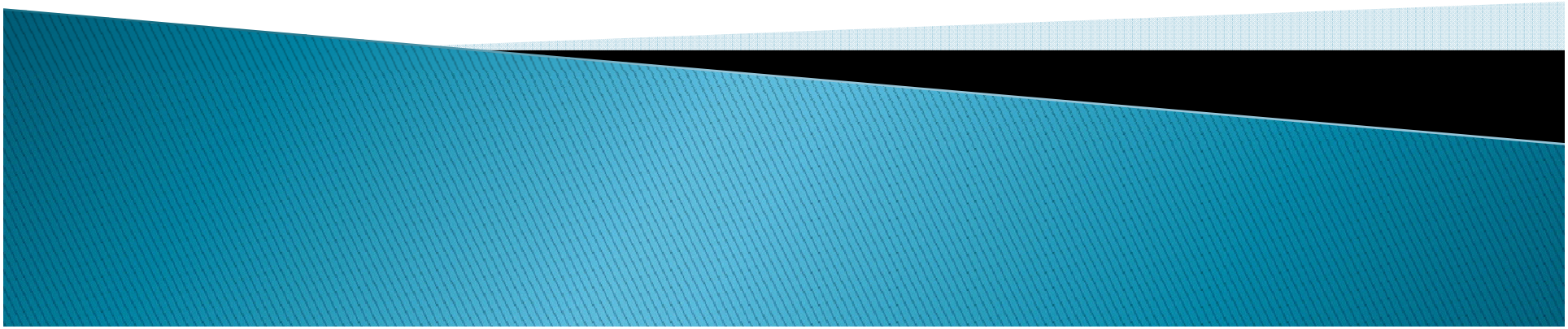
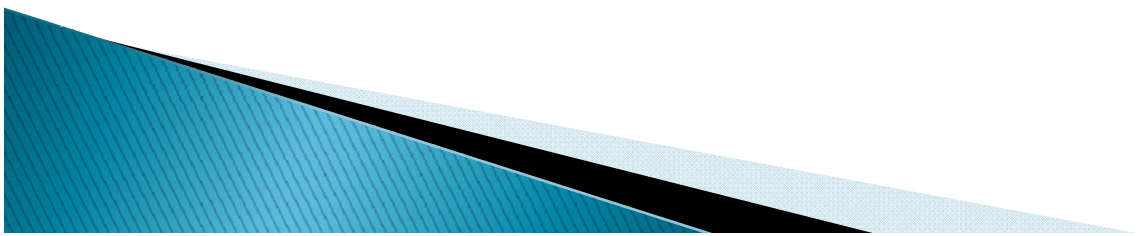
Context

Andrew Harp



Outline

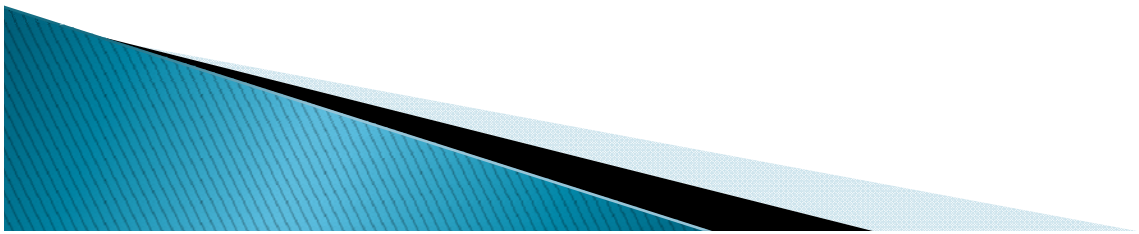
- What is context?
- Learning Spatial Context: Using Stuff to Find Things
  - Geremy Heitz and Daphne Koller
- Putting Objects in Perspective
  - Derek Hoiem, Alexei A. Efros and Martial Hebert



What is Context?

- Linguistically, Context refers to the conditions in which something exists or occurs.

- Context can be recursive!



Examples of context:

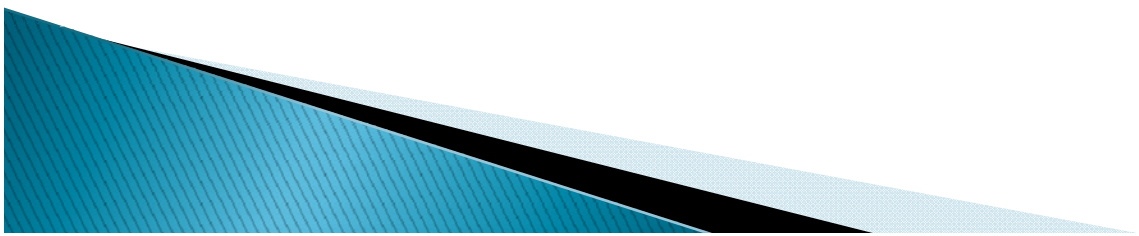
What does “compound” mean in these examples?

- compound
- compound

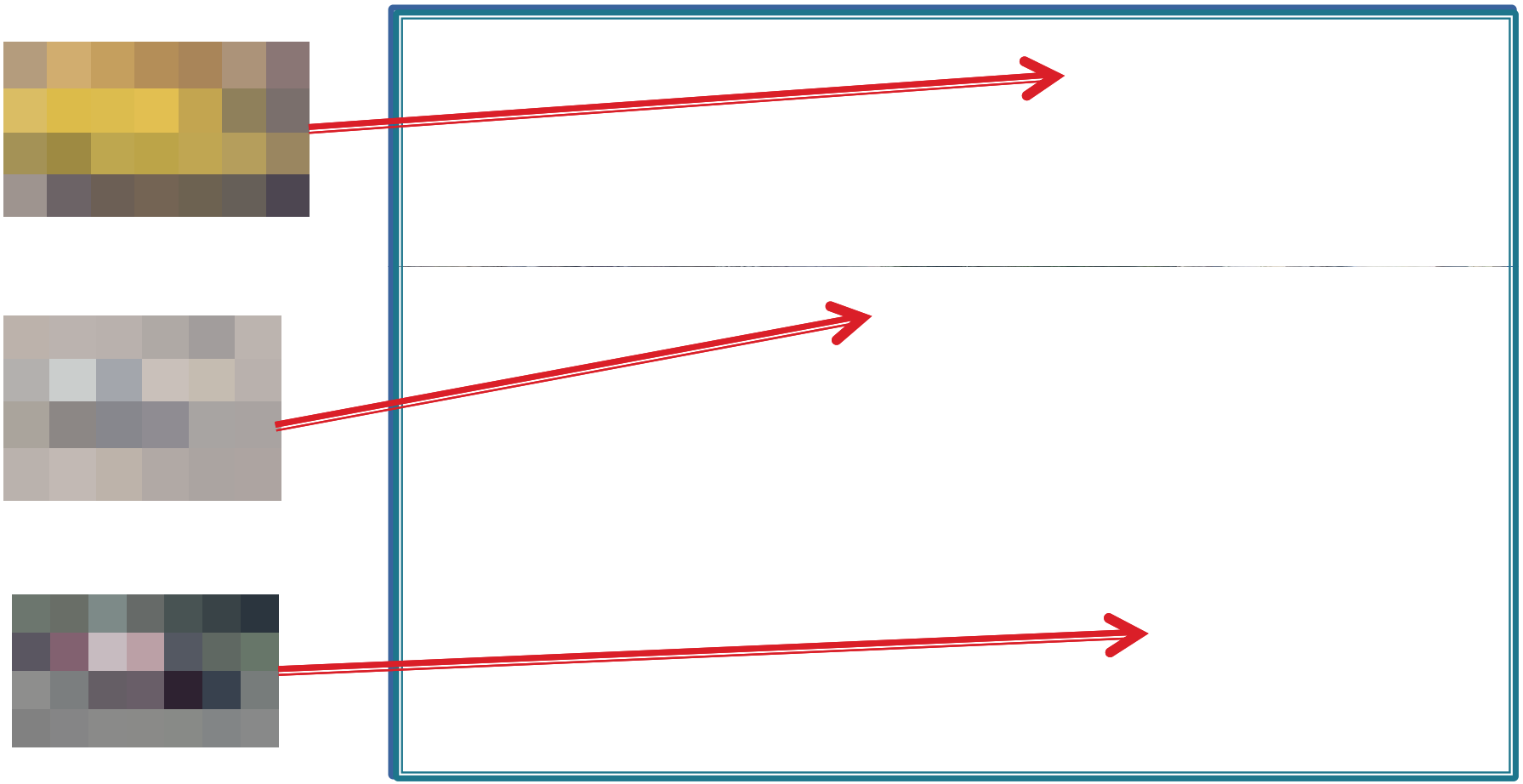
Examples of context:

What does “compound” mean in these examples?

