SCENE AND CONTEXT

SCENE UNDERSTANDING

What is goal of scene understanding:

- Build machine that can see like humans to automatically interpret the content of the images.

Comparing with traditional vision problem:

- Study on larger scale
- Human vision related tasks

LARGER SCALE

More image information.
Context information.

focal length = 35 mm

HUMAN VISION RELATED TASK

More similar as the way that human understand the image
Infer more useful information from image






HOW DO HUMAN LEARN?

Bayesian Rules:

$$P(A \mid B) = P(B \mid A) \cdot P(A) / P(B)$$

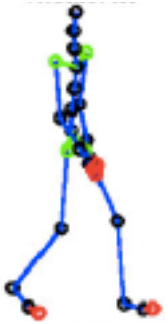
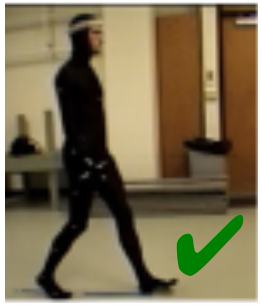
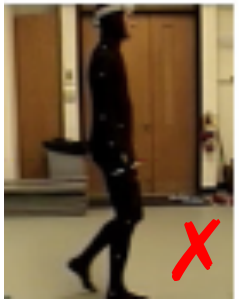
In practice: Infer abstract knowledge based on observation

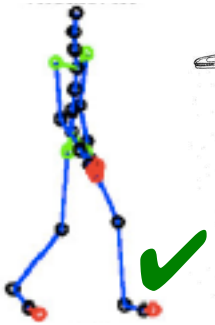
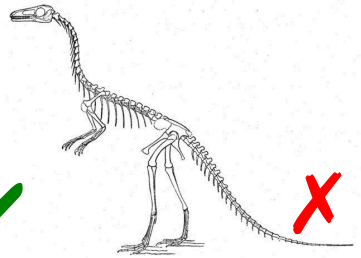
$$P(W \mid I) = P(I \mid W) \cdot P(W) / P(I)$$
$$\propto P(I \mid W) \cdot P(W)$$

Posterior probability

Likelihood: The probability of getting I given model W

Prior: The probability of W w/o seeing any observation

HOW DO HUMAN LEARN?

To teach human baby what is “horse”: show 3 pictures and let them learn by themselves.

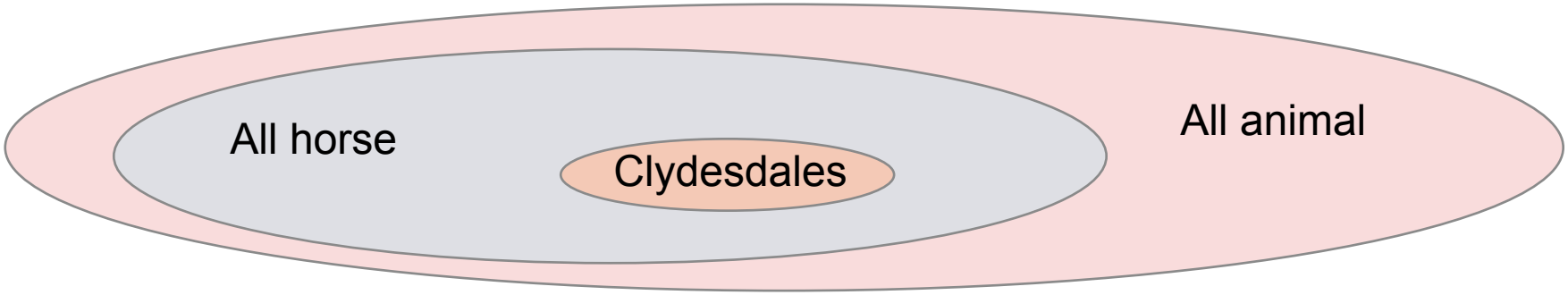
They can be very successful to learn the correct concept.

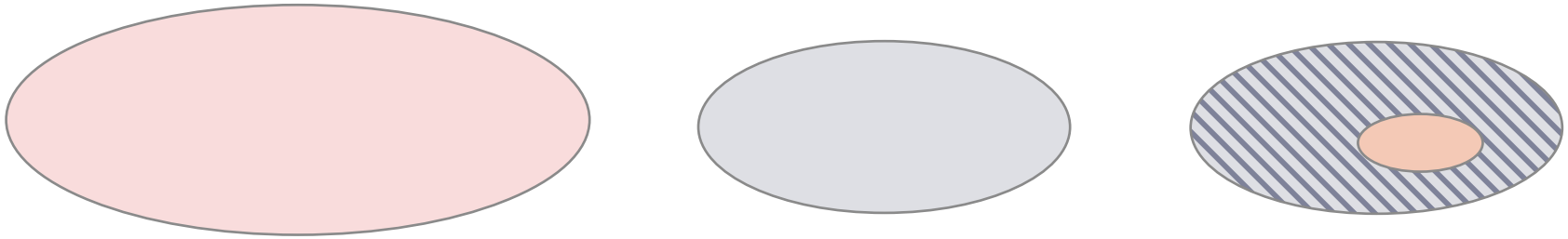
But all the following concepts can explain the images:

- “horse” = all horse
- “horse” = all horse but not Clydesdales
- “horse” = all animal

$I =$

“horse”





W_1 W_2 W_3

Likelihood: What is the probability of sampling these 3 horse images given the concept?

$$P(I \mid W_3) \approx P(I \mid W_2) > P(I \mid W_1)$$

Prior: How likely would the concept be the referent of a common word?

$$P(W_1) \approx P(W_2) > P(W_3)$$

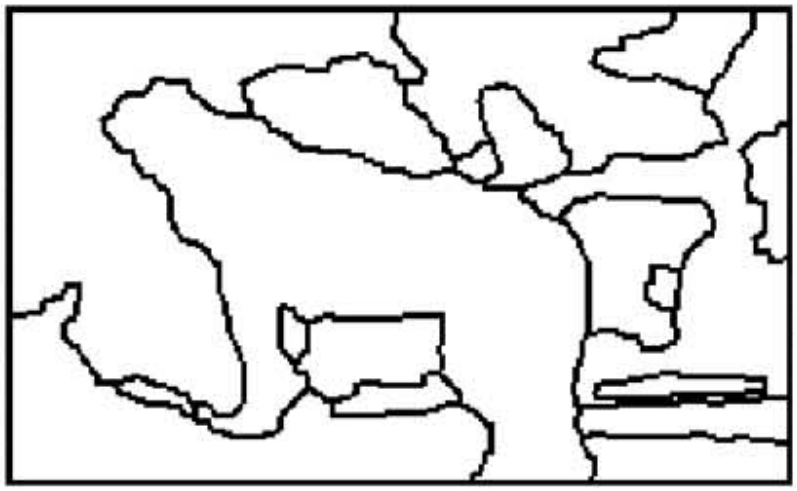
$$W_2 = \underset{W}{\operatorname{argmax}}(P(I \mid W) \cdot P(W))$$

TASK 1: IMAGE SEGMENTATION

I: **Gray scaled image**

***W*: A pixel-wise label map (segmentation)**

$$W_{best} = \arg\max_{W}(P(W \mid I)) = \operatorname{argmax}_{W}(P(I \mid W) \cdot P(W))$$



How to formulate the segmentation problem?

Warning: A huge wave of equations is approaching!

FORMULATION #1

Image Lattice: $\Lambda = \{(i,j) : 1 \leq i \leq L, 1 \leq j \leq H\}$

Image: $\mathbf{I}_\Lambda$

For any point $v \in \Lambda$ **either** $\mathbf{I}_v \in \{0, \ldots, G\}$

Lattice partition into K disjoint regions:

$$\Lambda = \cup_{i=1}^{K} R_i, \quad R_i \cap R_j = \emptyset, \quad \forall i \neq j$$

Region is discrete label map: $R \subset \Lambda$

Region Boundary is Continuous: $\Gamma_i = \partial R_i$

FORMULATION #2

Each Image Region $\mathbf{I}_R$ is a realization from a probabilistic model $p(\mathbf{I}_R; \Theta)$

Θ are parameters of model indexed by ℓ

A segmentation is denoted by a vector of hidden variables W; K is number of regions

$$W = (K, \{(R_i, \ell_i, \Theta_i); \quad i = 1, 2, \ldots, K\})$$

Bayesian Framework:

$$W \sim p(W|\mathbf{I}) \propto p(\mathbf{I}|W)p(W), \quad W \in \Omega.$$

Space of all segmentations

Posterior

Likelihood

Prior

$$W \sim p(W|\mathbf{I}) \propto p(\mathbf{I}|W)p(W), \quad W \in \Omega.$$

PRIOR OVER SEGMENTATIONS

$$p(W) \propto p(K) \prod_{i=1}^{K} p(R_i)p(\ell_i)p(\Theta_i|\ell_i)$$

$p(K) \propto e^{-\lambda_0 K}$ Want less regions

$p(\Gamma) \propto e^{-\mu \oint_\Gamma ds}$ Want round-ish regions

# of model params

$p(\ell)$ ~ uniform

$p(\Theta|\ell) \propto e^{-\nu|\Theta|}$ Want less complex models

$p(A) \propto e^{-\gamma A^c}$ $A = |R|$ Want small regions

$$p(W) \propto \exp\left\{-\lambda_0 K - \sum_{i=1}^{K}\left[\mu \oint_{\partial R_i} ds + \gamma|R_i|^c + \nu|\Theta_i|\right]\right\}$$

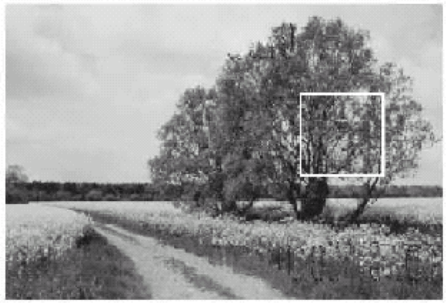
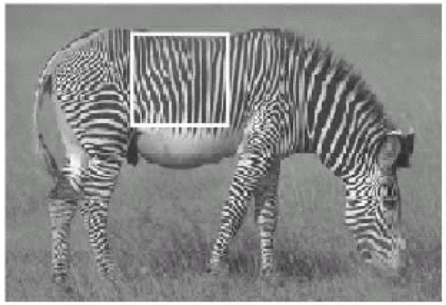
LIKELIHOOD FOR IMAGES

Visual Patterns are independent stochastic processes

$$p(\mathbf{I}|W) = \prod_{i=1}^{K} p(\mathbf{I}_{R_i}; \Theta_i, \ell_i)$$

- ℓ **is model-type index** $\ell \in \{g_1, g_2, g_3, g_4\}$
- Θ_i **is model parameter vector**
- $\mathbf{I}_{R_i}$ **is image appearance in i-th region**

FOUR GRAY-LEVEL MODELS

Uniform	Clutter	Texture	Shading
			
Gaussian	Intensity Histogram	Gabor Response Histogram	B-Spline

Given parameter Θ and region image ***I***,
we can measure $P(I_R \mid \Theta_i, l_i)$ by defined distance.

WHAT DID WE JUST DO?

Def. of Segmentation:

$$W = (K, \{(R_i, \ell_i, \Theta_i); \quad i = 1, 2, \ldots, K\})$$

Score (probability) of Segmentation:

$$W \sim p(W|\mathbf{I}) \propto p(\mathbf{I}|W)p(W), \quad W \in \Omega.$$

Likelihood of Image = product of region likelihoods

$$p(\mathbf{I}|W) = \prod_{i=1}^{K} p(\mathbf{I}_{R_i}; \Theta_i, \ell_i)$$

Regions defined by k-partition:

$$\Lambda = \cup_{i=1}^{K} R_i, \quad R_i \cap R_j = \emptyset, \quad \forall i \neq j$$

WHAT DO WE DO WITH SCORES?

Given the image ***I***

Search

for optimal ***W***

SEARCH THROUGH WHAT? ANATOMY OF SOLUTION SPACE

Space of all k-partitions

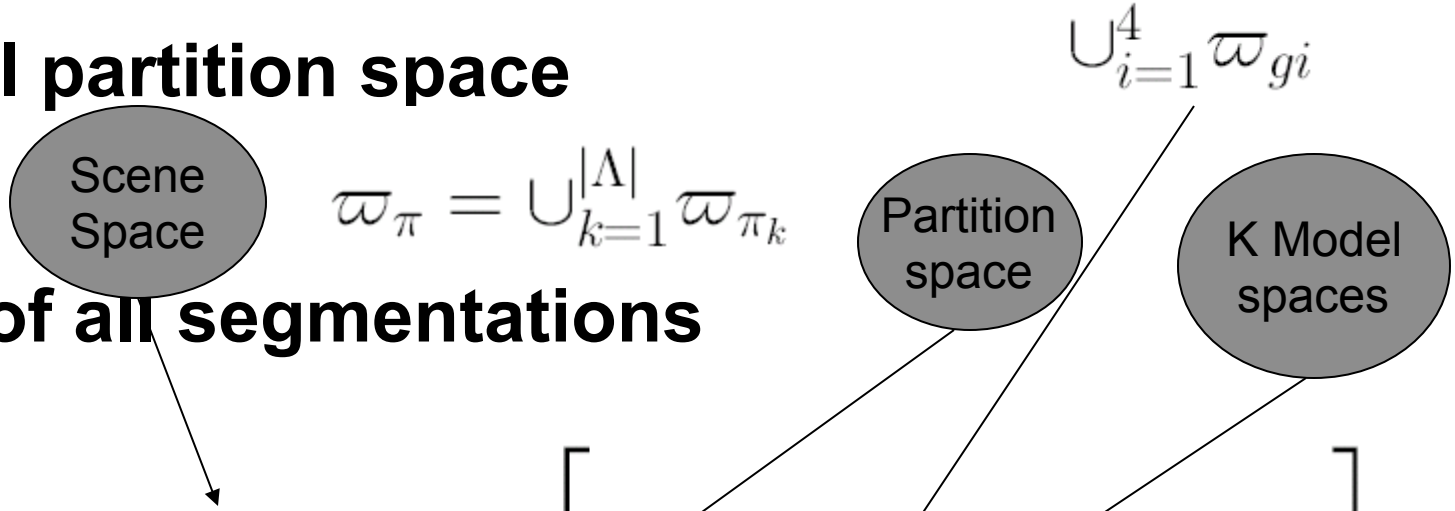
$$\pi_k = (R_1, R_2, \ldots, R_k), \quad \cup_{i=1}^{k} R_i = \Lambda, \quad R_i \cap R_j = \emptyset, \ \forall i \neq j$$

General partition space

$$\varpi_\pi = \cup_{k=1}^{|\Lambda|} \varpi_{\pi_k}$$

Space of all segmentations

$$\Omega = \cup_{k=1}^{|\Lambda|} \Omega_k = \cup_{k=1}^{|\Lambda|} \left[\varpi_{\pi_k} \times \underbrace{\varpi_\Theta \times \cdots \times \varpi_\Theta}_{k} \right]$$



SEARCHING THROUGH SEGMENTATIONS

Exhaustive Enumeration of all segmentations

Takes too long!

Greedy Search (Gradient Ascent)

Local minima!

MCMC based exploration

Described later!

WHY MCMC

What is it: Markov chain Monte Carlo

-A clever way of searching through a high-dimensional space
-A general purpose technique of generating samples from a probability

$$W \sim p(W|\mathbf{I}) \propto p(\mathbf{I}|W)p(W), \quad W \in \Omega.$$

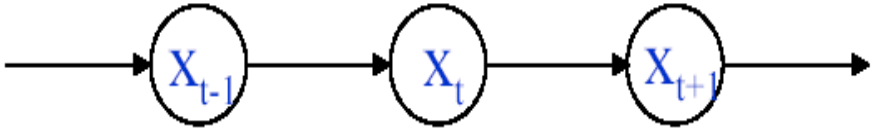
What does it do?

-Iteratively searches through space of all segmentations by constructing a Markov Chain which converges to stationary distribution

What is Markov Chain?

A **Markov chain** is a mathematical model for stochastic systems whose states, discrete or continuous, are governed by a transition probability. The current state in a Markov chain only depends on the most recent previous states, e.g. for a 1st order Markov chain.

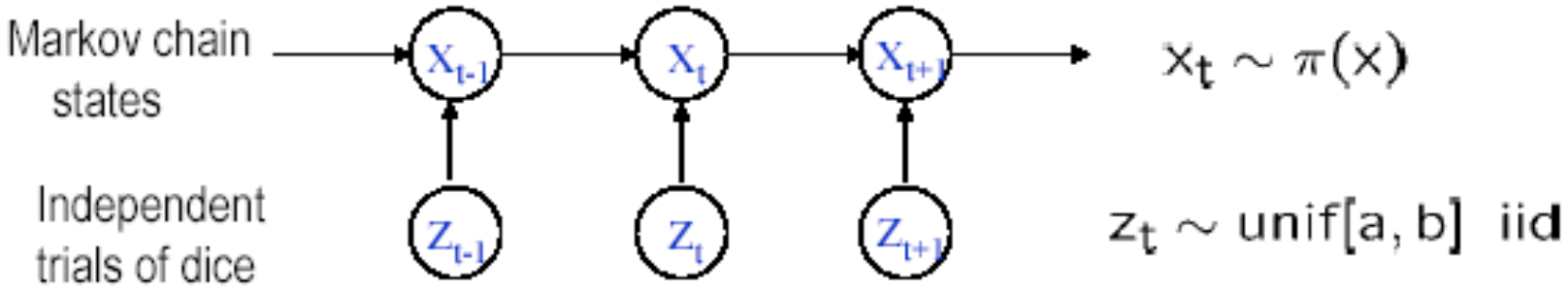
$$x_t|x_{t-1},\ldots,x_0 \sim P(x_t|x_{t-1},\ldots,x_0) = P(x_t|x_{t-1})$$



The **Markovian property** means "locality" in space or time, such as Markov random fields and Markov chain. Indeed, a discrete time Markov chain can be viewed as a special case of the Markov random fields (causal and 1-dimensional).

What is Markov Chain Monte Carlo ?

MCMC is a **general purpose technique** for generating **fair samples** from a probability in high-dimensional space, using random numbers (dice) drawn from uniform probability in certain range. A Markov chain is designed to have $\pi(x)$ being its **stationary (or invariant) probability.**



This is a non-trivial task when $\pi(x)$ is very complicated in very high dimensional spaces !

