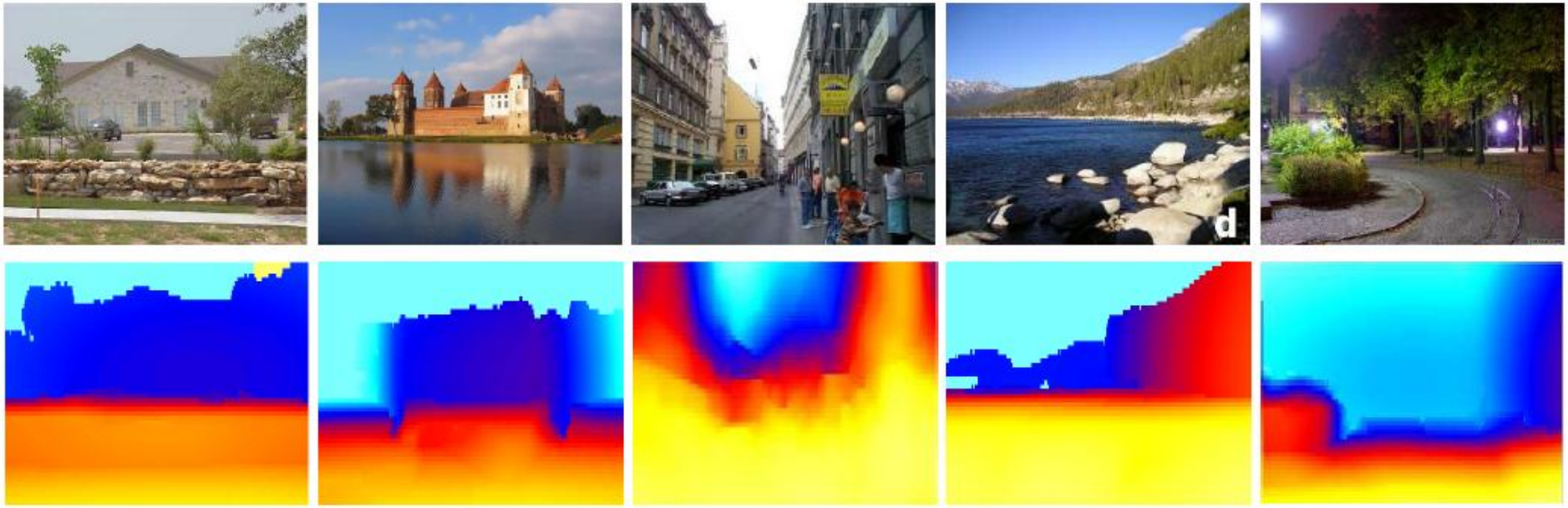


Fig. 13. Typical results from our algorithm. (Top row) Original images, (Bottom row) depthmaps (shown in log scale, yellow is closest, followed by red and then blue) generated from the images using our plane parameter MRF. (*Best viewed in color.*)

Thank you!