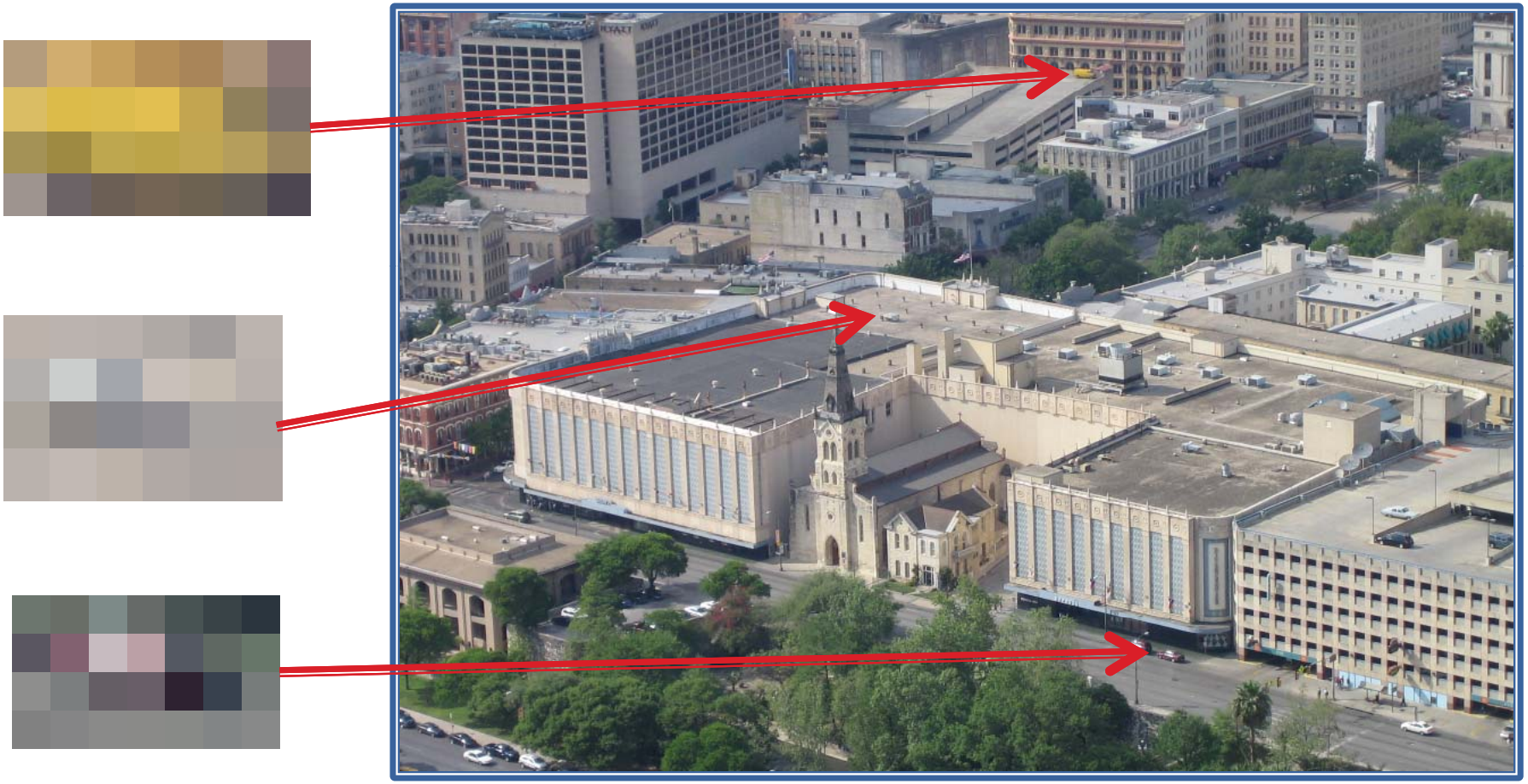
- The villain’s compound is heavily guarded.
- She suffered a compound fracture from the fall.



Examples of Context (contd)

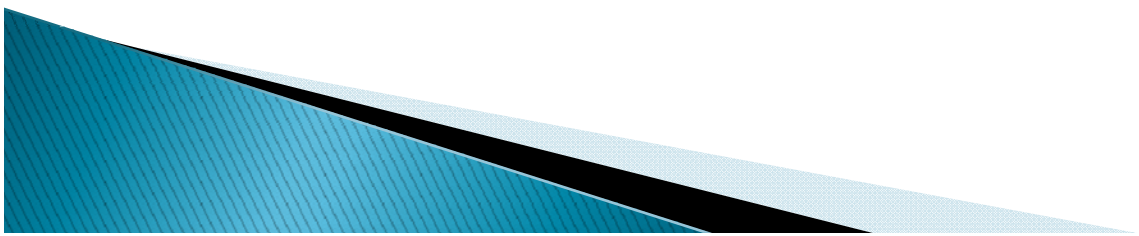


Examples of Context (contd)



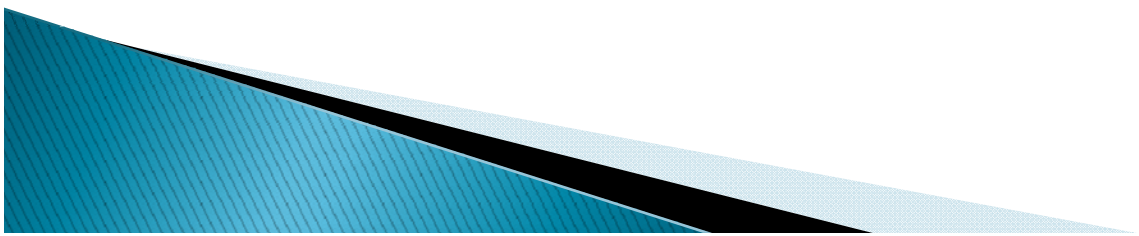
What is it good for?

- Lets computers understand an object or scene in the same way it can help humans understand a word in a sentence.
- Determining what objects are, even if the object can exist separately of the context.
- Generally, context determines priors on object interpretations.



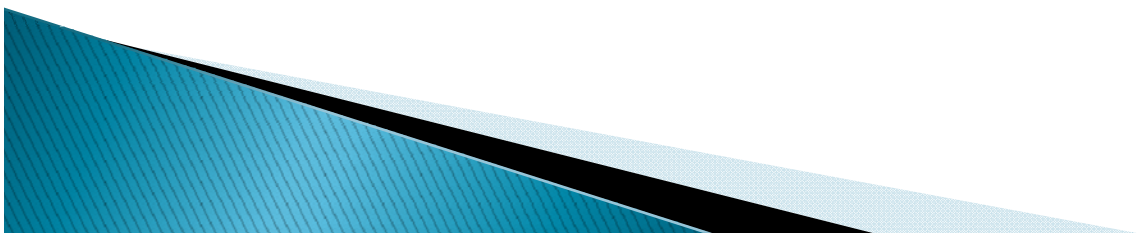
Outline

- What is context?
- Learning Spatial Context: Using Stuff to Find Things
  - Geremy Heitz and Daphne Koller
- Putting Objects in Perspective
  - Derek Hoiem, Alexei A. Efros and Martial Hebert



Types of Context

- Scene-Thing
  - scale
  - “gist”
  - Determines priors for objects
- Thing-Thing
  - Object cooccurrence
- Stuff-Stuff
  - E.g. beach, water
- Stuff-Thing
  - Texture regions relative to objects

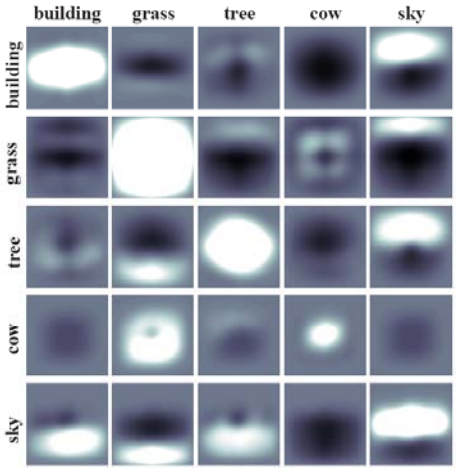


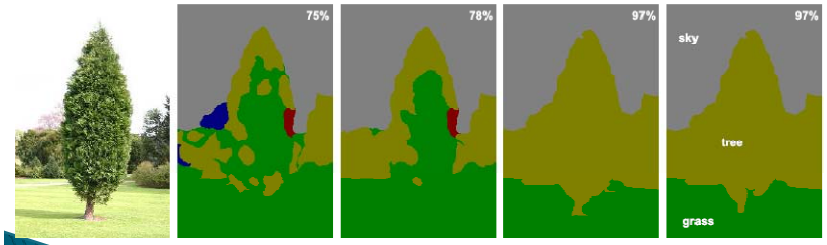
- Scene–Thing:


car “likely”

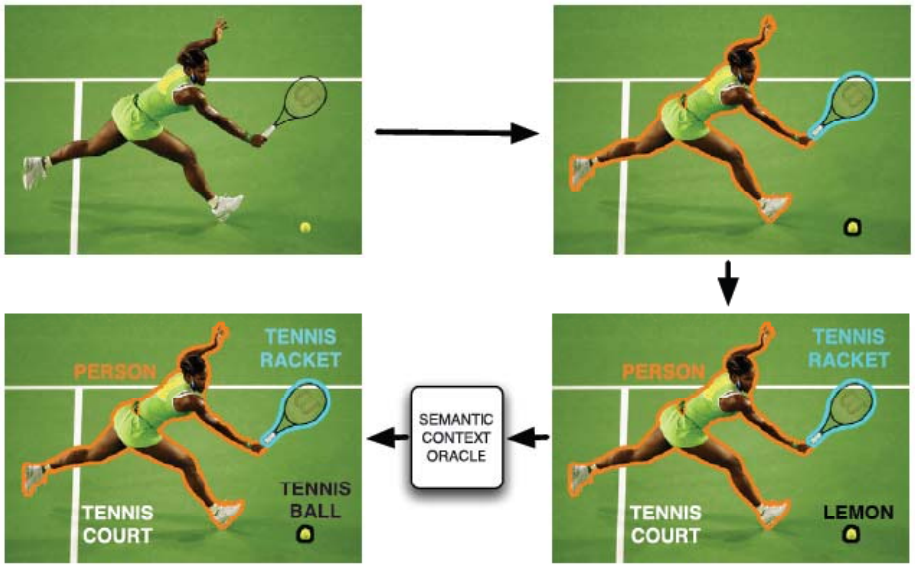
keyboard “unlikely”

- Stuff–Stuff: [Gould et al., IJCV 2008]





- Thing-Thing: [Rabinovich et al., ICCV 2007]

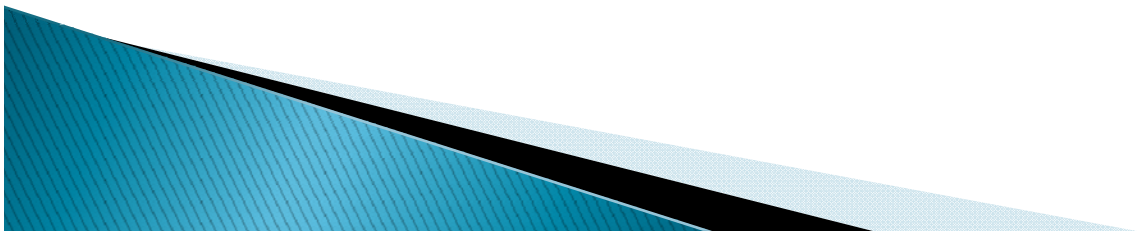


[Heitz 2008]

Thing and Stuff Context

- Things in the context of stuff, and vice versa.