DESIGNING MARKOV CHAINS

Three Markov Chain requirements

Ergodic: from an initial segmentation W0, any other state W can be visited in finite time (no greedy algorithms); ensured by jump-diffusion dynamics

Aperiodic: ensured by random dynamics

Detailed Balance: every move is reversible

$$P(x)P(x \to x') = P(x')P(x' \to x)$$

5 DYNAMICS

1.) Boundary Diffusion $\frac{d\Gamma_{ij}(s)}{dt}$

2.) Model Adaptation $\frac{d\Theta_i}{dt}$

3.) Split Region

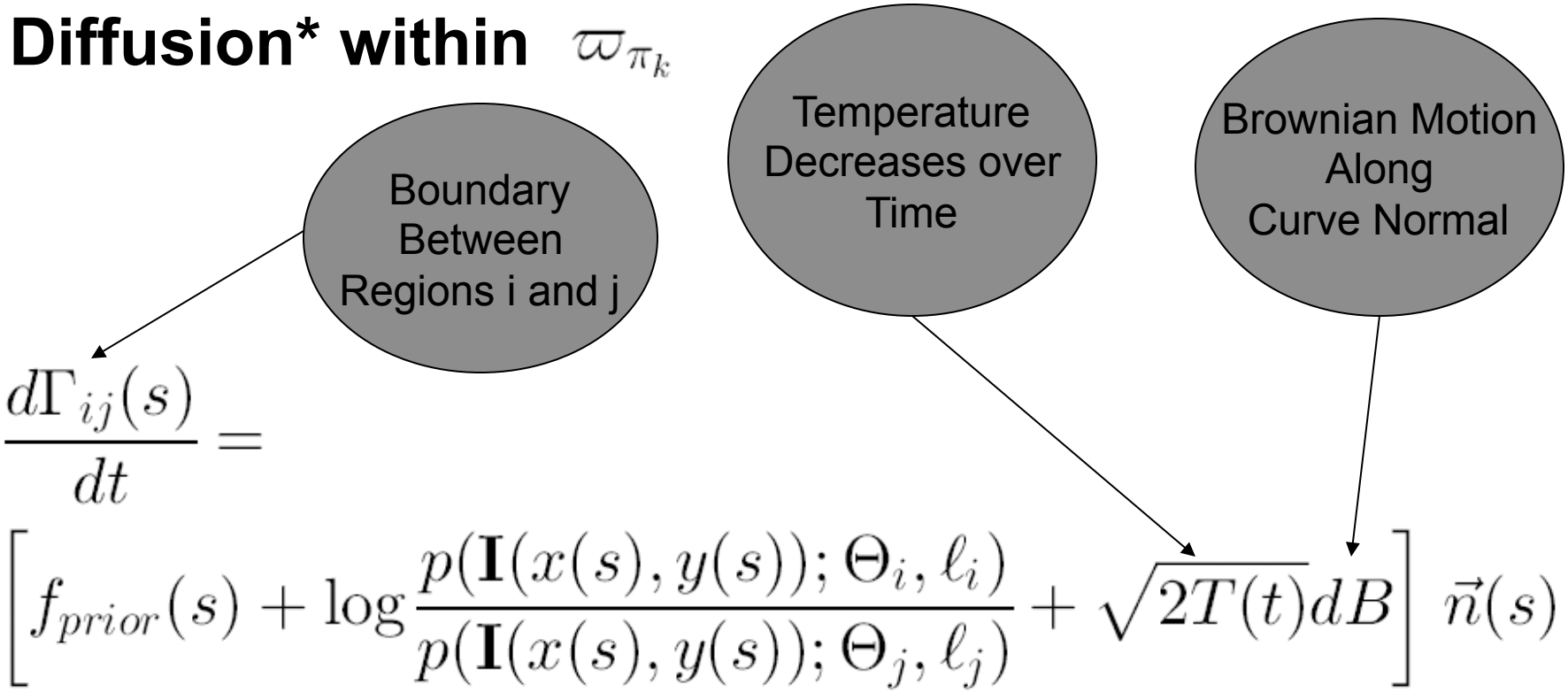
4.) Merge Region

5.) Switch Region Model

At each iteration, we choose a dynamic with probability q(1),q(2),q(3),q(4),q(5)

DYNAMICS 1: BOUNDARY DIFFUSION

Diffusion* within ϖ_{π_k}



$$\frac{d\Gamma_{ij}(s)}{dt} = \left[f_{prior}(s) + \log \frac{p(\mathbf{I}(x(s), y(s)); \Theta_i, \ell_i)}{p(\mathbf{I}(x(s), y(s)); \Theta_j, \ell_j)} + \sqrt{2T(t)} dB \right] \vec{n}(s)$$

*Movement within partition space

DYNAMICS 2: MODEL ADAPTATION

Fit the parameters* of a region by steepest ascent

$$\frac{d\Theta_i}{dt} = \frac{\partial \log p(\mathbf{I}_{R_i}; \Theta_i, \ell_i)}{\partial \Theta_i}.$$

*Movement within cue space

DYNAMICS 3-4: SPLIT AND MERGE

Split one region into two

Remaining Variables Are unchanged

$$W = (K,\ (R_k, \ell_k, \Theta_k),\ W_-) \longleftrightarrow$$
$$(K+1,\ (R_i, \ell_i, \Theta_i), (R_j, \ell_j, \Theta_j),\ W_-) = W'$$

Conditional Probability of how likely chain proposes to move to W' from W

Data-Driven Speedup

$$G(W \to dW') = q(3)q(R_k)q(\Gamma_{ij}|R_k)q(\ell_i)$$
$$q(\Theta_i|R_i,\ell_i)q(\ell_j)q(\Theta_j|R_j,\ell_j)dW'.$$

Probability of acceptance

$$\alpha(W \to dW') = \min\left(1, \frac{G(W' \to dW)p(W'|\mathbf{I})dW'}{G(W \to dW')p(W|\mathbf{I})dW}\right)$$

Standard Metropolis-Hastings Algorithm

DYNAMICS 3-4: SPLIT AND MERGE

Merge two Regions

Remaining Variables Are unchanged

$$W = (K,\ (R_k, \ell_k, \Theta_k),\ W_-) \longleftrightarrow$$

$$(K+1,\ (R_i, \ell_i, \Theta_i), (R_j, \ell_j, \Theta_j),\ W_-) = W'$$

Probability of Proposed Merge

Data-Driven Speedup

$$G(W' \to dW) = q(4)q(R_i, R_j)q(\ell_k)q(\Theta_k | R_k, \ell_k)dW$$

DYNAMICS 5: MODEL SWITCHING

Change models

$$W = (\ell_i, \Theta_i,\ W_{-}) \longleftrightarrow (\ell'_i, \Theta'_i,\ W_{-}) = W'$$

Proposal Probabilities

Data-Driven Speedup

$$G(W \to dW') = q(5)q(R_i)q(\ell'_i)q(\Theta'_i|R_i, \ell'_i)dW',$$
$$G(W' \to dW) = q(5)q(R_i)q(\ell_i)q(\Theta_i|R_i, \ell_i)dW.$$

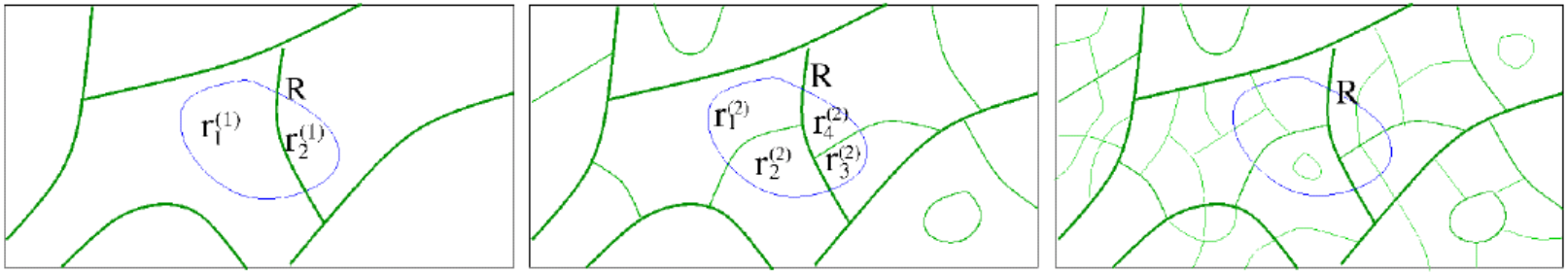
MOTIVATION OF DD

Region Splitting: How to decide where to split a region?

Model Switching: Once we switch to a new model or generate a new region, what parameters do we jump to?

K-PARTITION PARTICLES

Edge detection gives us a good idea of where we expect a boundary to be located



scale 1 scale 2 scale 3

A candidate region R_k is superimposed on the partition maps at three scales for computing a candidate boundary Γ_{ij} for the pending split.

$q(\Gamma|R)$

$$q^{(s)}(\pi_k) = \frac{1}{|\Pi_k^{(s)}|} \sum_{j=1}^{|\Pi_k^{(s)}|} G(\pi_k - \pi_{k,j}^{(s)}), \quad \text{for } s = 1, 2, 3. \ \forall k.$$

Parzen window

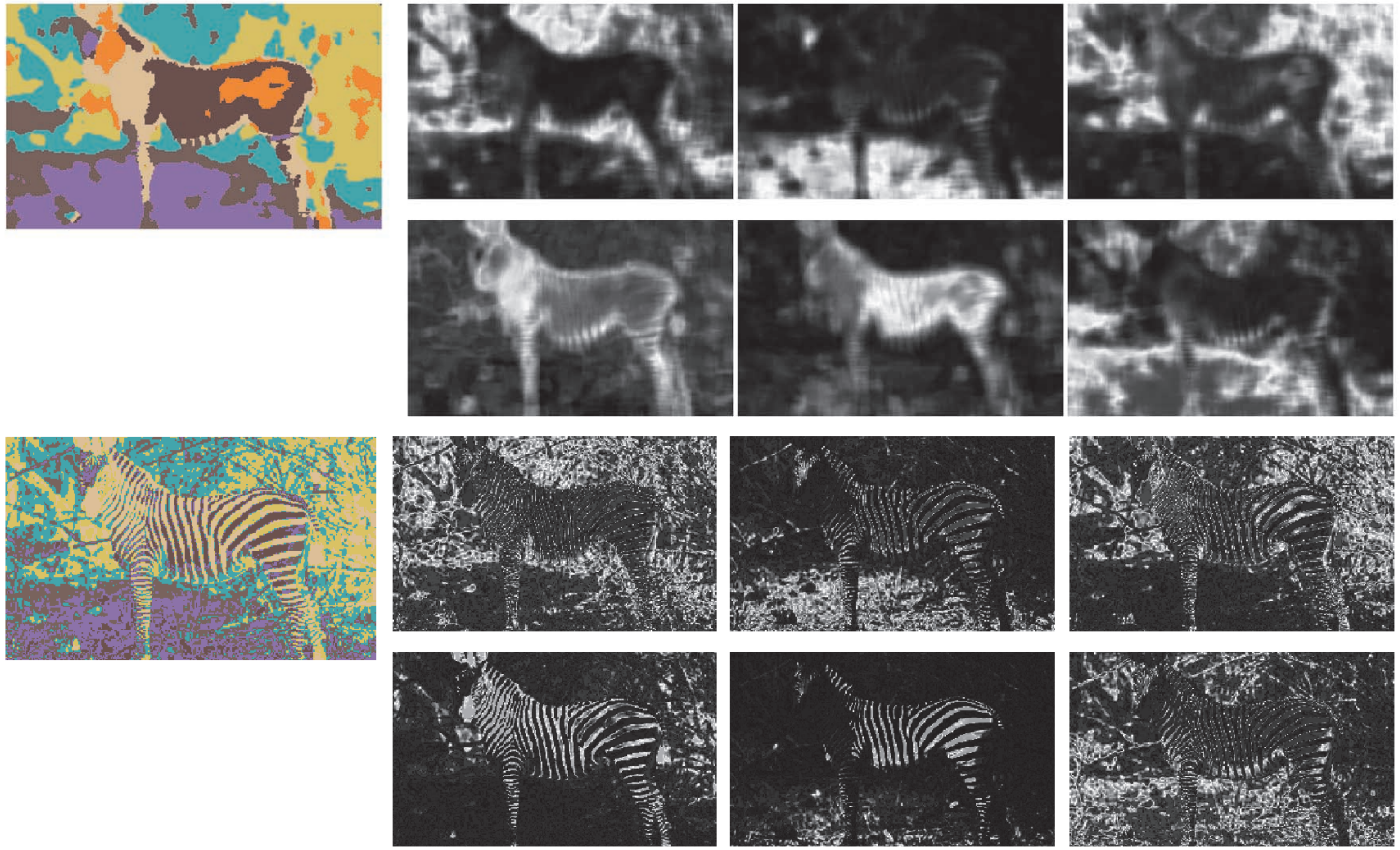
All k-partition in s scale

$$q(\pi_k) = \sum_s q(s) q^{(s)}(\pi_k), \quad \forall k.$$

Prob. of choosing scale s

PRECOMPUTE MODEL PARTICLES

Get clusters of each model, and softly assign pixel to clusters with saliency:



$S_{i,v}^{l=2}$

$S_{i,v}^{l=1}$

PRECOMPUTE MODEL PARTICLES

For a region R, ℓ,

$$p_i = \frac{1}{|R|} \sum_{v \in R} S_{i,v}^{\ell}, \quad i = 1, 2, \ldots, m, \quad \forall \ell.$$

Sampling in practice:

$$\Theta \sim q(\Theta | R, \ell) = \sum_{i=1}^{m} p_i G(\Theta - \Theta_i^{\ell}).$$

1. Choose a model according to p
2. Add some small disturbance

DATA DRIVEN METHODS

Focus on boundaries and model parameters derived from data: compute these before MCMC starts

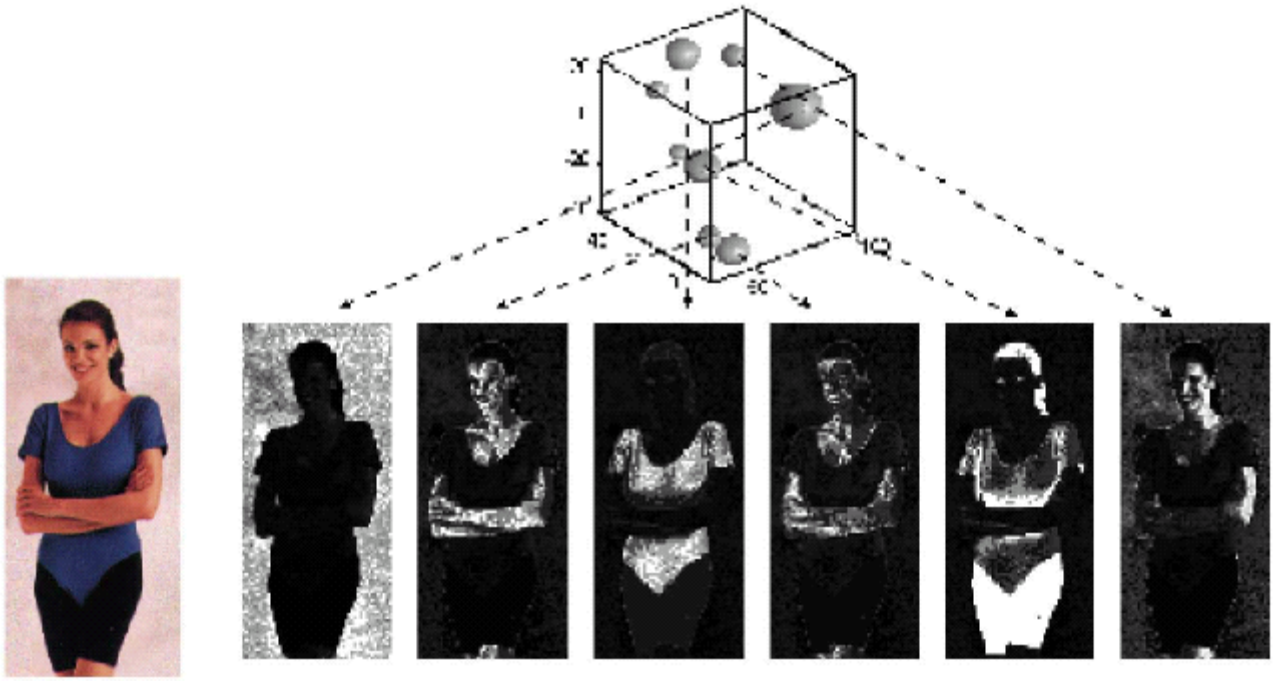
Cue Particles: Clustering in Model Space

K-partition Particles: Edge Detection

Particles Encode Probabilities Parzen Window Style

CUE PARTICLES IN ACTION

Clustering in Color Space



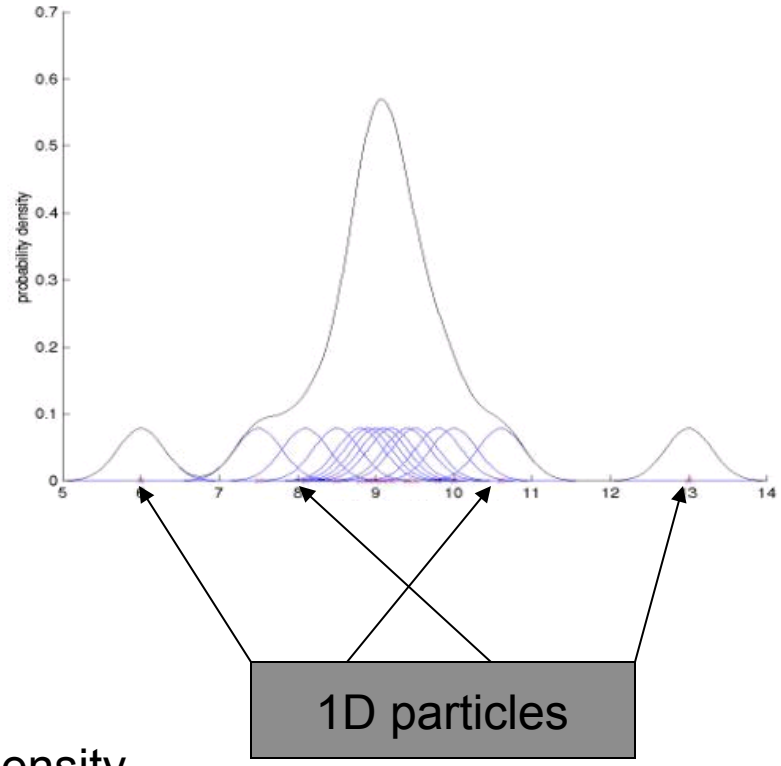
Input **I**

Color clusters and their saliency maps $S_i^{c_1}, i = 1, ..., 6$

PARTICLES OR PARZEN WINDOW* LOCATIONS?

What is this particle business about?

A particle is just the position of a parzen-window which is used for density estimation



*Parzen Windowing also known as:
Kernel Density Estimation, Non-parametric density estimation

ARE YOU AWAKE: WHAT DID WE JUST DO?

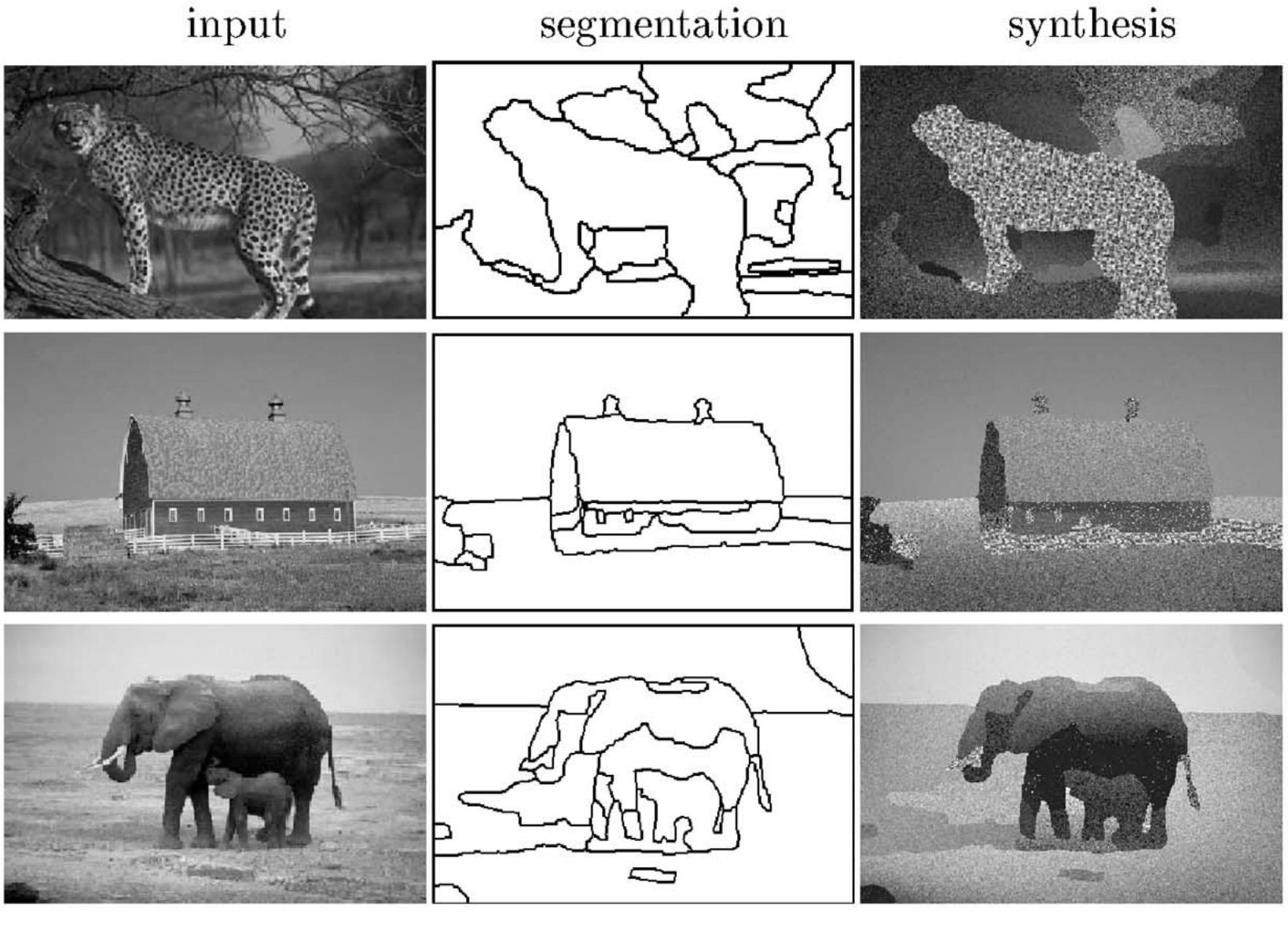
Define scores (Probability of Segmentation)
→ Search

5 MCMC dynamics to sample for solution

→ solution space is so huge

Data-Driven Speedup (key to making MCMC work in finite time)

RESULT OF DDMCMC

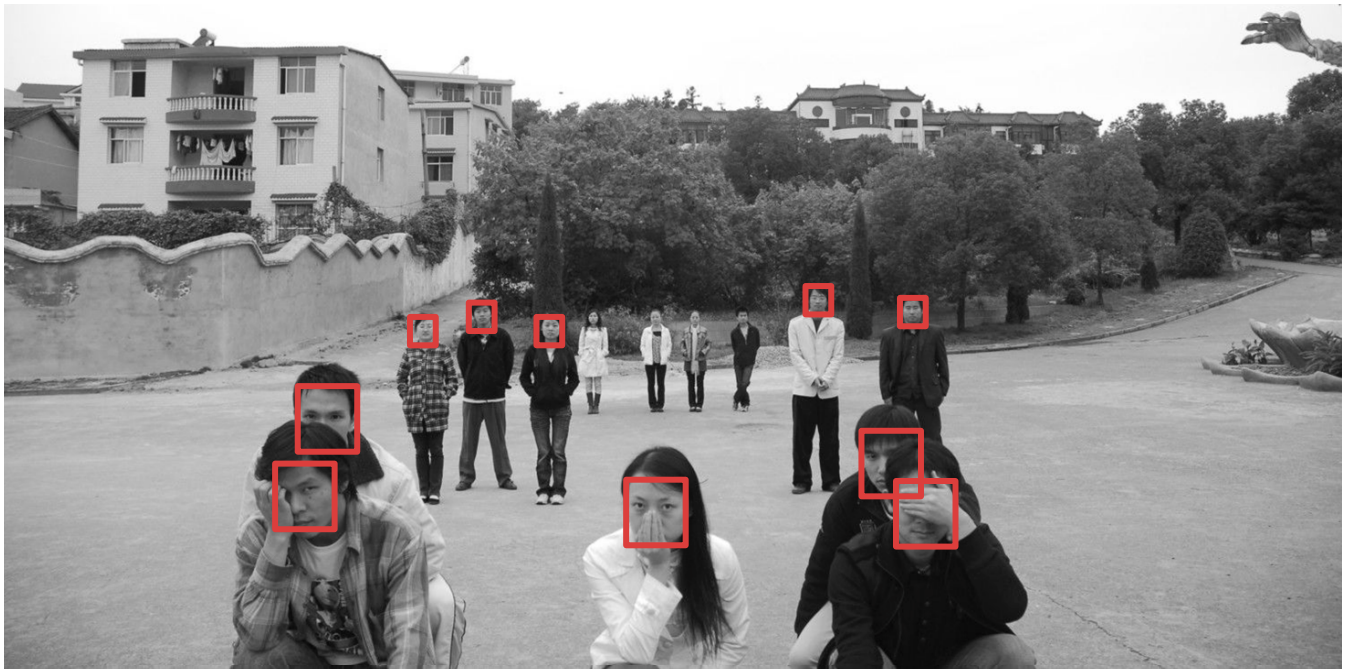


TASK 2: OBJECT (FACE) DETECTION

I: **Grayscale image**

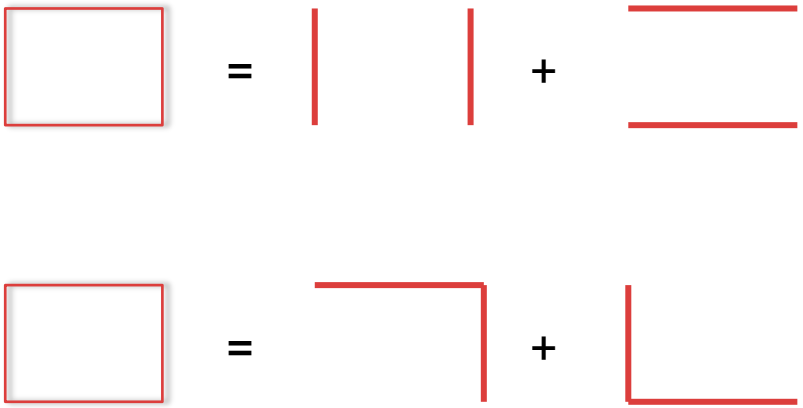
***W*: The objects in the image**

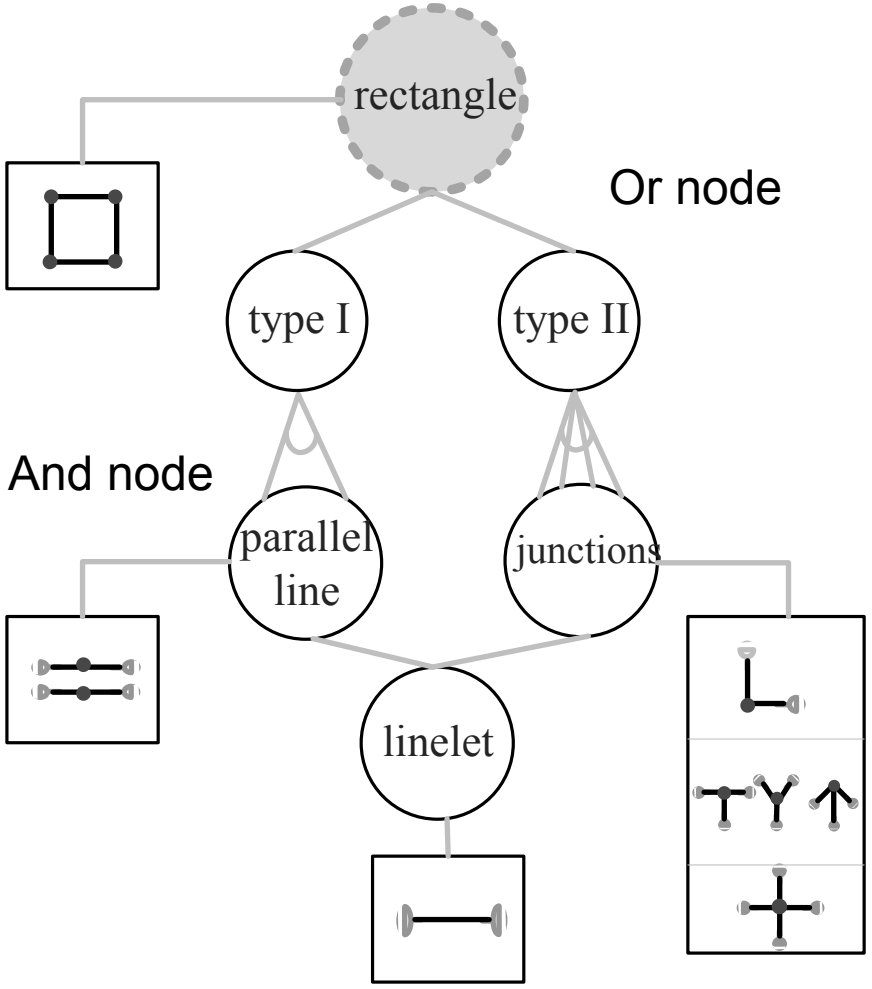
$$W_{best} = \arg\max_{W}(P(W \mid I)) = \operatorname{argmax}_{W}(P(I \mid W) \cdot P(W))$$



HOW TO MEASURE PRIOR: *P(W)*

Example: rectangle detection



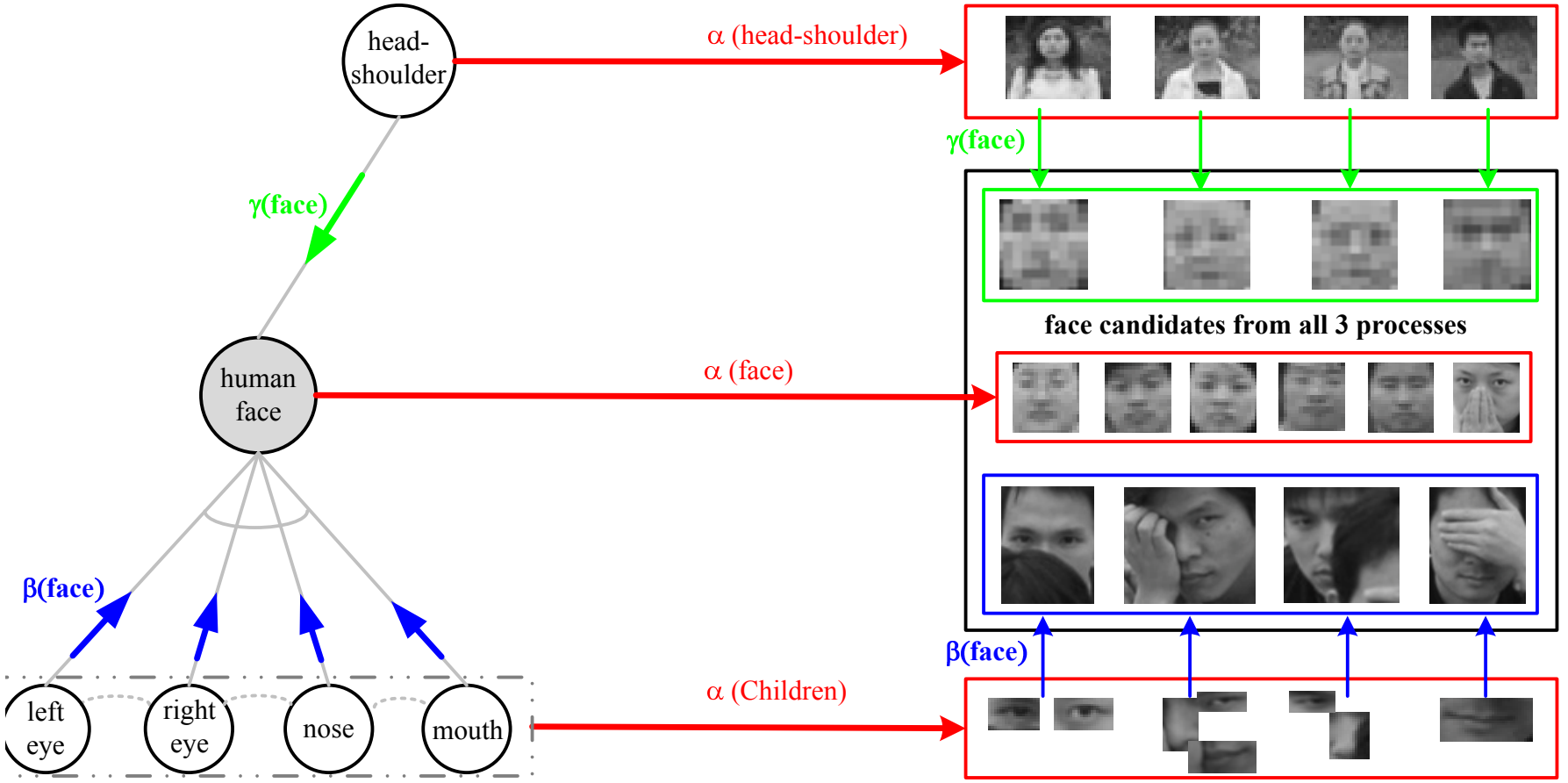


HOW TO MEASURE PRIOR: *P(W)*

Example: face detection

AoG representation of human face

Integrating α, β, γ inference processes in face detection



AND OR GRAPH PRIOR

Define the And/Or graph (AOG)

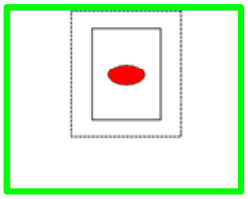
Each object is represented by a parse graph, which is an instance of the AOG by selecting parameter for each node.

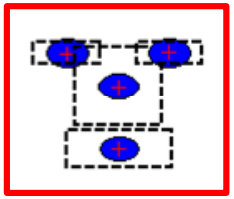
$$\mathcal{E}(pg) = -\sum_{<O,A>\in E^{pg}_{or}} \log p(A|O) \qquad \text{Or node}$$

$$-\sum_{<P,A>\in E^{pg}_{dec}} \log p(X(A)|X(P)) \qquad \text{And node: top down}$$

$$-\sum_{<C_i,C_j>\in E^{pg}_{rel}} \log p(X(C_i), X(C_j)) \qquad \text{And node: bottom up}$$

All these probability distribution can be learned from training data





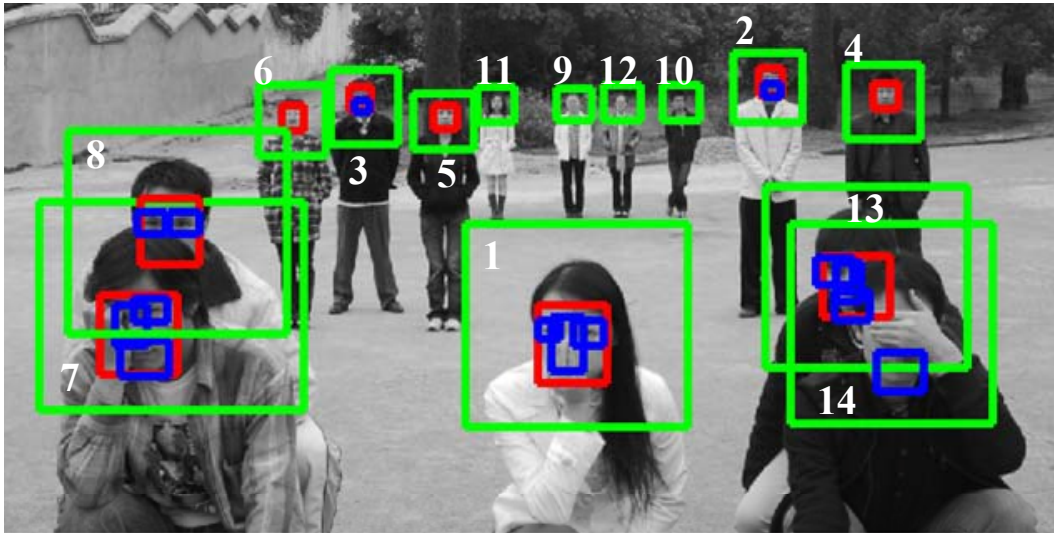
AND OR GRAPH LIKELIHOOD

$$\log(P(I \mid W))$$

The scores of part detector, trained separately.

Inference:

1. Find activation of each part
2. Convert head shoulder detection to face detection, merge if possible
3. Convert eye/nose/mouth detection, merge if possible
4. Select the top ones



WAKE UP

What have we learned up to now:

Bayesian framework: $P(W \mid I) \propto P(I \mid W) \cdot P(W)$

A framework inferring model base on observation

1. **likelihood: observation related**
2. **prior: prior knowledge about the problem**

AOG: a graph model of prior knowledge

Inference:

1. **If space of W is large: sampling or greedy search**
2. **or: sacrifice model and get global optima**

CONTEXT

AOG models relation between parts with graph

Problem： hand-coded context

1. miss the useful context in prior knowledge

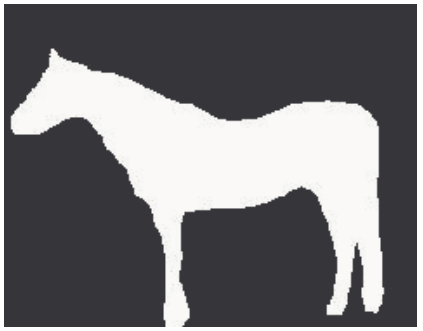
2. add useless context and hard to calibrate

More flexible way of handling context information:

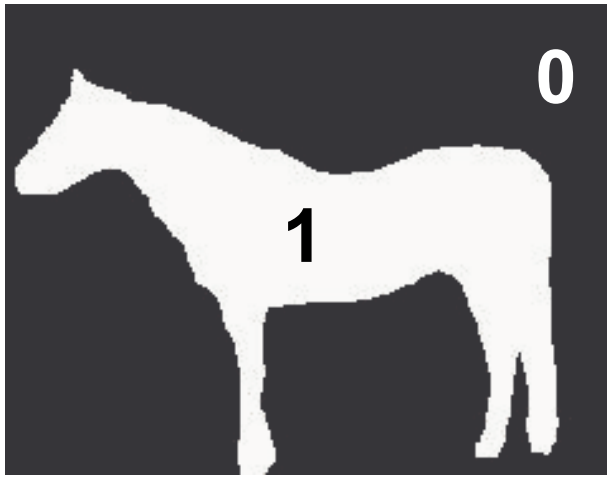
Auto Context

AUTO CONTEXT

Task: Binary segmentation



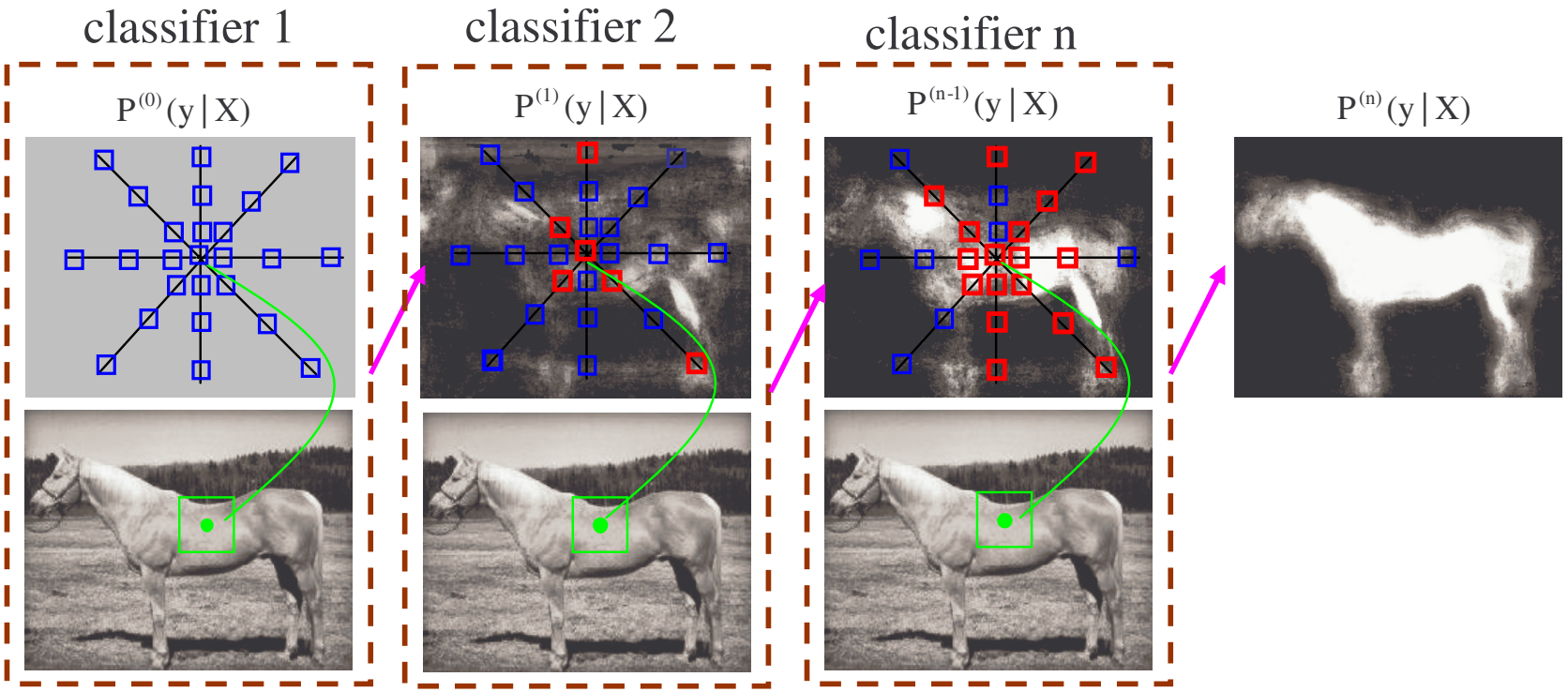
Method: Take feature from each pixel. Train a binary classifier to make decision for each pixel. (data unit: pixel)



AUTO CONTEXT: IDEA

Idea: Heavy use of context and let classifier choose

1. **use feature from nearby region uniformly**
2. **select feature by decision tree**
3. **use response of weak classifier**



AUTO CONTEXT

BETTER FEATURE?
BETTER CLASSIFIER?