- Components:
  - Things (T)
  - Feature descriptors for windows (W)
  - Feature descriptor for regions (F)
  - Stuff classes for regions (S)
  - Relationship indicator variables (R)



Things

- Discrete Objects
- Have specific size and shape.
- Generally mobile

- Examples:
  - Car
  - Person
  - Bicycle

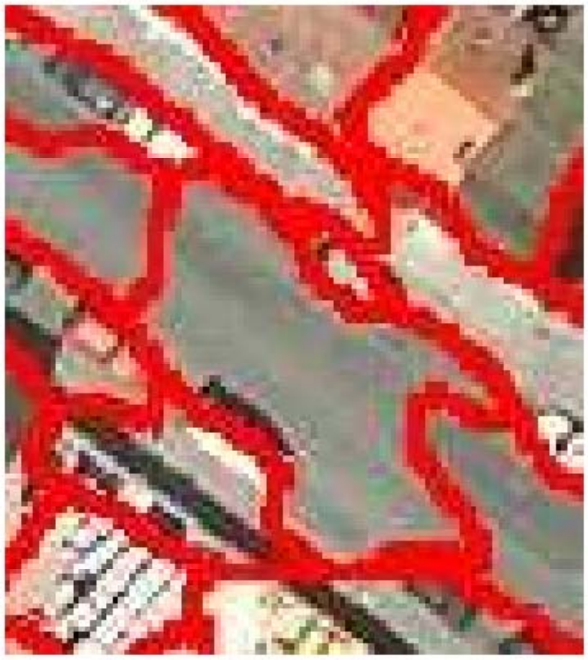
- In TAS:
  - Detected with local window detectors

[Heitz 2008]

Stuff

- Generally immobile
- Shapeless
- Examples:
  - Road
  - Buildings

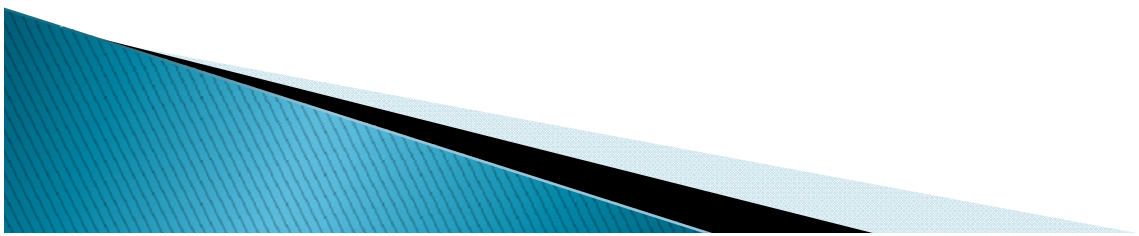
- In TAS:
  - Labeled regions defined by superpixels
  - Assumed to be independent of each other.
  - Labeled by the homogeneous or repetitive pattern of fine-scale properties



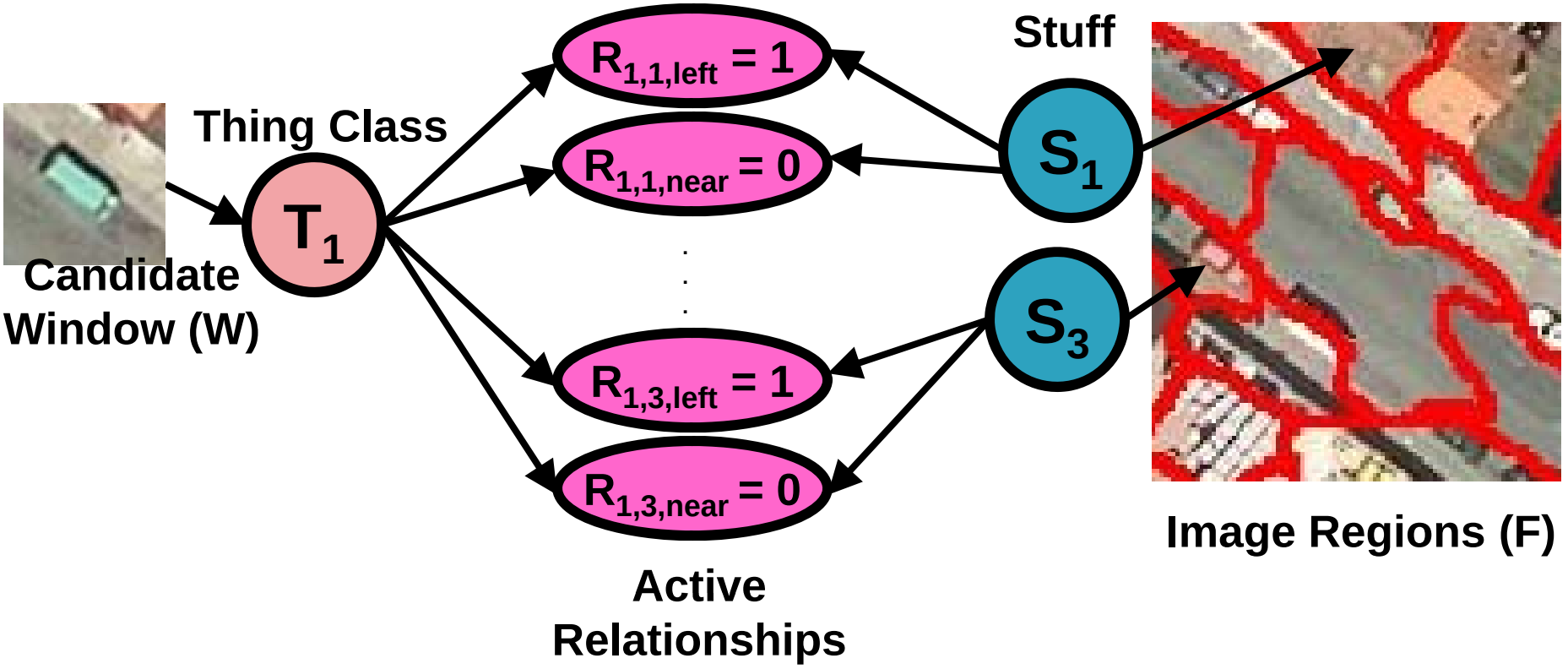
Satellite Regions [Heitz 2008]

Relationships

- Describes a possible relation between a Stuff and a Thing
- Large number of candidate relations are generated, best are picked algorithmically
- Examples:
  - Thing **Above** Stuff
  - Thing **Right of** Stuff
  - Thing **In** Stuff
- K possible relationships are mapped to all I * J Thing/Stuff combinations, for a total of I*J*K relationship indicator variables.



Thing/Stuff Relationship



[Heitz 2008]

Modular representation

- Keeps complexity down
  - Things only depend on Stuff, not other things.
  - Stuff only depends on things.

$$P(\boldsymbol{T}, \boldsymbol{S}, \boldsymbol{F}, \boldsymbol{R} \mid \boldsymbol{W}) = \prod_i P(T_i \mid W_i) \prod_j P(S_j) P(\boldsymbol{F}_j \mid S_j) \prod_{ijk} P(R_{ijk} \mid T_i, S_j)$$

  - All probabilities are drawn from simple table conditional probability distributions (CPDs).
- In reality, thing/things and stuff/stuff are not independent, but the probability distribution becomes modular.

Training Initialization

- Stuff
  - Clustered based on detected features from Superpixel regions
- Things
  - Local object detector is trained from annotated training set
- Relationships
  - Potential relationships manually defined
  - All are initially inactive

Training Algorithm

Algorithm LearnTAS
Input: Candidate relationships $\mathcal{C}$, Dataset $\mathcal{D} = \{(\boldsymbol{W}[m], \boldsymbol{T}[m], \boldsymbol{F}[m], \boldsymbol{R}[m])\}$
$\mathcal{R} \leftarrow \emptyset$ (all relationships "inactive")
Repeat until convergence
Repeat until convergence (EM over Parameters)
E-step: $Q[m] \leftarrow P(\boldsymbol{S} \mid \boldsymbol{T}, \boldsymbol{F}, \boldsymbol{R}; \theta_{\mathcal{R}}) \quad \forall m$
M-step: $\theta_{\mathcal{R}} \leftarrow \text{argmax} E_Q \left[\sum_m \ell(\boldsymbol{S}, \boldsymbol{T}, \boldsymbol{F}, \boldsymbol{R} \mid \boldsymbol{W}; \theta_{\mathcal{R}})\right]$
Repeat until convergence (Greedy Structure Search)
Forall k, $score_k = \sum_m \ell(\boldsymbol{T} \mid \boldsymbol{F}, \boldsymbol{R}; \theta_{\mathcal{R} \oplus k})$ (score with k "activated")
$\mathcal{R} \leftarrow \mathcal{R} \oplus k^*$ where $k^* = \text{argmax } score_k$
Return Set $\mathcal{R}$ of "active" relationships, TAS parameters $\theta_{\mathcal{R}}$

Fig. 4. Learning a TAS model. Here ℓ represents the log-likelihood of the data, and $\oplus$ represents the set exclusive-or operation.

[Heitz 2008]

Training Algorithm

Algorithm LearnTAS

Input: Candidate relationships $\mathcal{C}$, Dataset $\mathcal{D} = \{(\boldsymbol{W}[m], \boldsymbol{T}[m], \boldsymbol{F}[m], \boldsymbol{R}[m])\}$

$\mathcal{R} \leftarrow \emptyset$ (all relationships "inactive")

Repeat until convergence

Repeat until convergence (EM over Parameters)

E-step: $Q[m] \leftarrow P(\boldsymbol{S} \mid \boldsymbol{T}, \boldsymbol{F}, \boldsymbol{R}; \theta_{\mathcal{R}}) \quad \forall m$

M-step: $\theta_{\mathcal{R}} \leftarrow \text{argmax} E_Q \left[\sum_m \ell(\boldsymbol{S}, \boldsymbol{T}, \boldsymbol{F}, \boldsymbol{R} \mid \boldsymbol{W}; \theta_{\mathcal{R}})\right]$

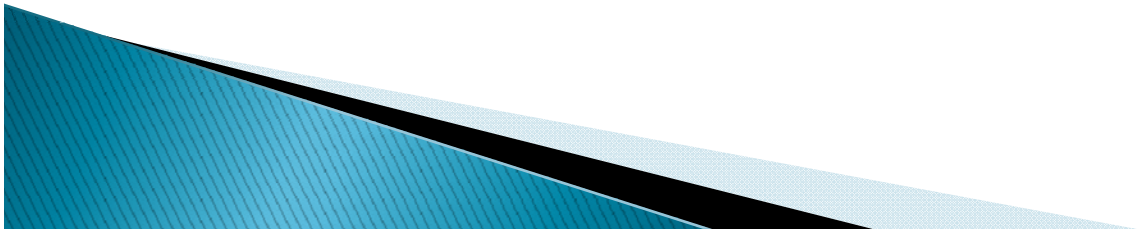
Repeat until convergence (Greedy Structure Search)

Forall k, $score_k = \sum_m \ell(\boldsymbol{T} \mid \boldsymbol{F}, \boldsymbol{R}; \theta_{\mathcal{R} \oplus k})$ (score with k "activated")

$\mathcal{R} \leftarrow \mathcal{R} \oplus k^*$ where $k^* = \text{argmax } score_k$

Return Set $\mathcal{R}$ of "active" relationships, TAS parameters $\theta_{\mathcal{R}}$

Candidate relationships arbitrarily generated.
Priors for Stuff given Features generated from clustering.