Problem of auto-context:

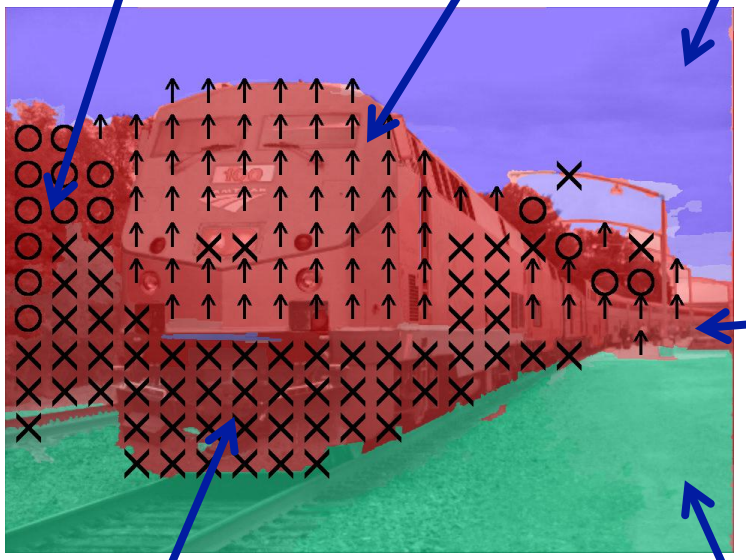
1. **may still lost important context information**
2. **weak feature: pixel value**
3. **no weight on context, heavily relies on classifier**

To get powerful context information from SEGMENTATION

1. **define extent of region for context**
2. **more informative than a single pixel**
3. **take reliability of segments as weight**

GEOMETRIC CONTEXT

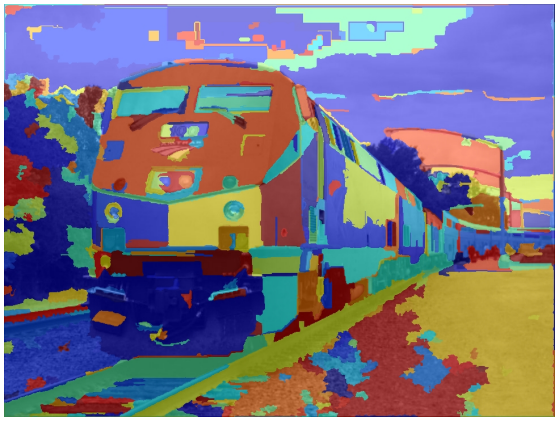
Task: Geometric Labeling (pixel-wise labeling)



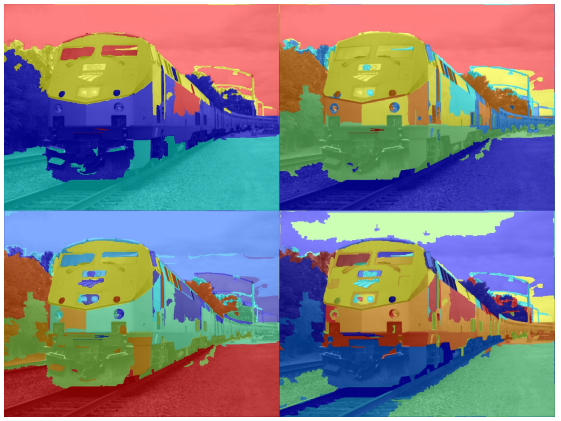
GEOMETRIC CONTEXT

How to get context information?

(a) Input



(b) Superpixels



(c) Multiple Hypotheses

label

#hyp

superpixel

image

vote of hyp

reliability of hyp

$$C(y_i = v|\mathbf{x}) = \sum_{j}^{n_h} P(y_j = v|\mathbf{x}, \mathbf{h}_{ji}) P(\mathbf{h}_{ji}|\mathbf{x})$$

How to get? Training

GEOMETRIC CONTEXT

Generate multiple hypotheses (segmentation) on training image, and label them as "ground", "sky", "vertical", and "mixed".

Compute feature for each hypothesis

Train adaboost (w. decision tree)

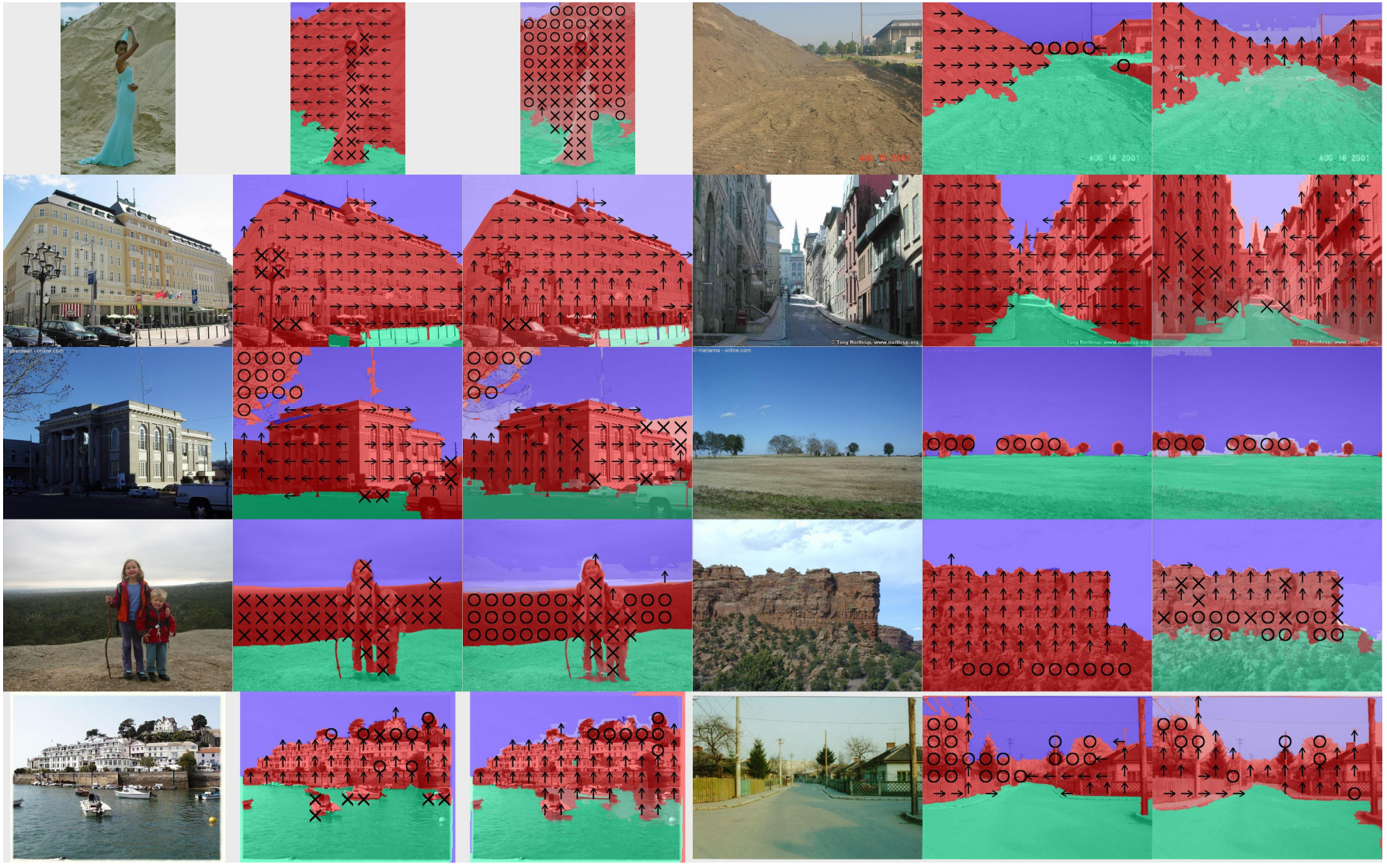
$P(v \mid h, x)$: score(v), v = "ground", "sky", and "vertical"

$P(h \mid x)$: score(v), v = "mixed"

$$C(y_i = v|\mathbf{x}) = \sum_{j}^{n_h} P(y_j = v|\mathbf{x}, \mathbf{h}_{ji}) P(\mathbf{h}_{ji}|\mathbf{x})$$

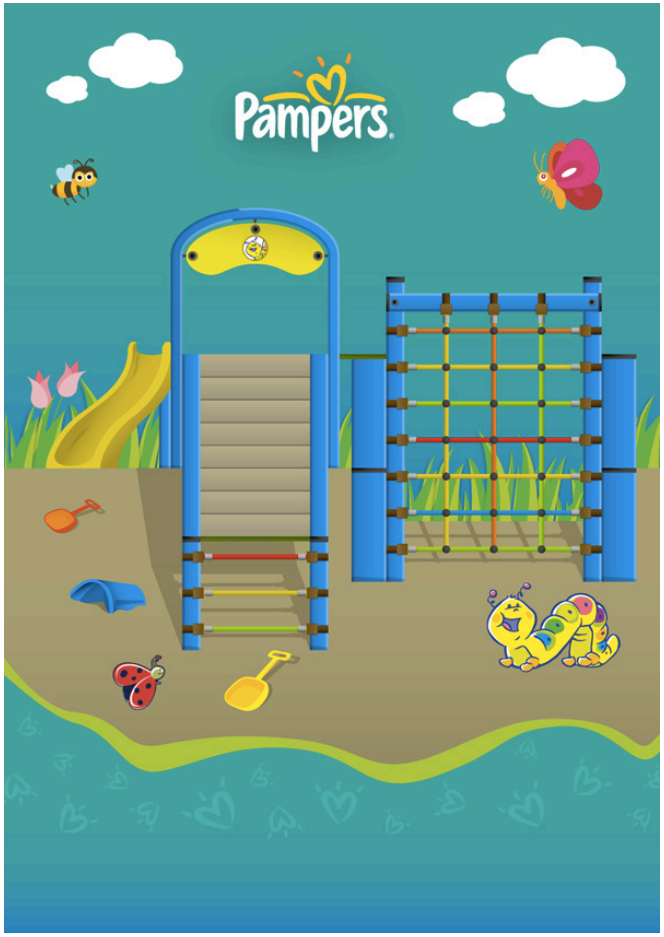
Feature Descriptions	Num
Color	**16**
C1. RGB values: mean	3
C2. HSV values: C1 in HSV space	3
C3. Hue: histogram (5 bins) and entropy	6
C4. Saturation: histogram (3 bins) and entropy	4
Texture	**15**
T1. DOOG filters: mean abs response of 12 filters	12
T2. DOOG stats: mean of variables in T1	1
T3. DOOG stats: argmax of variables in T1	1
T4. DOOG stats: (max - median) of variables in T1	1
Location and Shape	**12**
L1. Location: normalized x and y, mean	2
L2. Location: norm. x and y, 10^{th} and 90^{th} pctl	4
L3. Location: norm. y wrt horizon, 10^{th}, 90^{th} pctl	2
L4. Shape: number of superpixels in region	1
L5. Shape: number of sides of convex hull	1
L6. Shape: $num\ pixels/area(convex\ hull)$	1
L7. Shape: whether the region is contiguous $\in \{0, 1\}$	1
3D Geometry	**35**
G1. Long Lines: total number in region	1
G2. Long Lines: % of nearly parallel pairs of lines	1
G3. Line Intsctn: hist. over 12 orientations, entropy	13
G4. Line Intsctn: % right of center	1
G5. Line Intsctn: % above center	1
G6. Line Intsctn: % far from center at 8 orientations	8
G7. Line Intsctn: % very far from center at 8 orient.	8
G8. Texture gradient: x and y "edginess" (T2) center	2

GEOMETRIC CONTEXT



GEOMETRIC CONTEXT: AUTO IMAGE POP UP

Pop up card



GEOMETRIC CONTEXT: AUTO IMAGE POP UP

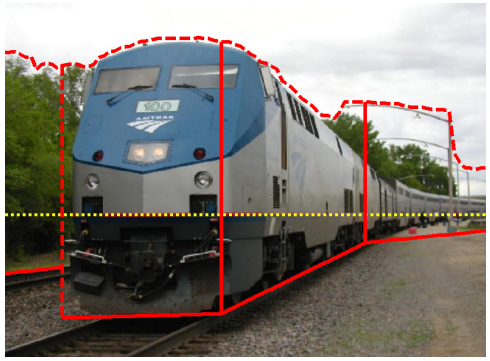
How to do it automatically?

Input image

GC

Fit polylines

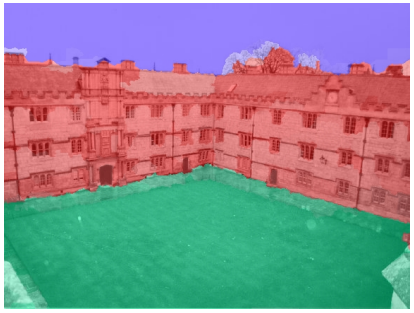
3D model



Cut and fold

GEOMETRIC CONTEXT: AUTO IMAGE POP UP

Input



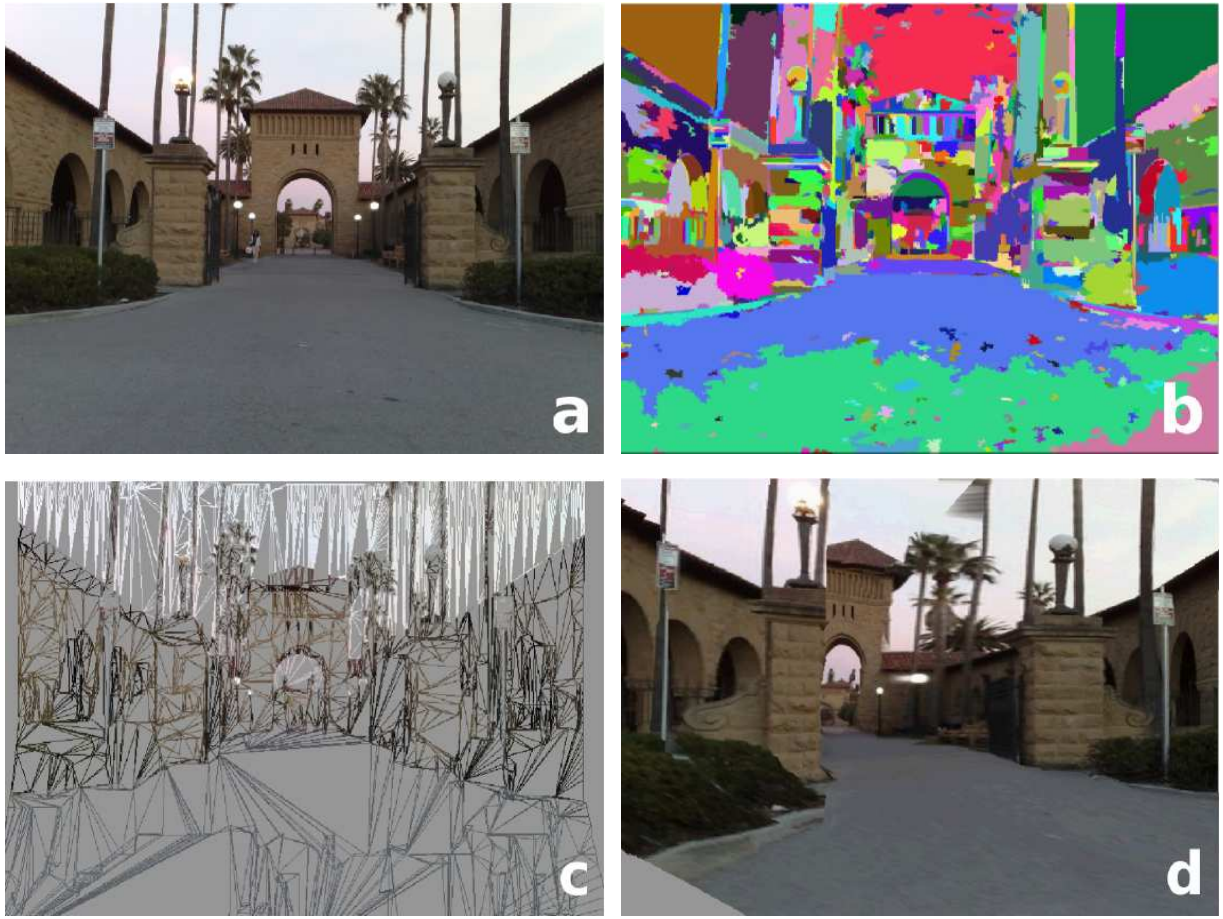
Labels

Novel View

Novel View

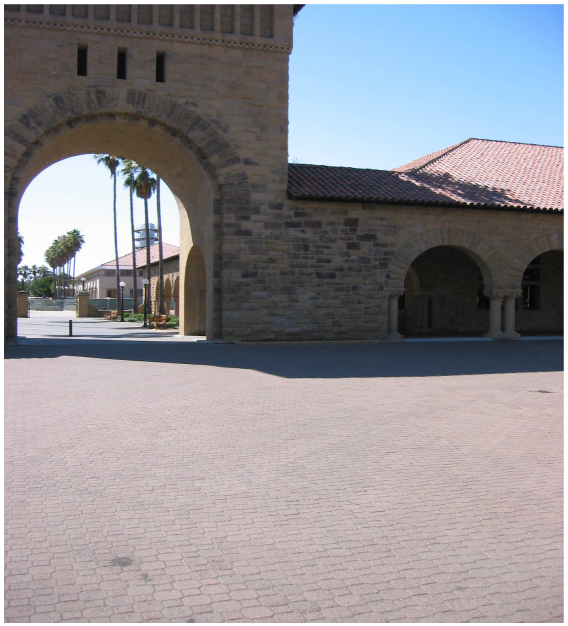
CAN WE GET MORE DETAILS?

From single image to more detailed reconstruction

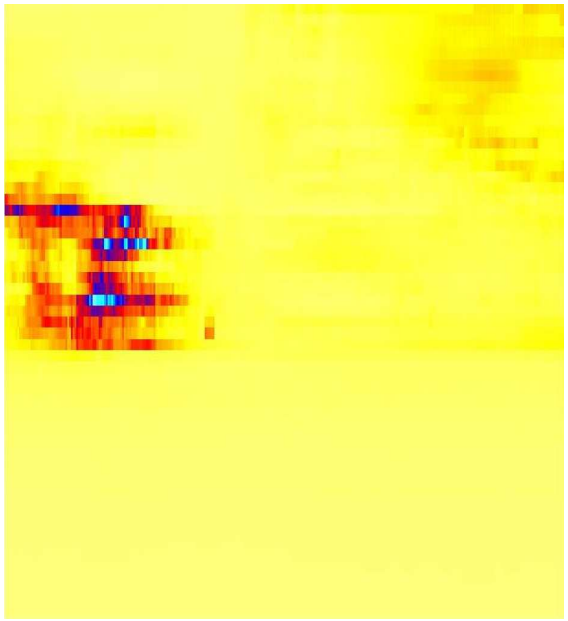


MAKE3D: A MORE DETAILED CONSTRUCTION

With the same technique as GC, we can estimate the 3D position and orientation of each superpixel.



Input image



Estimated depth map

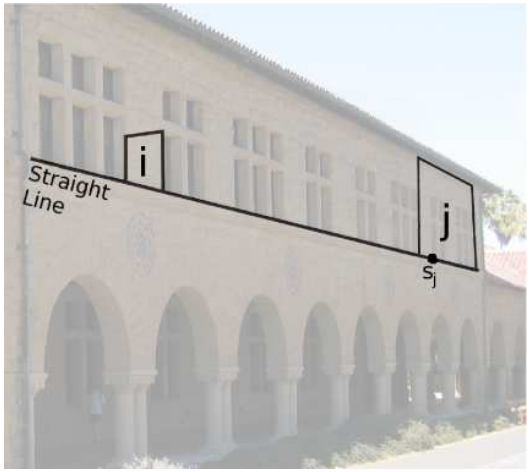
MAKE3D: A MORE DETAILED CONSTRUCTION

Pairwise constraint:

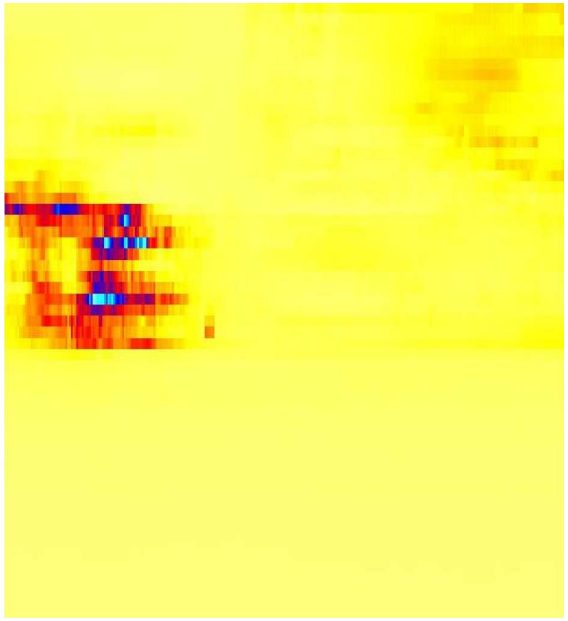
Connectivity

Co-planar

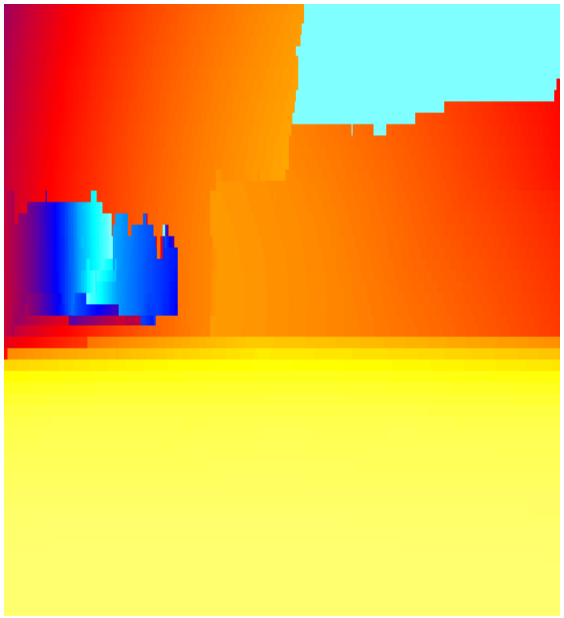
Co-linear



Input image

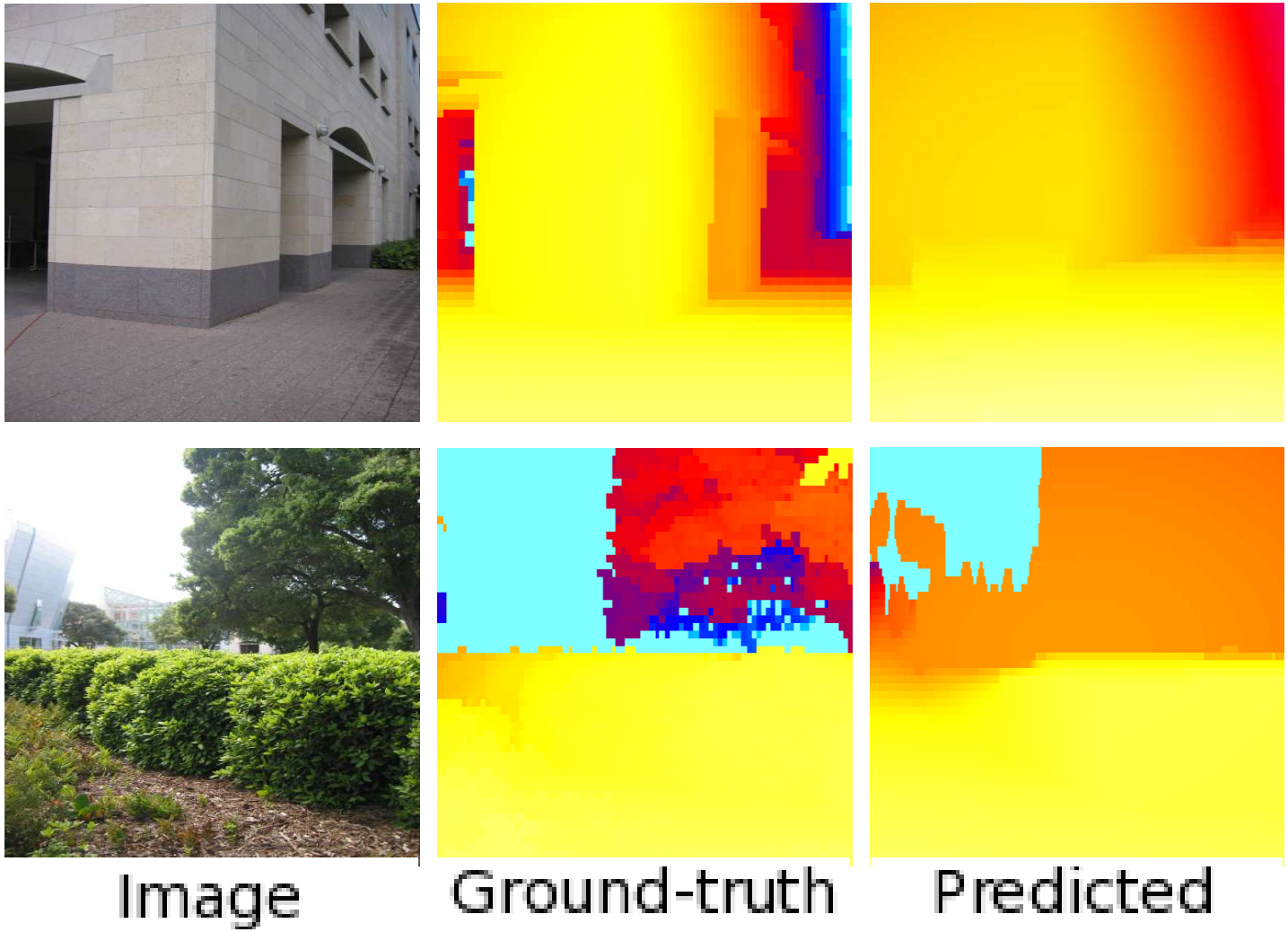


Estimated depth map



MRF optimization

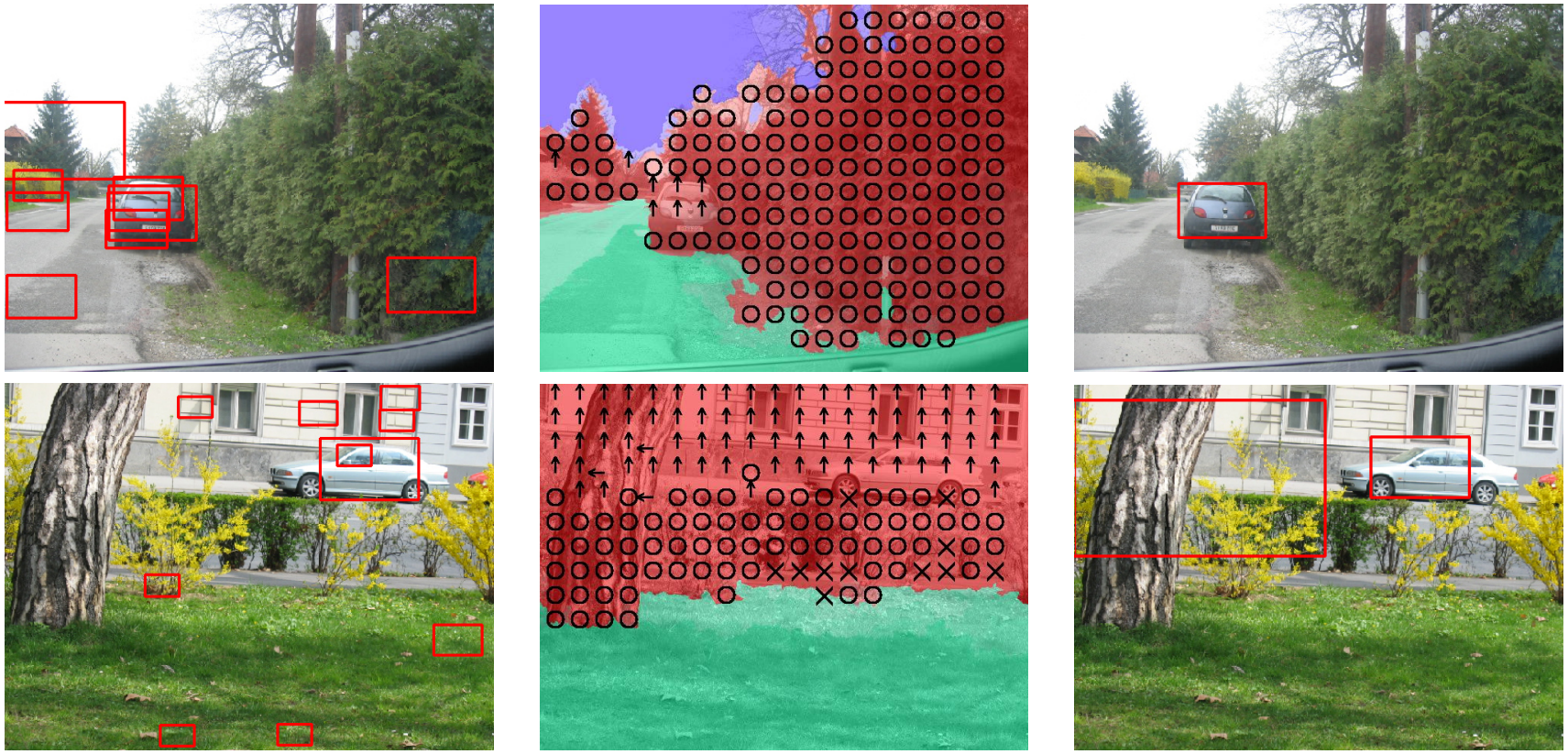
MAKE3D: A MORE DETAILED CONSTRUCTION



MAKE3D: A MORE DETAILED CONSTRUCTION

GEOMETRIC CONTEXT: DETECTION

Refine object detection:

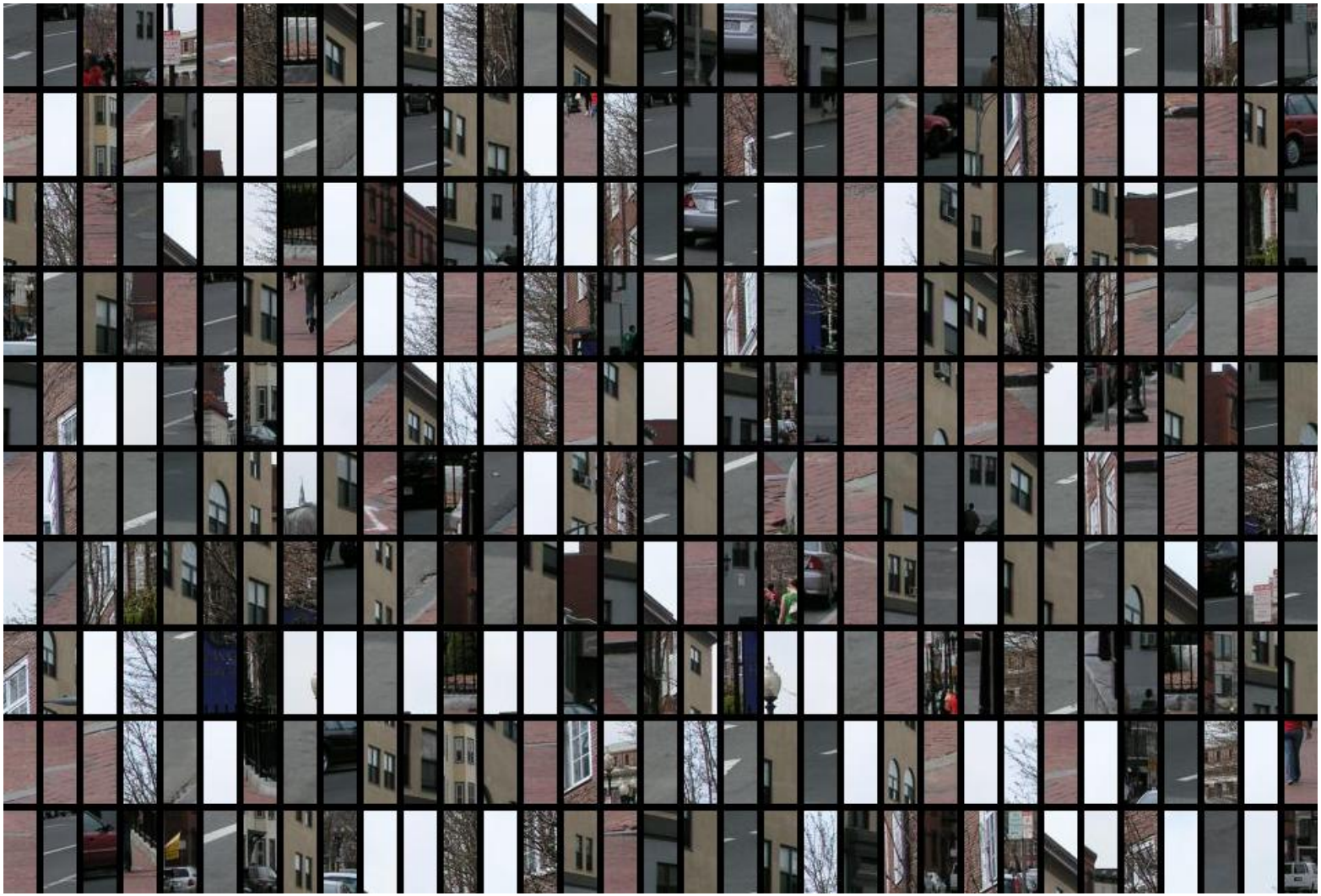


(a) Local Features Only (b) Geometric Labels (c) With Context

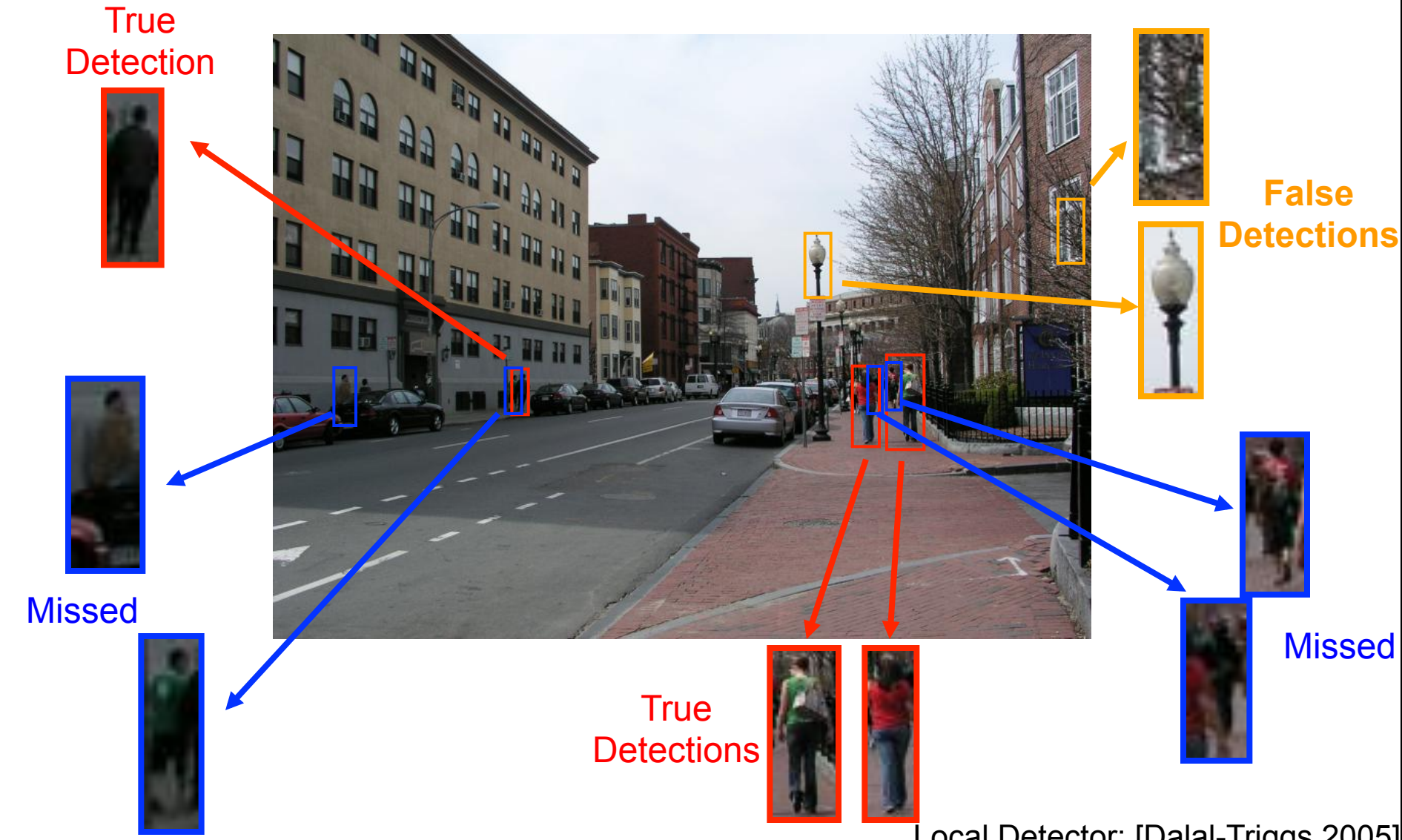
UNDERSTANDING AN IMAGE

TODAY: LOCAL AND INDEPENDENT

WHAT THE DETECTOR SEES

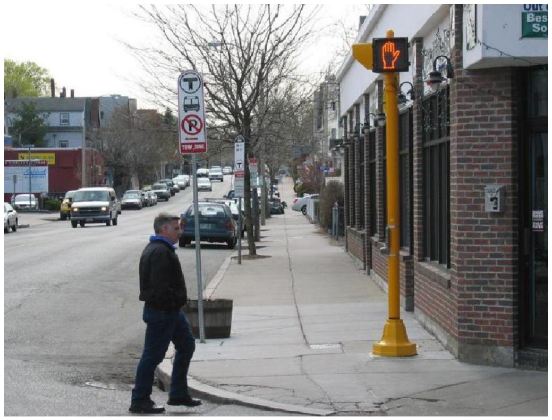


LOCAL OBJECT DETECTION



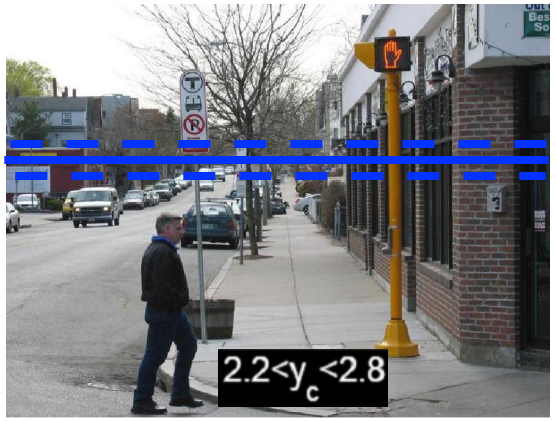
Local Detector: [Dalal-Triggs 2005]

WHAT DOES SURFACE AND VIEWPOINT SAY ABOUT OBJECTS?