Training Algorithm

Algorithm LearnTAS

Input: Candidate relationships $\mathcal{C}$, Dataset $\mathcal{D} = \{(\boldsymbol{W}[m], \boldsymbol{T}[m], \boldsymbol{F}[m], \boldsymbol{R}[m])\}$

$\mathcal{R} \leftarrow \emptyset$ (all relationships "inactive")

Repeat until convergence

Repeat until convergence (EM over Parameters)

E-step: $Q[m] \leftarrow P(\boldsymbol{S} \mid \boldsymbol{T}, \boldsymbol{F}, \boldsymbol{R}; \theta_{\mathcal{R}}) \quad \forall m$

M-step: $\theta_{\mathcal{R}} \leftarrow \text{argmax} E_Q \left[\sum_m \ell(\boldsymbol{S}, \boldsymbol{T}, \boldsymbol{F}, \boldsymbol{R} \mid \boldsymbol{W}; \theta_{\mathcal{R}})\right]$

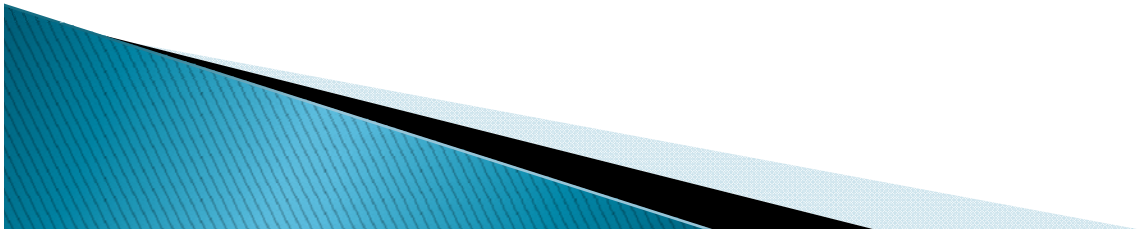
Repeat until convergence (Greedy Structure Search)

Forall k, $score_k = \sum_m \ell(\boldsymbol{T} \mid \boldsymbol{F}, \boldsymbol{R}; \theta_{\mathcal{R} \oplus k})$ (score with k "activated")

$\mathcal{R} \leftarrow \mathcal{R} \oplus k^*$ where $k^* = \text{argmax } score_k$

Return Set $\mathcal{R}$ of "active" relationships, TAS parameters $\theta_{\mathcal{R}}$

All relationships begin inactive.
There is a likelihood on how many relationships can become active.



Training Algorithm

Algorithm LearnTAS

Input: Candidate relationships $\mathcal{C}$, Dataset $\mathcal{D} = \{(\boldsymbol{W}[m], \boldsymbol{T}[m], \boldsymbol{F}[m], \boldsymbol{R}[m])\}$

$\mathcal{R} \leftarrow \emptyset$ (all relationships "inactive")

Repeat until convergence

Repeat until convergence (EM over Parameters)

E-step: $Q[m] \leftarrow P(\boldsymbol{S} \mid \boldsymbol{T}, \boldsymbol{F}, \boldsymbol{R}; \theta_{\mathcal{R}}) \quad \forall m$

M-step: $\theta_{\mathcal{R}} \leftarrow \text{argmax} E_Q \left[\sum_m \ell(\boldsymbol{S}, \boldsymbol{T}, \boldsymbol{F}, \boldsymbol{R} \mid \boldsymbol{W}; \theta_{\mathcal{R}})\right]$

Repeat until convergence (Greedy Structure Search)

Forall k, $score_k = \sum_m \ell(\boldsymbol{T} \mid \boldsymbol{F}, \boldsymbol{R}; \theta_{\mathcal{R} \oplus k})$ (score with k "activated")

$\mathcal{R} \leftarrow \mathcal{R} \oplus k^*$ where $k^* = \text{argmax } score_k$

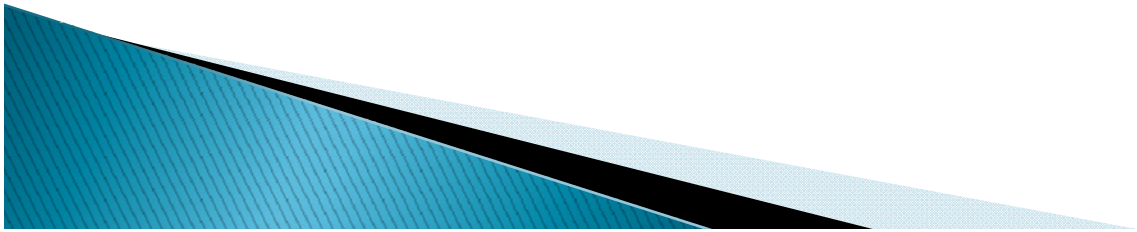
Return Set $\mathcal{R}$ of "active" relationships, TAS parameters $\theta_{\mathcal{R}}$

Use model and Ground Truth to estimate most likely Stuff classifications.
Q is the probability of the Stuff classes.
m is set of training images.

Training Algorithm

Algorithm LearnTAS
Input: Candidate relationships $\mathcal{C}$, Dataset $\mathcal{D} = \{(\boldsymbol{W}[m], \boldsymbol{T}[m], \boldsymbol{F}[m], \boldsymbol{R}[m])\}$
$\mathcal{R} \leftarrow \emptyset$ (all relationships "inactive")
Repeat until convergence
Repeat until convergence (EM over Parameters)
E-step: $Q[m] \leftarrow P(\boldsymbol{S} \mid \boldsymbol{T}, \boldsymbol{F}, \boldsymbol{R}; \theta_{\mathcal{R}}) \quad \forall m$
M-step: $\theta_{\mathcal{R}} \leftarrow \text{argmax} E_Q \left[\sum_m \ell(\boldsymbol{S}, \boldsymbol{T}, \boldsymbol{F}, \boldsymbol{R} \mid \boldsymbol{W}; \theta_{\mathcal{R}})\right]$
Repeat until convergence (Greedy Structure Search)
Forall k, $score_k = \sum_m \ell(\boldsymbol{T} \mid \boldsymbol{F}, \boldsymbol{R}; \theta_{\mathcal{R} \oplus k})$ (score with k "activated")
$\mathcal{R} \leftarrow \mathcal{R} \oplus k^*$ where $k^* = \text{argmax } score_k$
Return Set $\mathcal{R}$ of "active" relationships, TAS parameters $\theta_{\mathcal{R}}$

Pick model (collection of CPDs) that makes observed data (Things) and estimated data (Stuff) most probable.



Training Algorithm

Algorithm LearnTAS

Input: Candidate relationships $\mathcal{C}$, Dataset $\mathcal{D} = \{(\boldsymbol{W}[m], \boldsymbol{T}[m], \boldsymbol{F}[m], \boldsymbol{R}[m])\}$

$\mathcal{R} \leftarrow \emptyset$ (all relationships "inactive")

Repeat until convergence

Repeat until convergence (EM over Parameters)

E-step: $Q[m] \leftarrow P(\boldsymbol{S} \mid \boldsymbol{T}, \boldsymbol{F}, \boldsymbol{R}; \theta_{\mathcal{R}}) \quad \forall m$

M-step: $\theta_{\mathcal{R}} \leftarrow \text{argmax} E_Q \left[\sum_m \ell(\boldsymbol{S}, \boldsymbol{T}, \boldsymbol{F}, \boldsymbol{R} \mid \boldsymbol{W}; \theta_{\mathcal{R}})\right]$

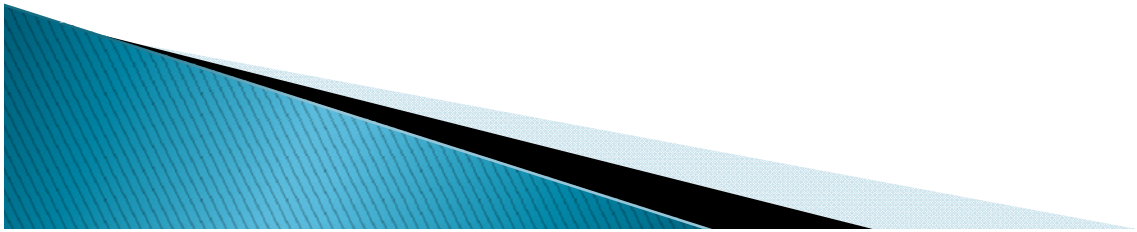
Repeat until convergence (Greedy Structure Search)

Forall k, $score_k = \sum_m \ell(\boldsymbol{T} \mid \boldsymbol{F}, \boldsymbol{R}; \theta_{\mathcal{R} \oplus k})$ (score with k "activated")

$\mathcal{R} \leftarrow \mathcal{R} \oplus k^*$ where $k^* = \text{argmax } score_k$

Return Set $\mathcal{R}$ of "active" relationships, TAS parameters $\theta_{\mathcal{R}}$

Greedy structural search over all possible relationships.
Add one or subtract one, and figure out which change helped the most.