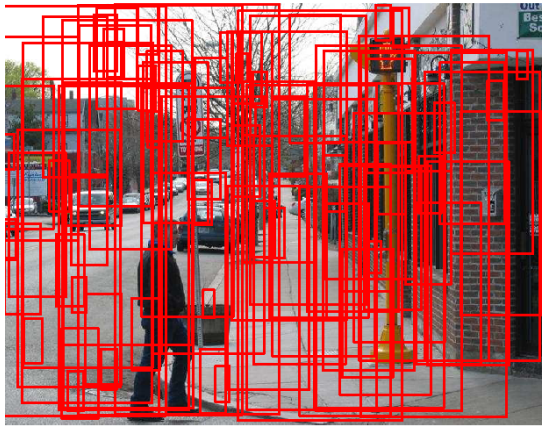


Image

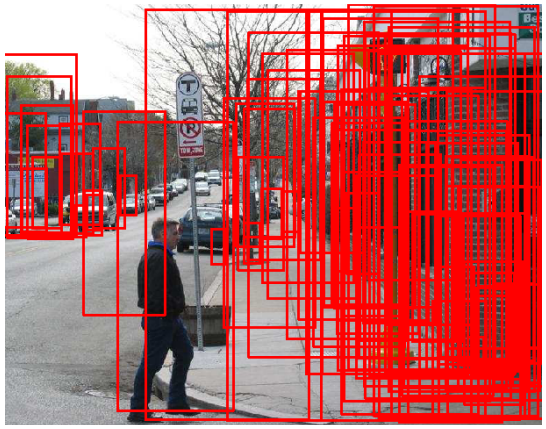
P(surfaces)



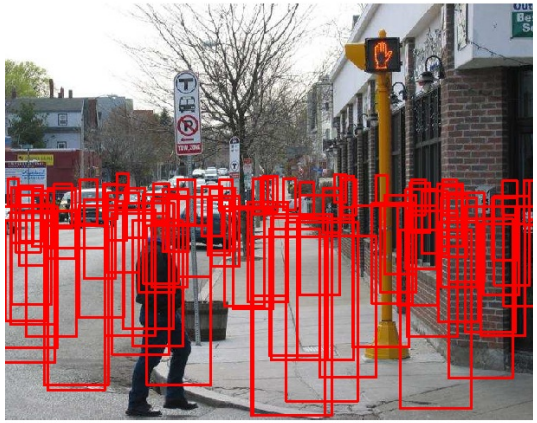
P(viewpoint)



P(object)



P(object | surfaces)



P(object | viewpoint)

OBJECT SIZE IN THE IMAGE

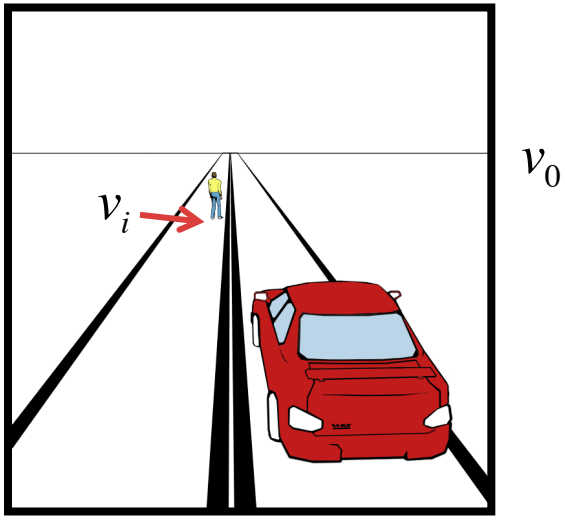
Object Image Height

Camera Height

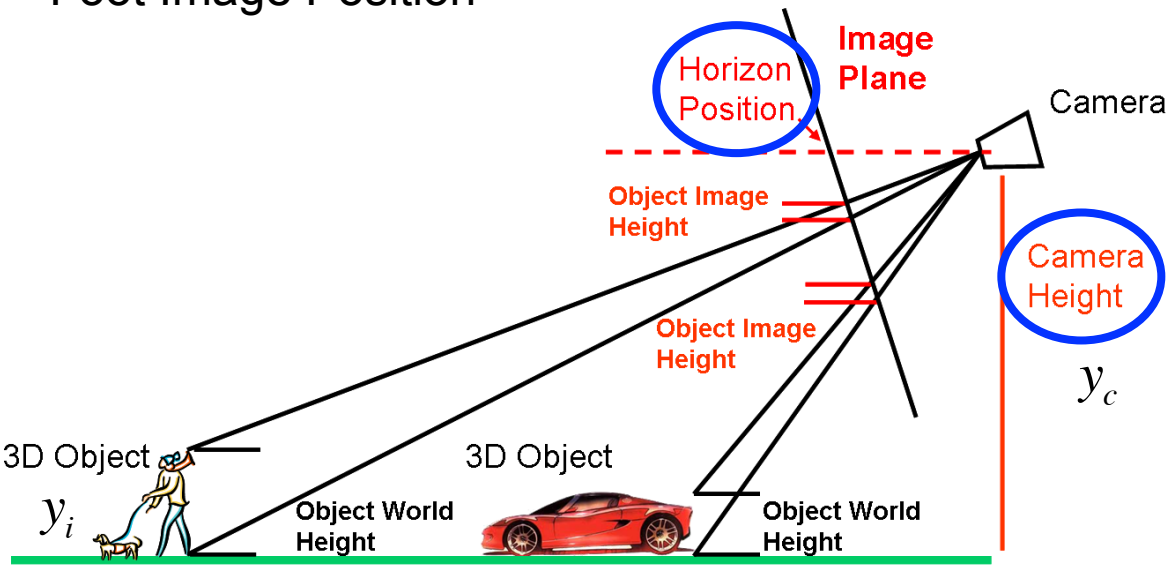
Object 3D Height → $y_i = \dfrac{h_i y_c}{v_i - v_0}$.

Horizon Image Position

Foot Image Position

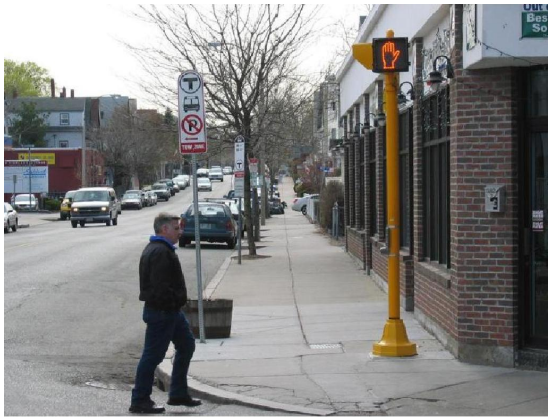


Image



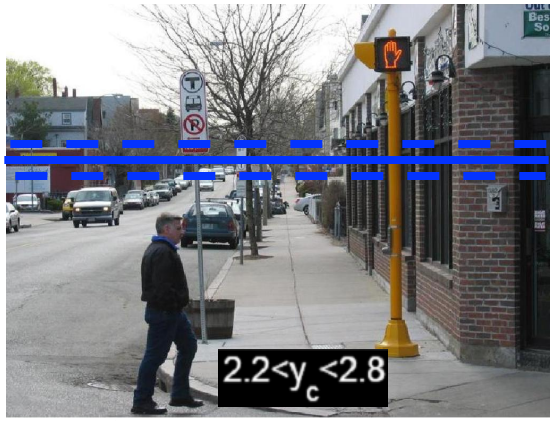
World

WHAT DOES SURFACE AND VIEWPOINT SAY ABOUT OBJECTS?

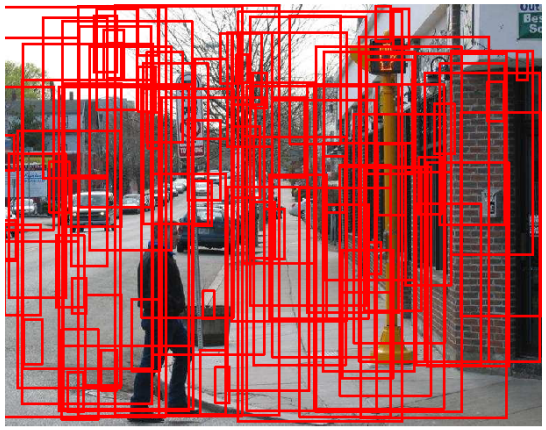


Image

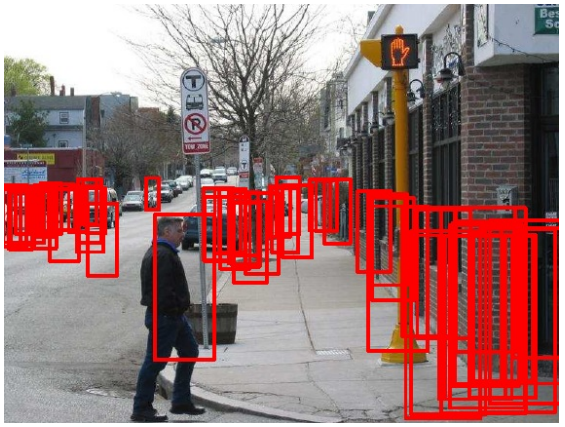
P(surfaces)



P(viewpoint)



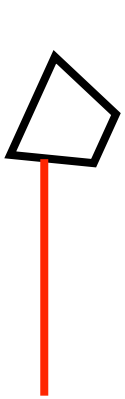
P(object)

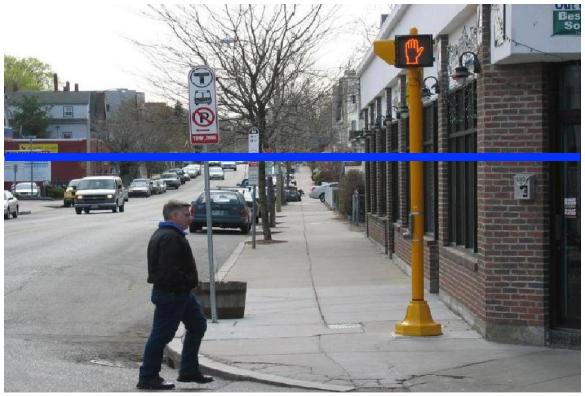


P(object | surfaces, viewpoint)

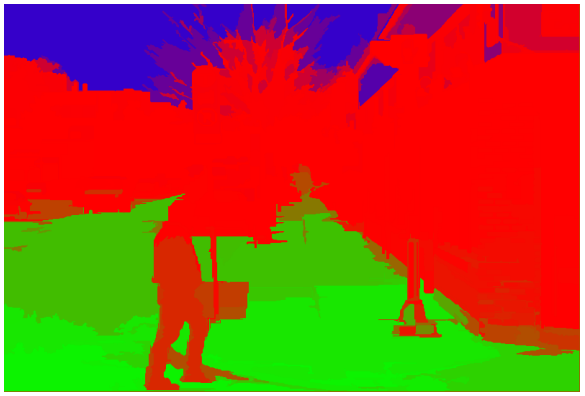
SCENE PARTS ARE ALL INTERCONNECTED

Objects





Camera Viewpoint

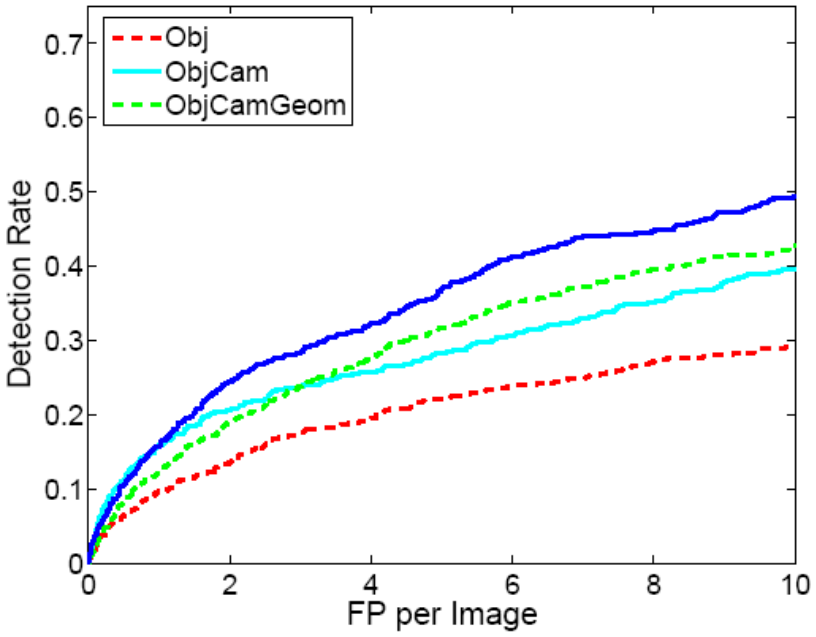


3D Surfaces

A DETECTION FRAMEWORK

This context model between: object, geometric, viewpoint can be used with any object detection system.

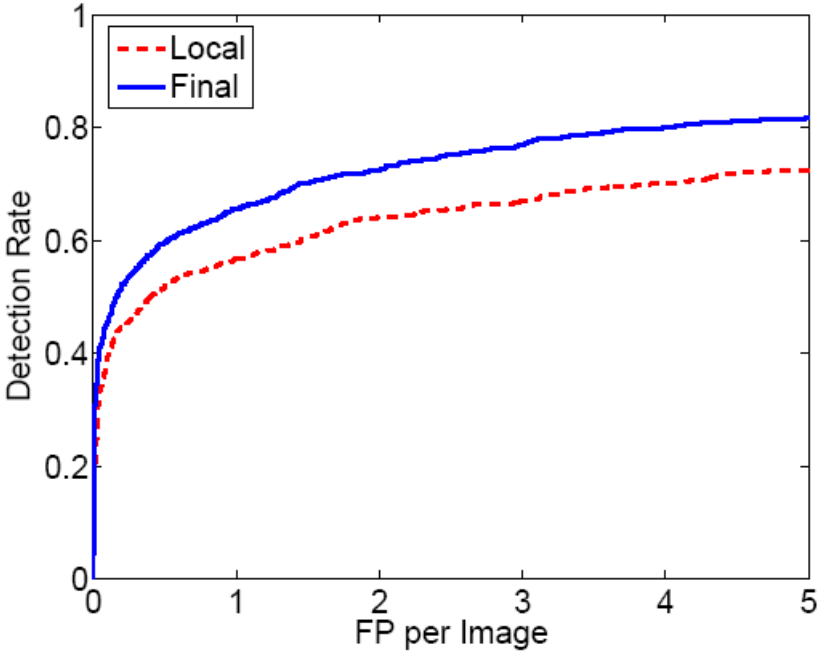
Pedestrian Detection



Local Detector from

[Murphy-Torralba-Freeman 2003]

Pedestrian Detection



Local Detector:

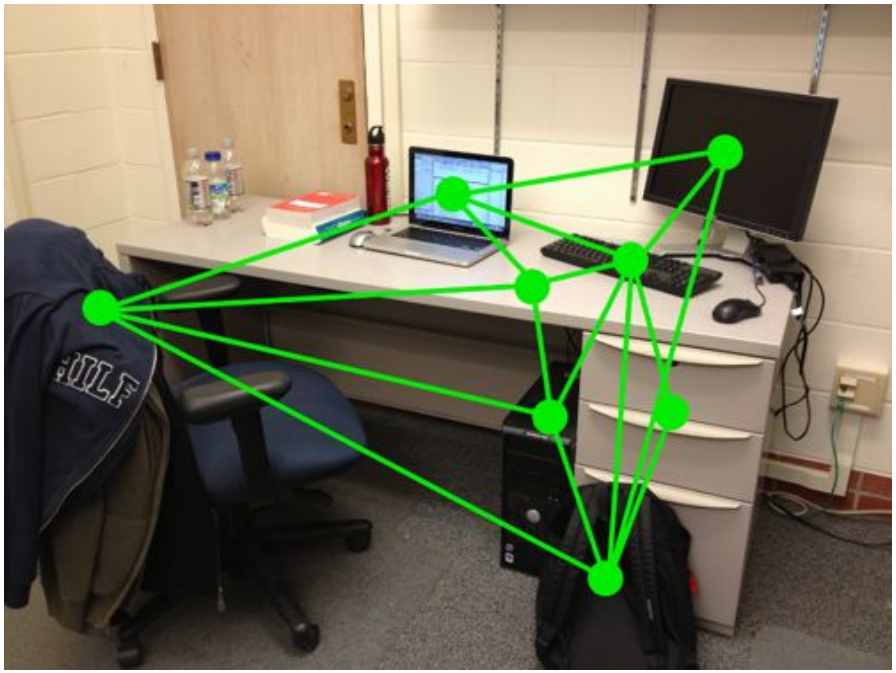
[Dalal-Triggs 2005] (SVM-based)

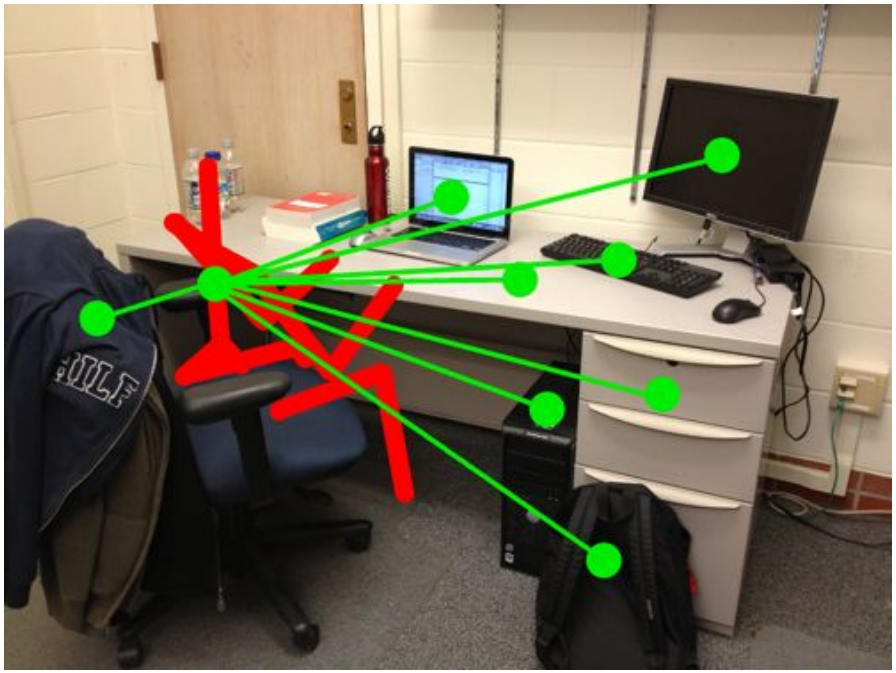
MORE EXAMPLE OF DIFFERENT CONTEXT

Context between human and object

Task: Object labeling on RGBD image

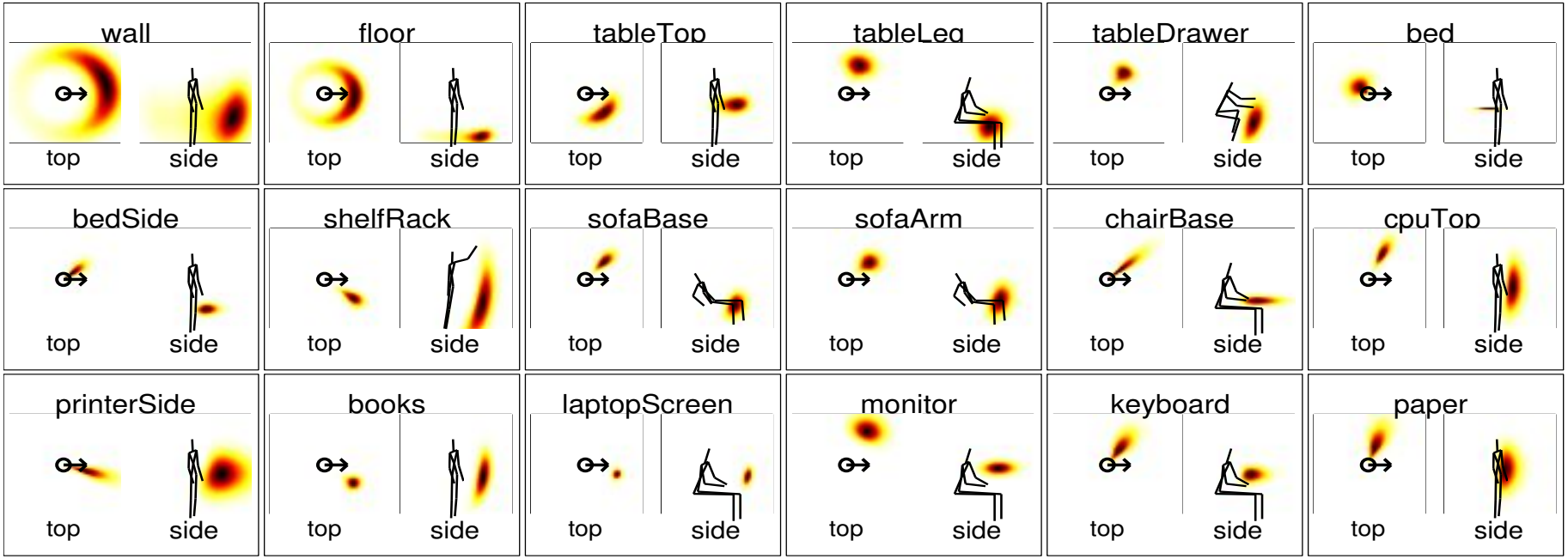
Idea: Use both obj-obj context and human-obj context





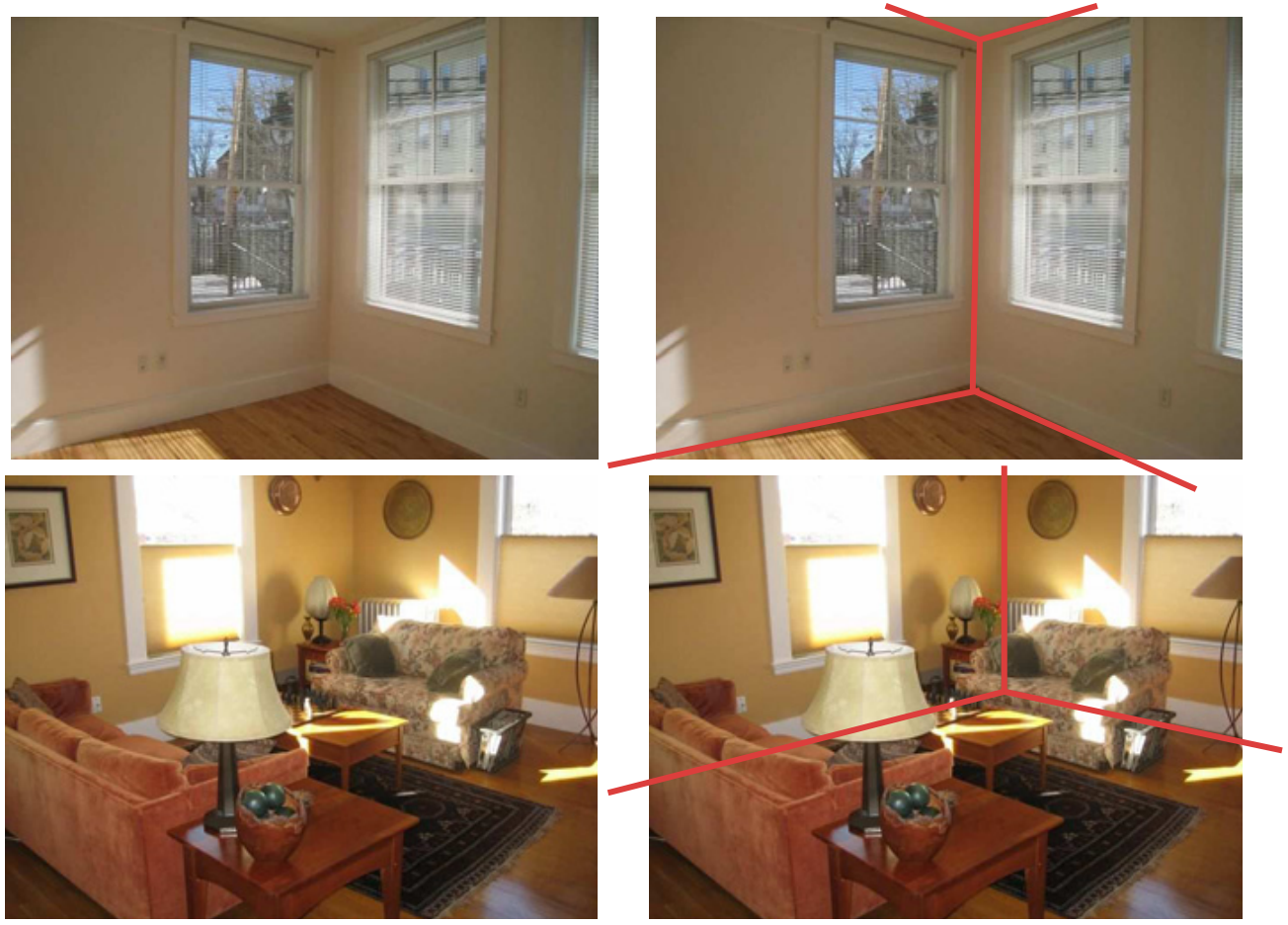
HUMAN-OBJECT CONTEXT

1. **Label all objects**
2. **Infer a human pose interacting with objects**
3. **Refine object labeling base on the inferred human pose**



GEOMETRIC CONTEXT: ROOM LAYOUT

Task: Estimate a box for the room

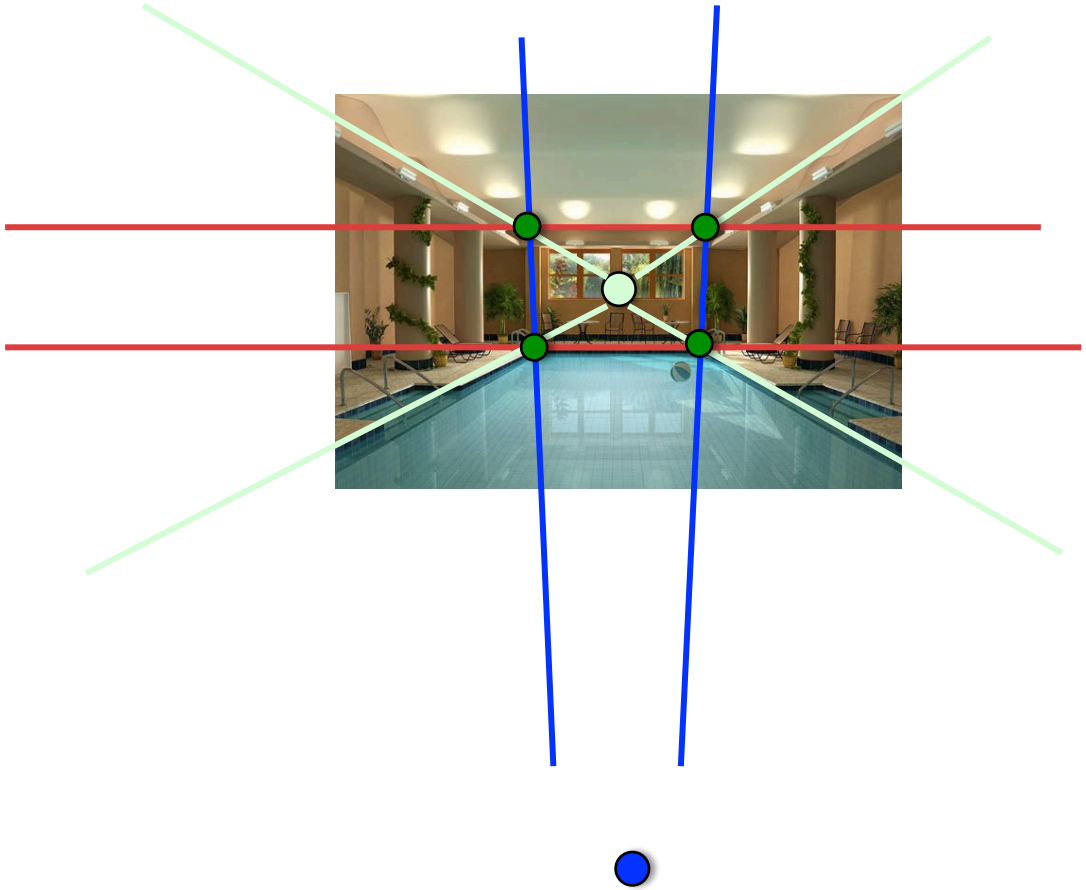


Clutter is a big problem

ROOM LAYOUT ESTIMATION

The # of possible room layout is huge, so sampling first.

Vanishing point

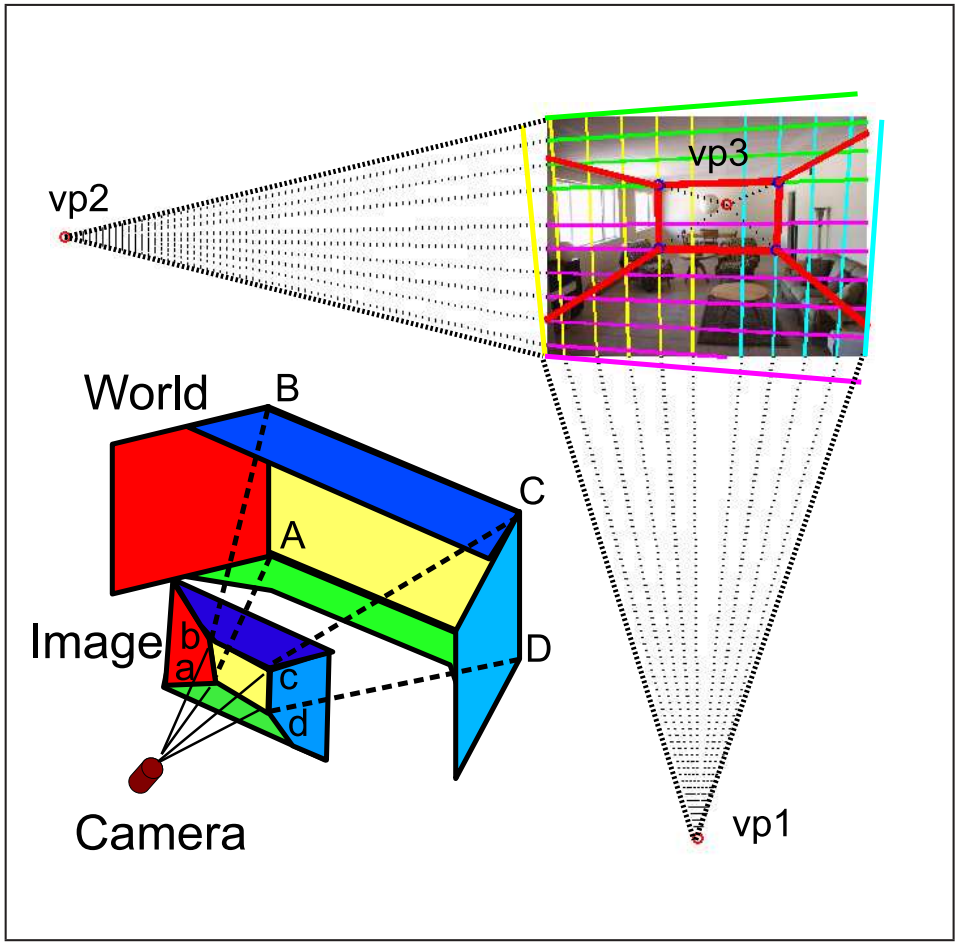


HYPOTHESIS GENERATION

Uniformly sample rays from two vanishing points

Two rays from each vp (4 rays) can generate a room layout hypothesis

The next step:
select the best one



STRUCTURE SVM

Goal: $y^* = \underset{y}{\operatorname{argmax}}\, f(x, y) = \underset{y}{\operatorname{argmax}}\, \omega^T \phi(x, y)$

room layout — trained classifier — image — feature

Training:

$$\min_{w,\xi} \frac{1}{2}||w||^2 + C\sum_i \xi_i$$

$$\text{s.t.} \quad \xi_i \geq 0 \; \forall i, \quad \text{and}$$

$$w^T\psi(x_i, y_i) - w^T\psi(x_i, y) \geq \Delta(y_i, y) - \xi_i$$

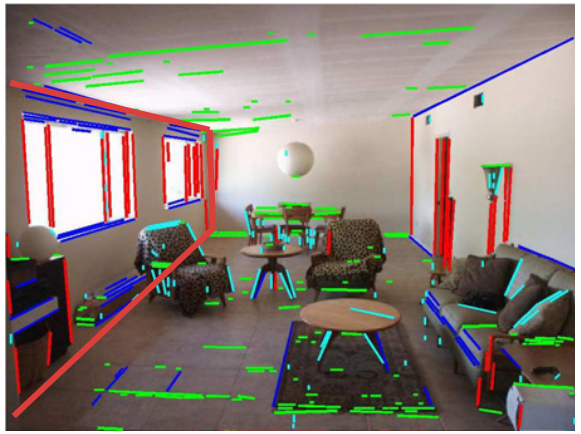
$$\forall i, \forall y \in \mathbb{Y} / \; y_i$$

Line seg. belong to the two vanishing points of F_k

Feature:

$$f_l(F_k) = \frac{\sum_{l_j \in C_k} |l_j|}{\sum_{l_j \in L_k} |l_j|}$$

Line segments in F_k



WHERE IS GC?

Use Geometric context to label pixel with label: “left wall”, “right wall”, “middle wall”, “ceiling”, “floor”, and “object”.

Reduce the weight of line segments in “object” region

PROBLEM

Room layout hypotheses are sampled uniformly, which cannot guarantee global (or even local) minimum.

What should we have to get global minimum?

1. **a quick way of computing feature**

   "Integral Geometry"

2. **a quick way of searching**

   "Branch and bound"

INTEGRAL GEOMETRY

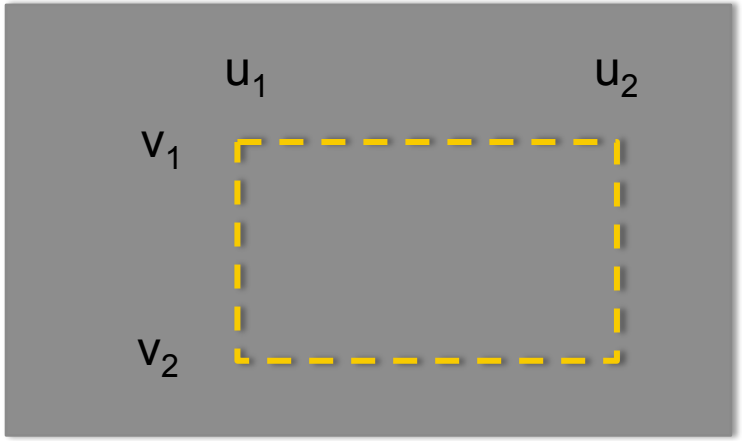
Suppose we have a pixel-wise feature map $f(x, y)$, and

The feature of any box region equals to the sum of features of included pixels,

$$F(u_1, v_1, u_2, v_2) = \sum_{x=u_1}^{u_2} \sum_{y=v_1}^{v_2} f(x, y)$$

Compute feature for M boxes with N pixels would cost O(MN)

How to compute feature for any box in constant time? O(M)



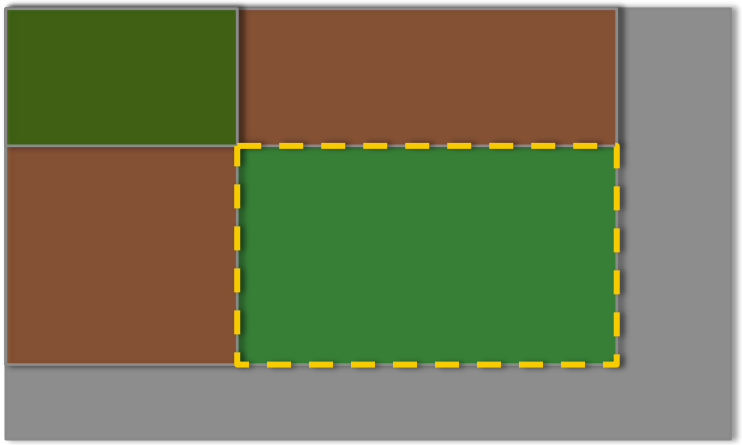
INTEGRAL GEOMETRY

1. **Compute integral image in O(N)**

$$I(u,v) = \sum_{x=1}^{u}\sum_{y=1}^{v} f(x,y)$$

2. **Compute feature for box region:**

$$\begin{aligned} F(u_1,v_1,u_2,v_2) = {} & I(u_2,v_2) \\ & -I(u_1-1,v_2)-I(u_2,v_1-1) \\ & +I(u_1-1,v_1-1) \end{aligned}$$



Original

5	**2**	**3**	4	1
1	**5**	**4**	2	3
2	2	1	3	4
3	5	6	4	5
4	1	3	2	6

Integral

5	7	10	14	15
6	13	**20**	26	30
8	17	25	34	42
11	25	39	52	65
15	30	47	62	81

$5 + 2 + 3 + 1 + 5 + 4 = 20$

Original

5	2	3	4	1
1	**5**	**4**	**2**	3
2	**2**	**1**	**3**	4
3	5	6	4	5
4	1	3	2	6

$5 + 4 + 2 + 2 + 1 + 3 = 17$

Integral

5	7	10	**14**	15
6	13	20	26	30
8	17	25	**34**	42
11	25	39	52	65
15	30	47	62	81

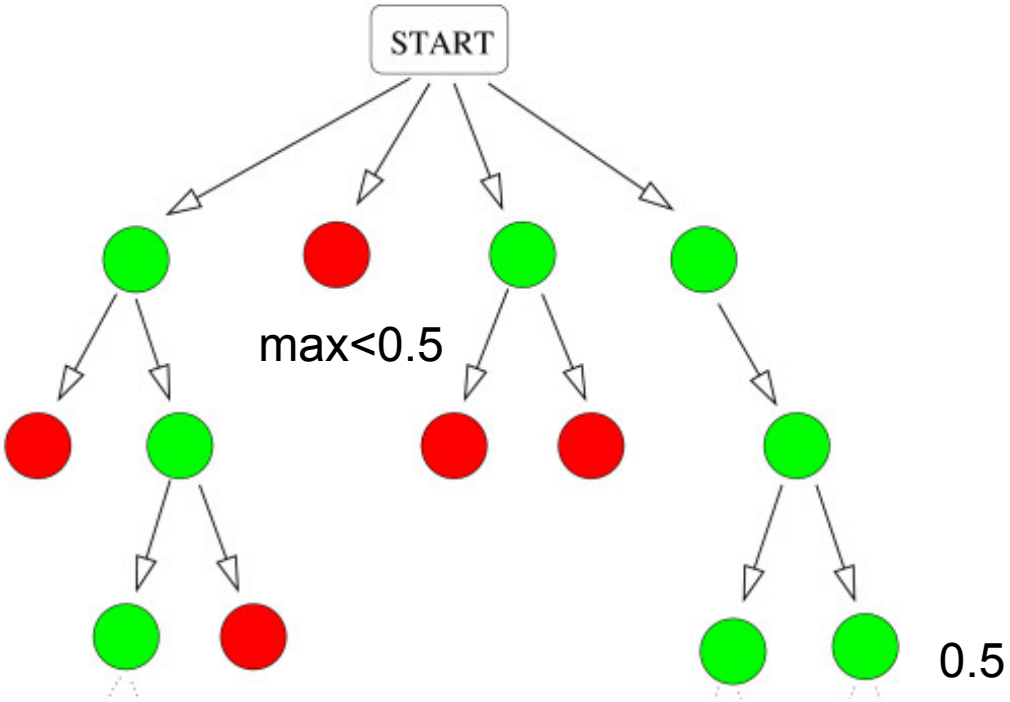
$34 - 14 - 8 + 5 = 17$

BRANCH AND BOUND

Task: Search for the rectangle give the best score

Branching. **Dividing a space of candidate rectangles into subspaces**

Bounding. **Pruning subspaces with a highest possible score lower than some guaranteed score in other subspaces**



BRANCH AND BOUND

To use branch-and-bound for given quality function *f*, we need to define upper bound function $\hat{f}$

$$i) \quad \hat{f}(\mathcal{R}) \geq \max_{R \in \mathcal{R}} f(R),$$

$$ii) \quad \hat{f}(\mathcal{R}) = f(R), \quad \text{if } R \text{ is the only element in } \mathcal{R}.$$

Computation of upper bound has to be efficient to make the search efficient.