Training Algorithm

Algorithm LearnTAS

Input: Candidate relationships $\mathcal{C}$, Dataset $\mathcal{D} = \{(\boldsymbol{W}[m], \boldsymbol{T}[m], \boldsymbol{F}[m], \boldsymbol{R}[m])\}$

$\mathcal{R} \leftarrow \emptyset$ (all relationships "inactive")

Repeat until convergence

Repeat until convergence (EM over Parameters)

E-step: $Q[m] \leftarrow P(\boldsymbol{S} \mid \boldsymbol{T}, \boldsymbol{F}, \boldsymbol{R}; \theta_{\mathcal{R}}) \quad \forall m$

M-step: $\theta_{\mathcal{R}} \leftarrow \text{argmax} E_Q \left[\sum_m \ell(\boldsymbol{S}, \boldsymbol{T}, \boldsymbol{F}, \boldsymbol{R} \mid \boldsymbol{W}; \theta_{\mathcal{R}})\right]$

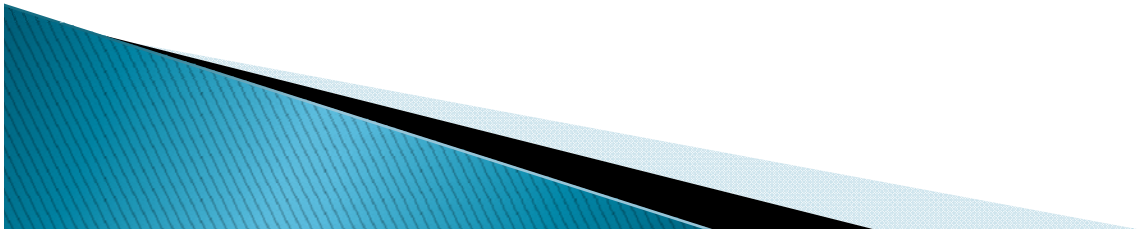
Repeat until convergence (Greedy Structure Search)

Forall k, $score_k = \sum_m \ell(\boldsymbol{T} \mid \boldsymbol{F}, \boldsymbol{R}; \theta_{\mathcal{R} \oplus k})$ (score with k "activated")

$\mathcal{R} \leftarrow \mathcal{R} \oplus k^*$ where $k^* = \text{argmax } score_k$

Return Set $\mathcal{R}$ of "active" relationships, TAS parameters $\theta_{\mathcal{R}}$

Return the complete model!



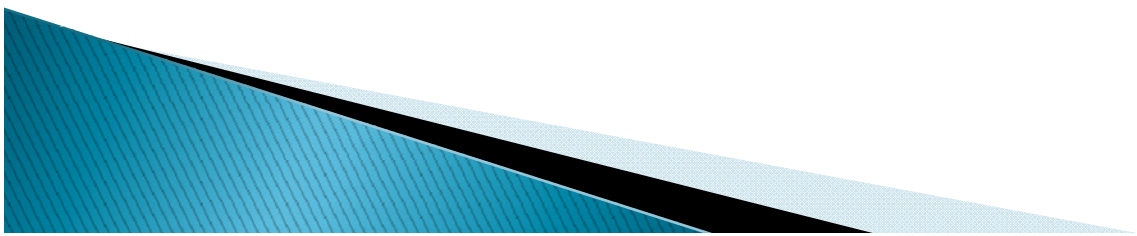
Classifying things:

Now that we have our model, we want to use it to classify things:

$$P(\boldsymbol{T} \mid \boldsymbol{F}, \boldsymbol{R}, \boldsymbol{W}) = \sum_{\boldsymbol{S}} P(\boldsymbol{T}, \boldsymbol{S} \mid \boldsymbol{F}, \boldsymbol{R}, \boldsymbol{W})$$

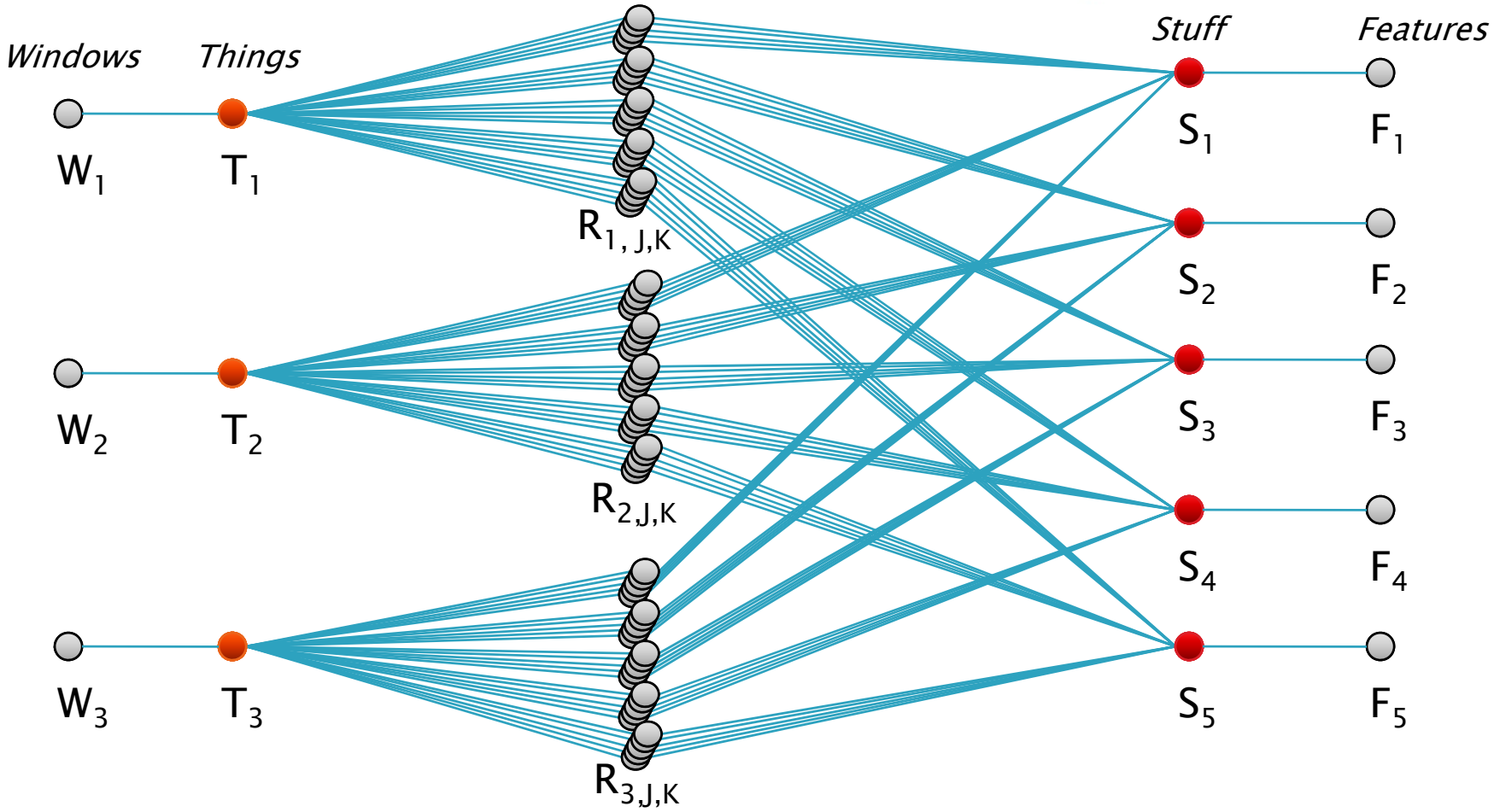
But, this is different from training because now Thing classes are unobserved as well as Stuff classes.

So finding this involves computing all possible combinations of Things and Stuff in the entire image!



Things + Stuff

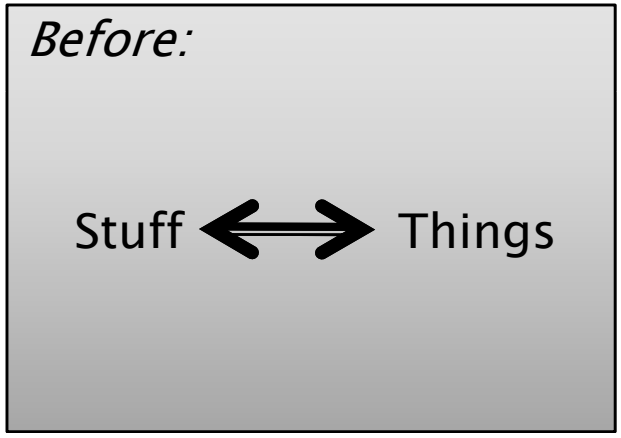
$$P(\boldsymbol{T} \mid \boldsymbol{F}, \boldsymbol{R}, \boldsymbol{W}) = \sum_{\boldsymbol{S}} P(\boldsymbol{T}, \boldsymbol{S} \mid \boldsymbol{F}, \boldsymbol{R}, \boldsymbol{W})$$

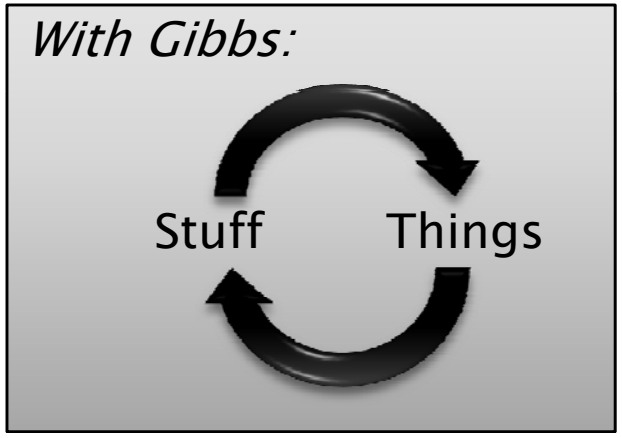


Conditional dependence of things on things and stuff on stuff makes it computationally intractable!

What to do?