BRANCH AND BOUND EXAMPLE: DETECTION

Assume the previous definition of box feature, and all dimension of feature are positive.

We trained a linear SVM $B^* = \underset{B}{\operatorname{argmax}}\, \omega^T F(B)$

***Upper bound function*:**

If R_{max} the largest rectangle and by R_{min} the smallest rectangle contained in a parameter region R,

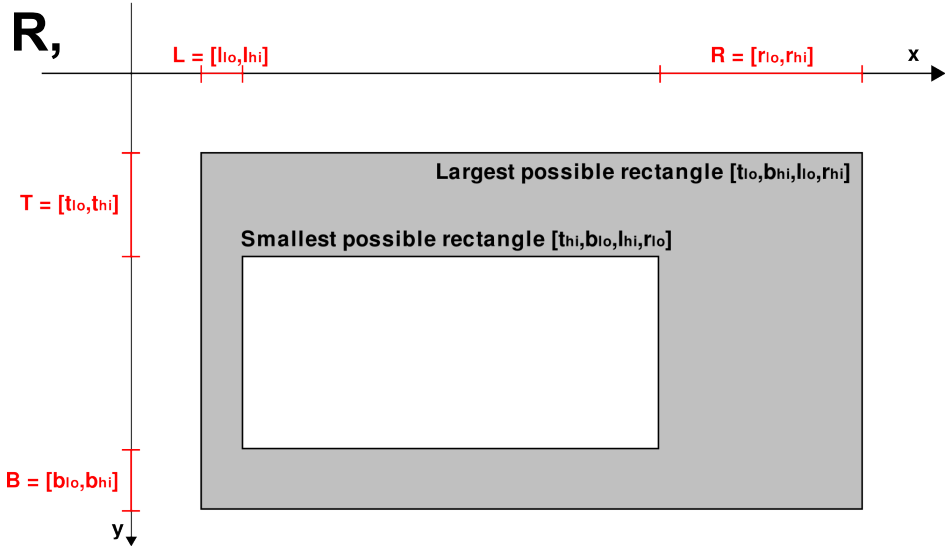
$$\hat{f}(\mathcal{R}) := f^+(R_{\max}) + f^-(R_{\min})$$

All dimension with positive weight

$$f(R) = f^+(R) + f^-(R)$$

$$f^+(R_{\max}) > f^+(R)$$

$$f^-(R_{\min}) > f^-(R)$$



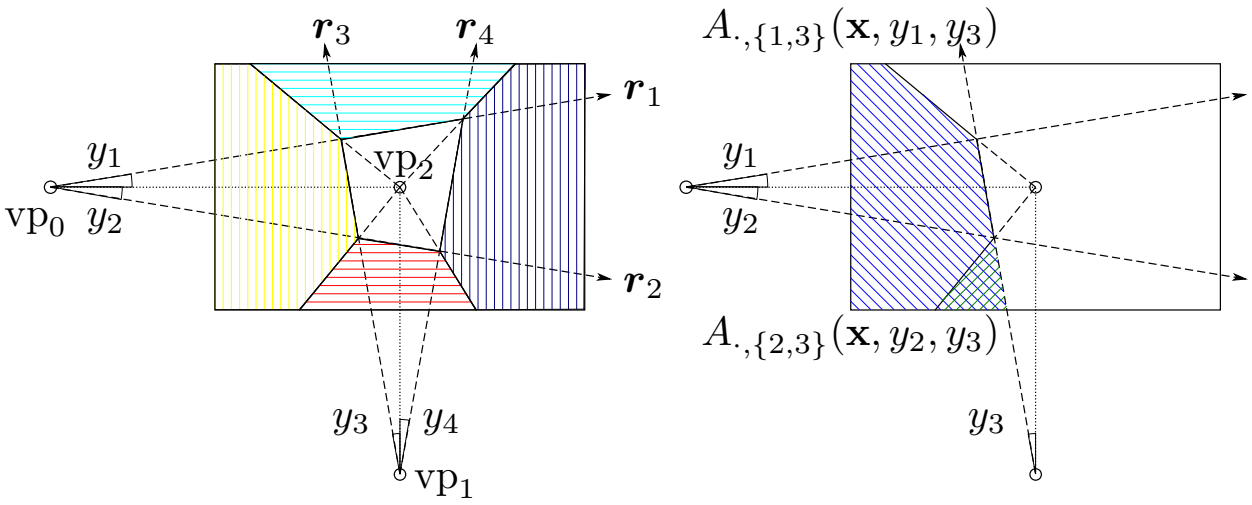
BACK TO: LAYOUT ESTIMATION

Exactly the same as object detection.

Feature:

1. **each pixel: geometric context trained before (all positive).**
2. **each wall: sum of feature of pixel inside**
3. **room layout: concatenation of feature of all walls**

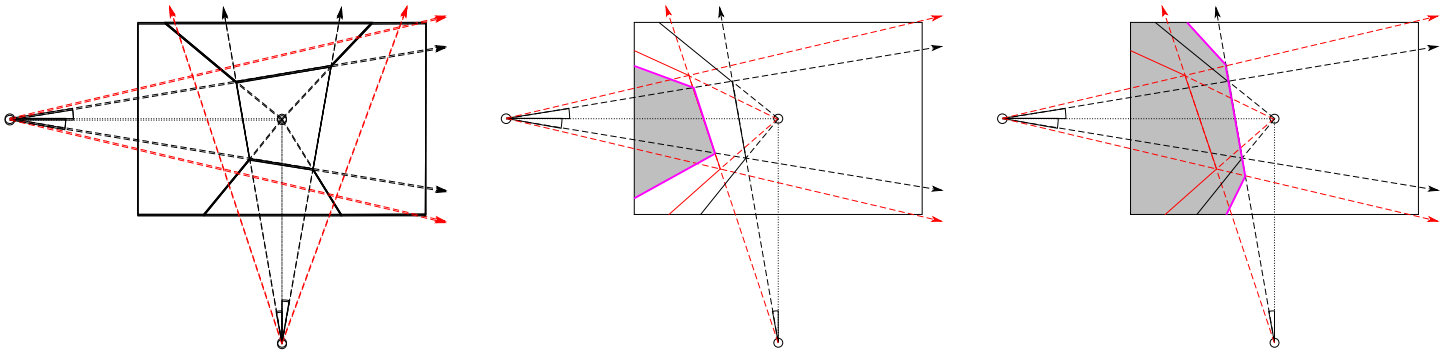
Feature of a region: integral image



ROOM LAYOUT EASTIMATION

Set of room layout: 4 ranges

Define maximal and minimal region for each wall



(a) Front Wall. (b) Minimal left wall. (c) Maximal left wall.

Branch and Bound

Usually 20 times faster,
and slightly better

$$f^+(x,y) = \sum_{\{(i,\alpha,r):i\in\{o,g\},\alpha\in\mathcal{F},w_{i,\alpha,r}>0\}} w_{i,\alpha,r}\phi_{i,\alpha,r}(x,y_\alpha),$$

$$f^-(x,y) = \sum_{\{(i,\alpha,r):i\in\{o,g\},\alpha\in\mathcal{F},w_{i,\alpha,r}\leq 0\}} w_{i,\alpha,r}\phi_{i,\alpha,r}(x,y_\alpha).$$

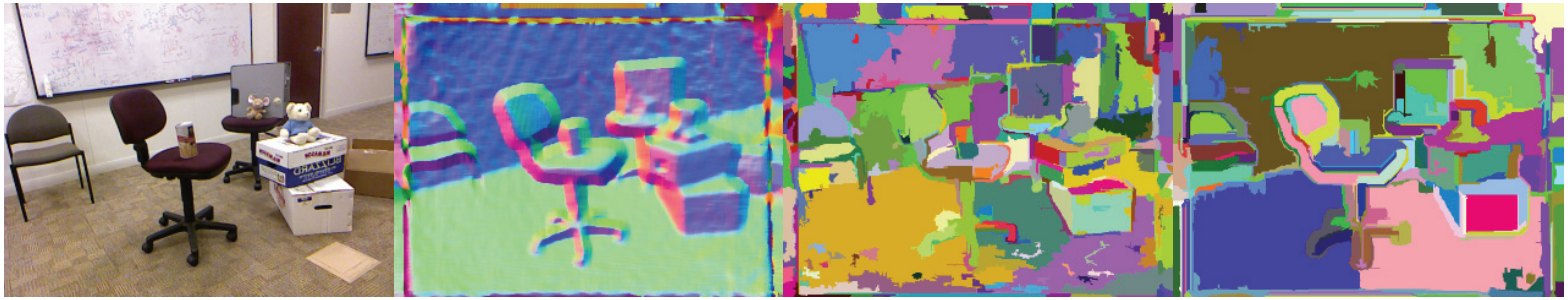
$$\hat{y} = \underset{y\in\mathcal{Y}}{\operatorname{argmax}}\, f^+(x,y) + f^-(x,y)$$

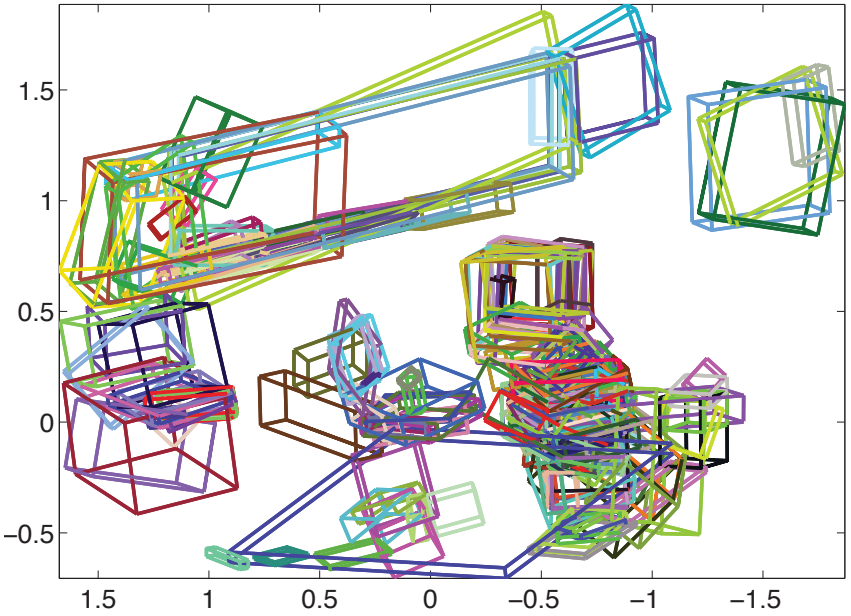
EXAMPLE: CONTEXT + BRANCH&BOUND

Task: Fit cuboids on RGBD image

CUBOID MATCHING IN RGBD IMAGE

Fit cuboid directly from image & 3D information





How to find those correct ones?

Minimizing error will give trivial empty solution.

CUBOID MATCHING IN RGBD IMAGE

Context information for regularization:

1. **surface coverage**

   the fitted cuboids should cover a large region.

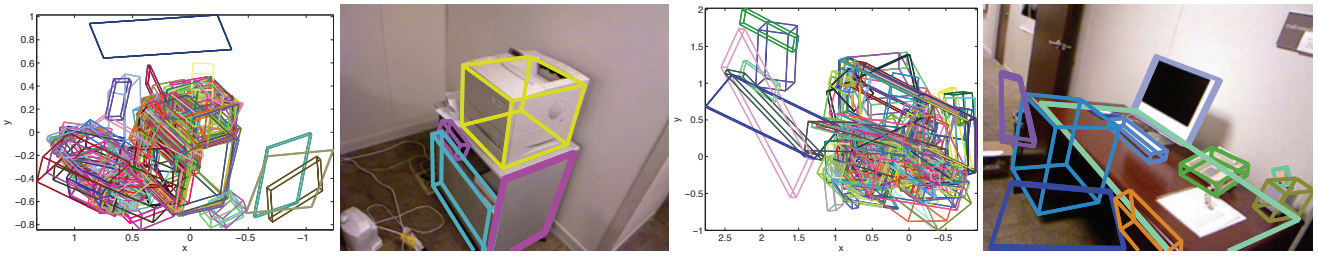
2. **volume exclusion**

   two cuboids shall not intersect, or the intersection should be minimized.

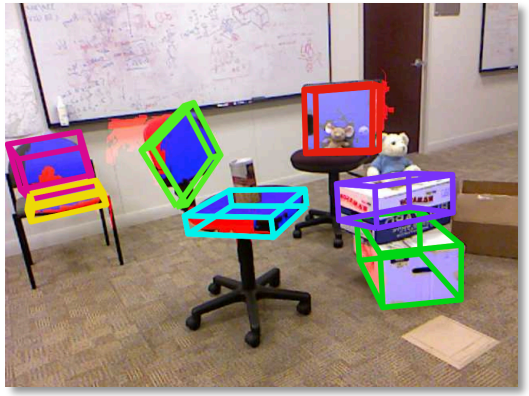
3. **occlusion constraints**

   a cuboid shall not be fully occluded by others

4. **small # of cuboids**



Context Model: Mixed IP



Unary Cost | Intersecti on | Number | Coverage | Occlusion

$$\min_{\mathbf{x}}\{U(\mathbf{x}) + \lambda P(\mathbf{x}) + \mu N(\mathbf{x}) - \gamma A(\mathbf{x}) + \xi O(\mathbf{x})\}$$

$$\min\{ \min c^T x \quad y_k + \xi \sum_{\{i,j\}} q_{i,j} w_{i,j} \}$$

s.t. z

$\forall\{i,$

$y_k \leq s.t. Ax \leq b$ superpixel k

$w_{i,j} \leq x_i,$

$\forall\{i,j\}$ that $x_i \in \{0,1\}, i \in I_z$

$x_i + x_i \leq$

$x_i = 0$ or 1. All variables are nonnegative.

2^{200} solution candidates

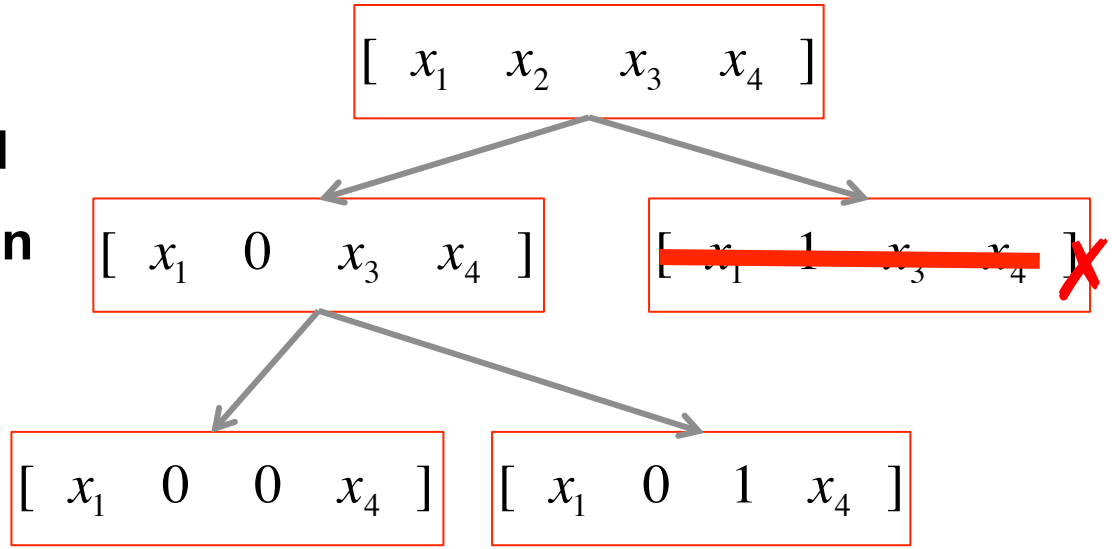
Branch and Bound Mixed Integer Programing
Global Optimal Solution in 10 seconds (Code Available)

CONTEXT MODEL: MIXED IP

$$x^* = \arg\min_x c^T x,$$
$$s.t. Ax \le b, x_i \in \{0,1\}, i \in I_z$$

Branch and Bound:

1. **Fix one x_i to be 0 or 1**
2. **Lower bound function**



$$f(X) = \min c^T x,$$
$$s.t. Ax < b,$$
$$x_i \in \{0,1\}, i \in I_z,$$
$$x_m = 0, m \in I_0, x_n = 1, n \in I_1,$$

$$\hat{f}(X) = \min c^T x,$$
$$s.t. Ax < b,$$
~~$x_i \in \{0,1\}, i \in I_z,$~~
$$x_m = 0, m \in I_0, x_n = 1, n \in I_1,$$

Typical linear programming
Call MATLAB

$$f([\ 1 \quad 0 \quad 0 \quad 1\ ]) = 0.5$$

$$\hat{f}([\ x_1 \quad 1 \quad x_3 \quad x_4\ ]) = 0.7$$

Do NOT need to check any solution in

$$[\ x_1 \quad 1 \quad x_3 \quad x_4\ ]$$