- Try Gibbs sampling:
  - Variant of Markov Chain Monte Carlo (MCMC)
  - Assume all but one parameters, estimate that parameter from others given data.
  - Repeat until convergence





Gibb's Sampling

$$P(\boldsymbol{T} \mid \boldsymbol{F}, \boldsymbol{R}, \boldsymbol{W}) = \sum_{\boldsymbol{S}} P(\boldsymbol{T}, \boldsymbol{S} \mid \boldsymbol{F}, \boldsymbol{R}, \boldsymbol{W})$$

becomes

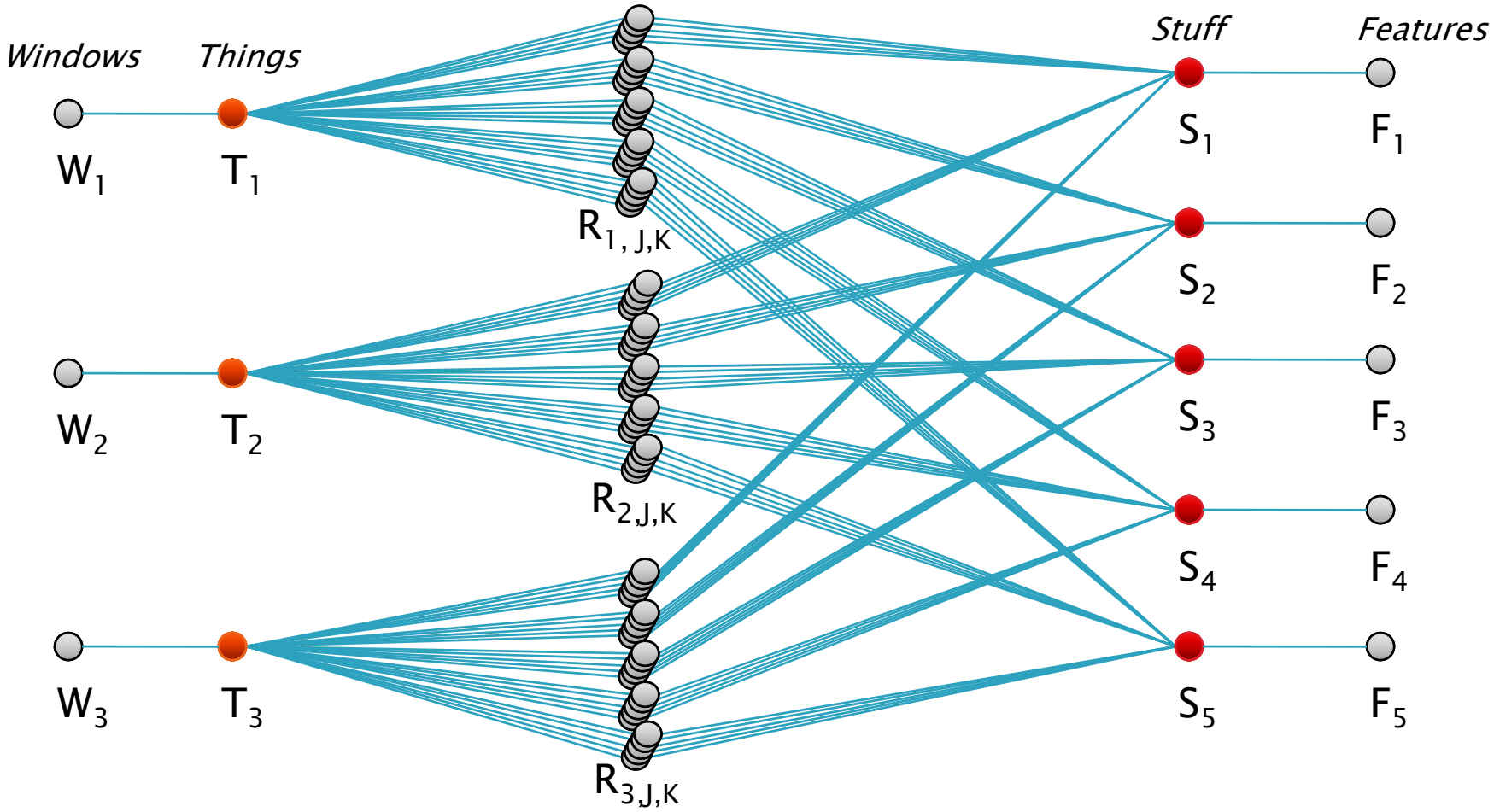
$$P(S_j \mid \boldsymbol{T}, \boldsymbol{F}, \boldsymbol{R}, \boldsymbol{W}) \propto P(S_j) P(F_j \mid S_j) \prod_{ik} P(R_{ijk} \mid T_i, S_j)$$

$$P(T_i \mid \boldsymbol{S}, \boldsymbol{F}, \boldsymbol{R}, \boldsymbol{W}) \propto P(T_i \mid W_i) \prod_{jk} P(R_{ijk} \mid T_i, S_j).$$

iterated until convergence

Things + Stuff

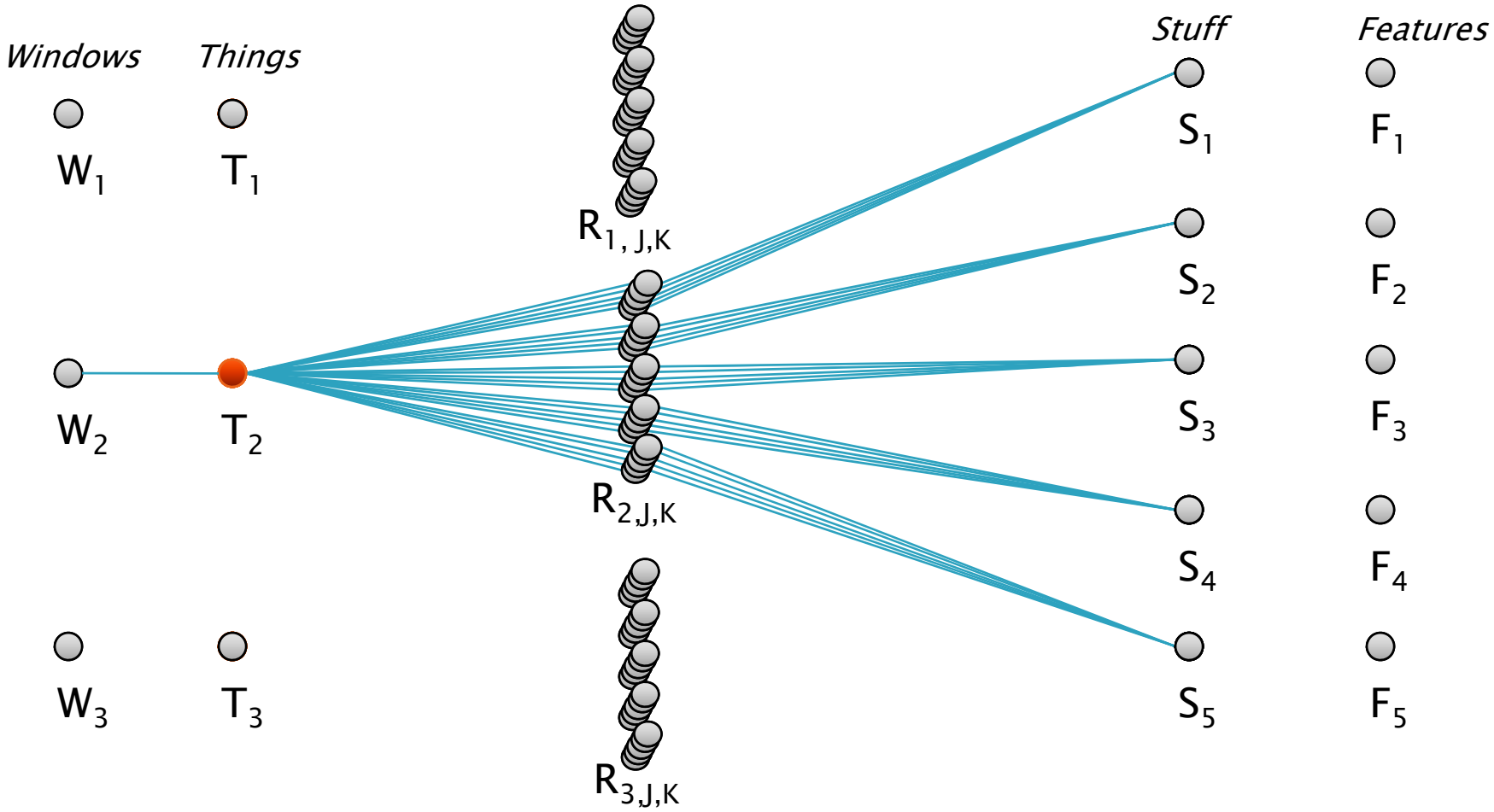
$$P(\boldsymbol{T} \mid \boldsymbol{F}, \boldsymbol{R}, \boldsymbol{W}) = \sum_{\boldsymbol{S}} P(\boldsymbol{T}, \boldsymbol{S} \mid \boldsymbol{F}, \boldsymbol{R}, \boldsymbol{W})$$



Conditional dependence of things on things and stuff on stuff makes it computationally intractable!

Just Things

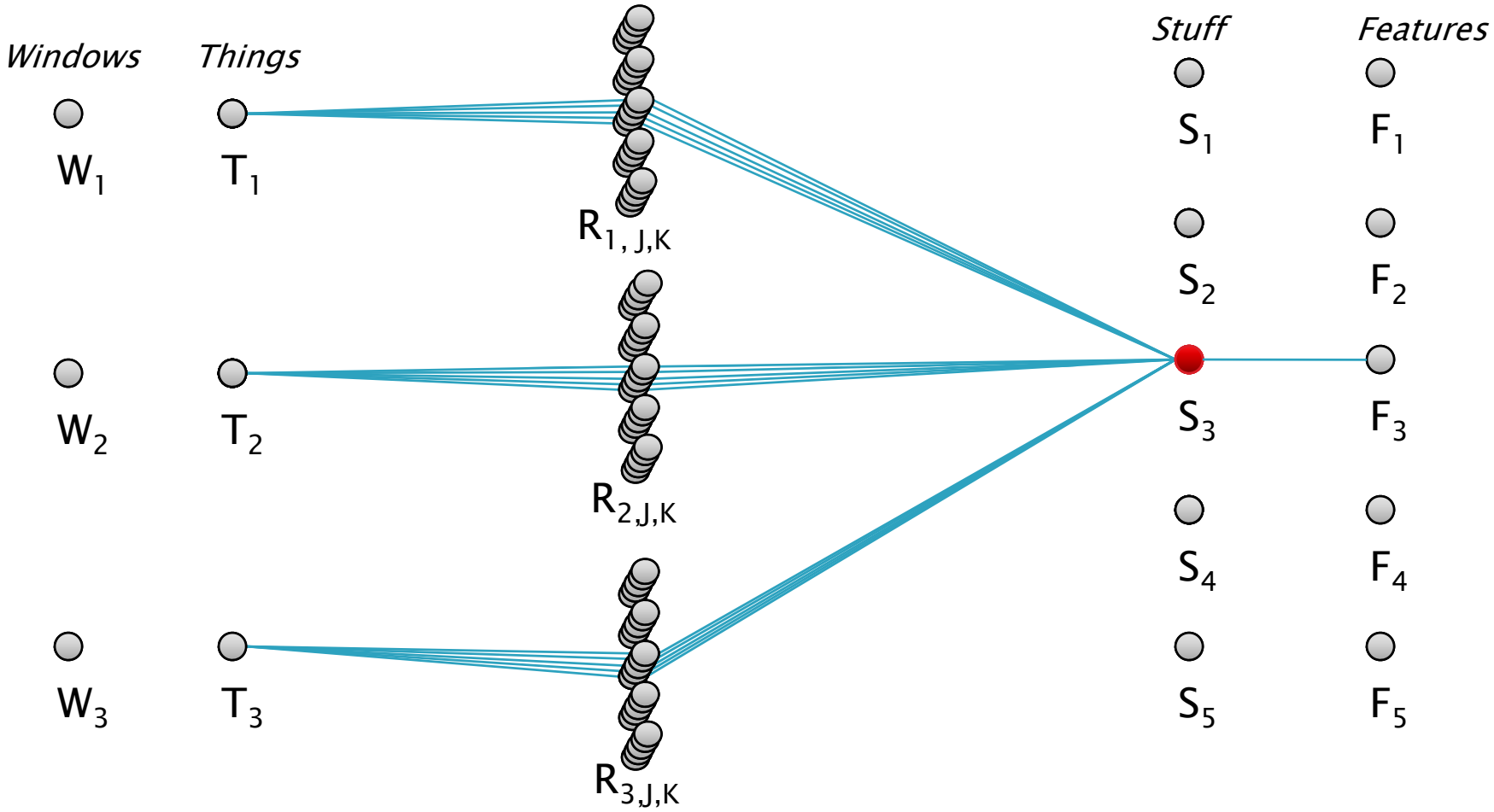
$$P(T_i \mid \boldsymbol{S}, \boldsymbol{F}, \boldsymbol{R}, \boldsymbol{W}) \propto P(T_i \mid W_i) \prod_{jk} P(R_{ijk} \mid T_i, S_j)$$



Computing Thing probabilities becomes linear on the number of Thing candidates!

Just Stuff

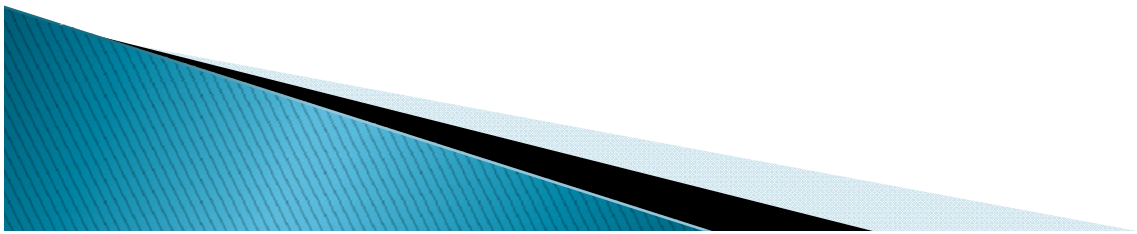
$$P(S_j \mid \boldsymbol{T}, \boldsymbol{F}, \boldsymbol{R}, \boldsymbol{W}) \propto P(S_j)P(F_j \mid S_j) \prod_{ik} P(R_{ijk} \mid T_i, S_j)$$



Computing stuff probabilities becomes linear on the number of regions!

Experiments:

- VOC2005 dataset:
  - 2232 images
  - manually annotated bounding boxes for:
    - Cars
    - People
    - Motorbikes
    - Bicycles
- VOC2006 dataset:
  - 2686 images
  - manually annotated bounding boxes for:
    - Cows
    - Sheep
- Bonus feature: Satellite Imagery
- Source code available
  - Website: http://ai.stanford.edu/~gaheitz/Research/TAS/
  - Includes all data from experiments



Example Detections

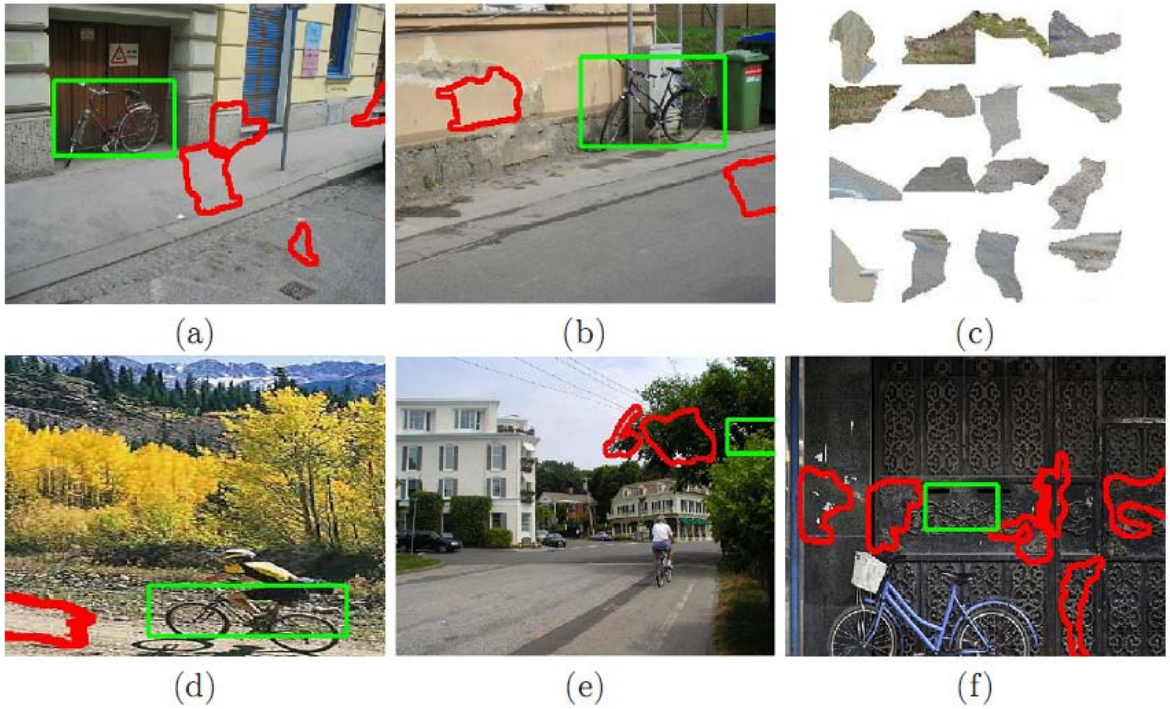
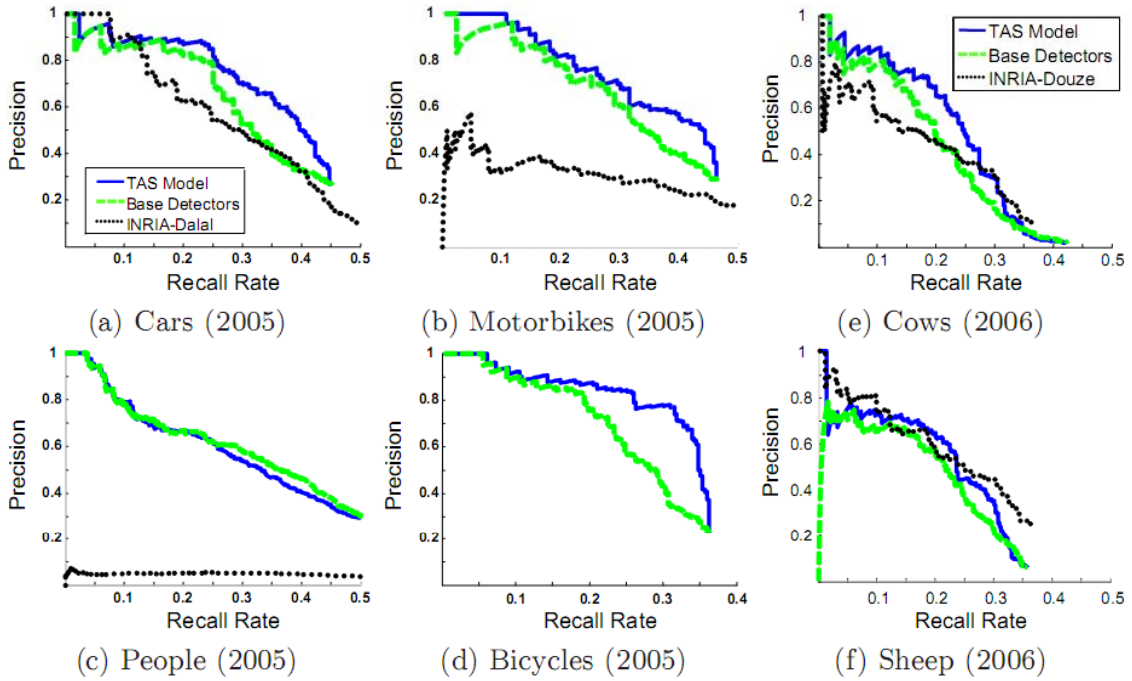


Fig. 5. (a,b) Example training detections from the bicycle class, with detection windows outlined by the green rectangles. The image regions with active relationships to the detection window are outlined in red. (c) 16 of the most representative regions for cluster #3. This cluster corresponds to "roads" or "bushes" as things that are gray/green and occur near cars. (d) A case where context helped find a true detection. (e,f) Two examples where incorrect detections are filtered out by context.

[Heitz 2008]

Pascal Results



(a) Cars (2005) (b) Motorbikes (2005) (e) Cows (2006)

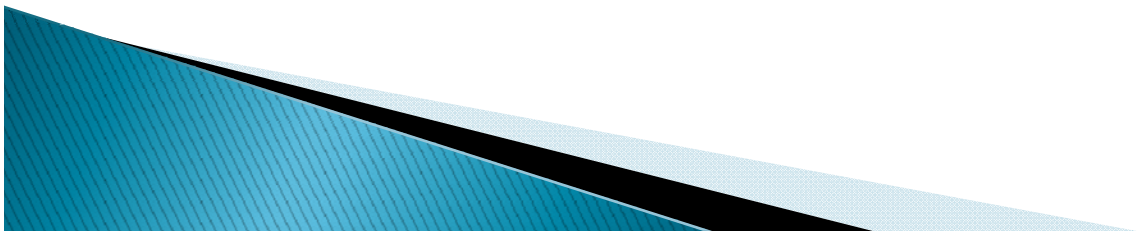
(c) People (2005) (d) Bicycles (2005) (f) Sheep (2006)

Object Class	Base AP	TAS AP (Fixed R)	TAS AP (Learned R)	Improvement (TAS - Base)
Cars	0.325	0.360	**0.363**	0.038
Motorbikes	0.341	**0.390**	0.373	0.032
People	**0.346**	0.346	0.337	-0.009
Bicycles	0.281	0.310	**0.325**	0.044
Cows	0.224	0.241	**0.258**	0.034
Sheep	0.206	0.233	**0.248**	0.042

[Heitz 2008]

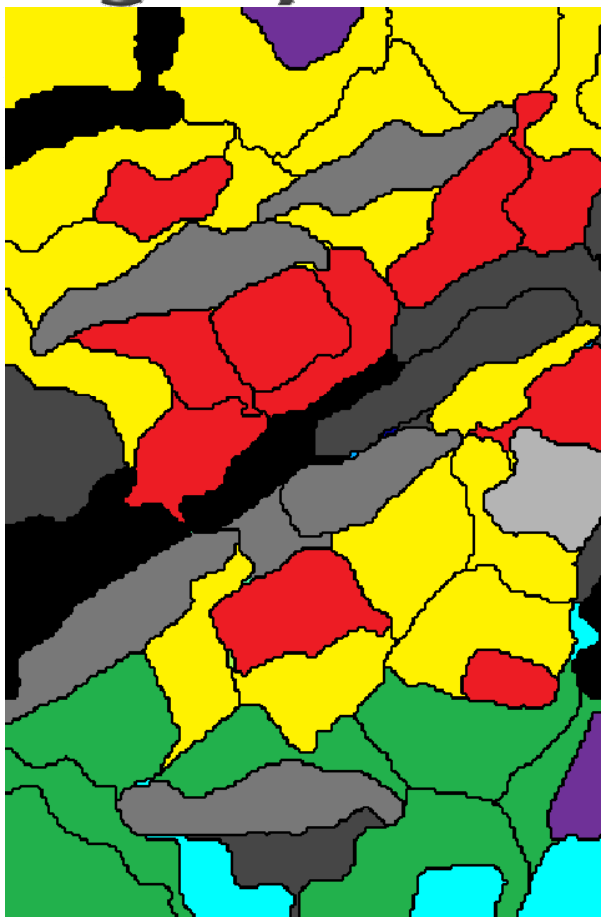
Satellite Experiment

- Training/Test Data:
  - 30 raw images pulled from Google Earth of size 792×636
  - Contain 1319 Hand-tagged cars
- Tested with 5-fold cross-validation
- Note that orthographic projection of plane aligned objects means objects are:
  - scale invariant
  - viewpoint invariant
  - but not rotationally invariant



Satellite Imagery

Prior:
Detector Only

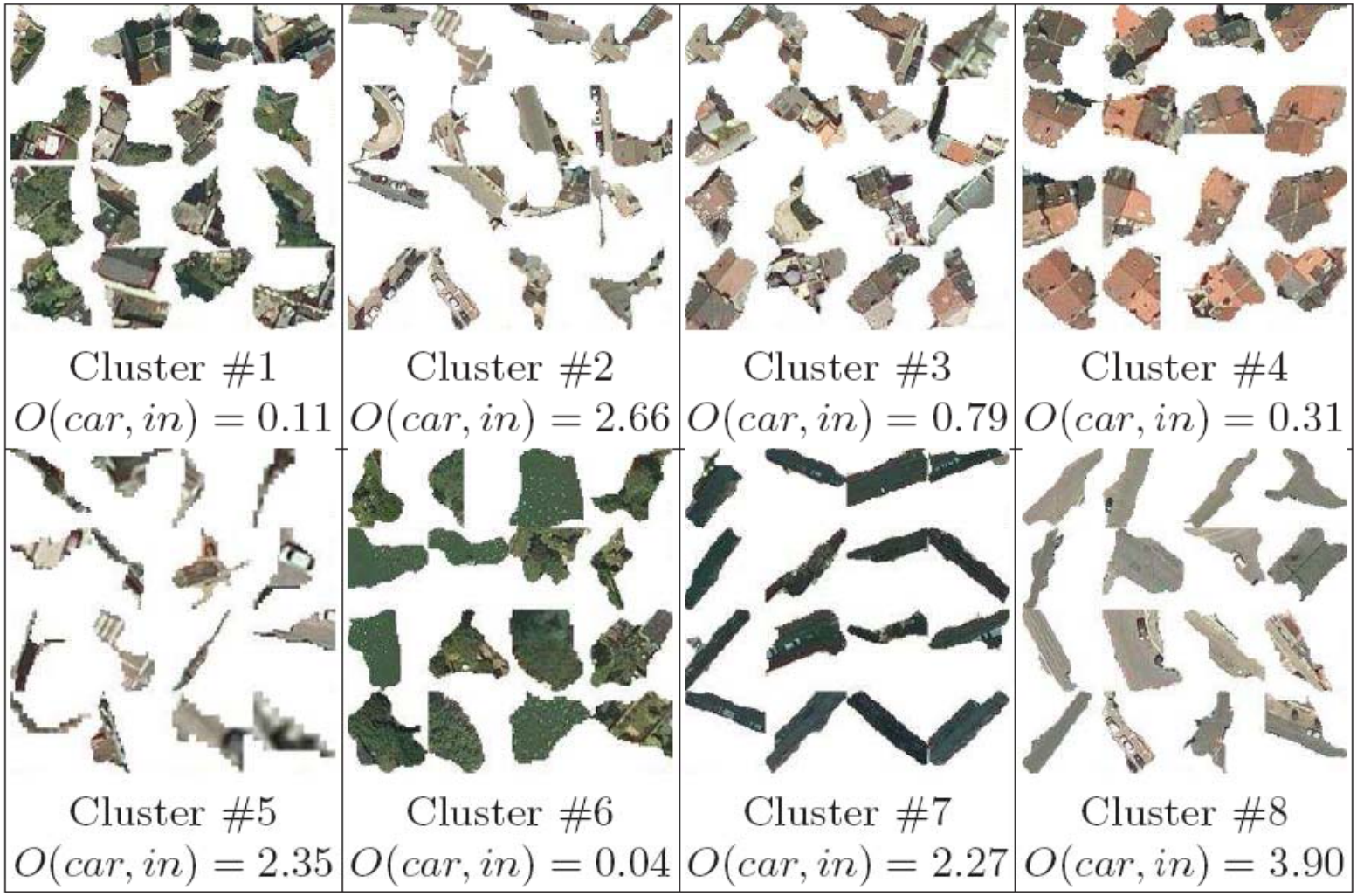


Posterior:
Region Labels

Posterior:
Detections

[Heitz 2008]

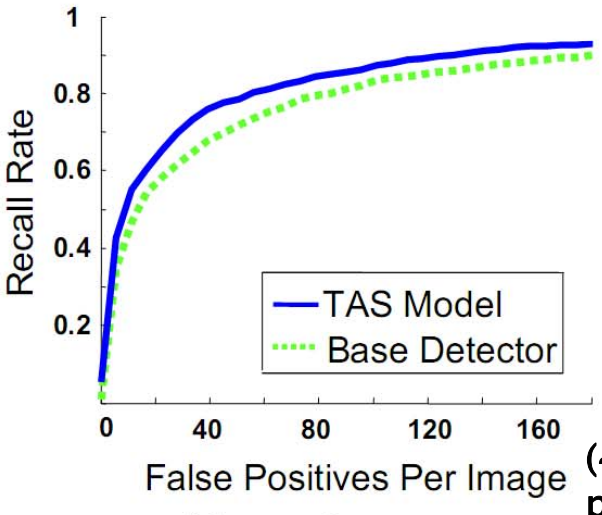
Learned Satellite Clusters



[Heitz 2008]

Satellite Results

(a) Base Detectors (b) TAS Detections



(c) PR Curve

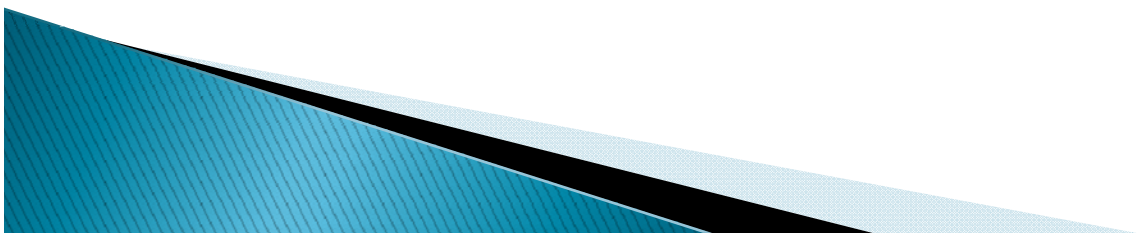
(~44 Cars per image)

Fig. 8. Example image, with detections found by the base detector (a), and by the TAS model (b) with a threshold of 0.15. The TAS model filters out many of the false positives far away from roads. (c) shows a plot of recall rate vs. false positives per image for the satellite data. The results here are averaged across 5 folds, and show a significant improvement from using TAS over the base detectors.

[Heitz 2008]

Outline

- What is context?
- Learning Spatial Context: Using Stuff to Find Things
  - Geremy Heitz and Daphne Koller
- Putting Objects in Perspective
  - Derek Hoiem, Alexei A. Efros and Martial Hebert



Perspective

Computer's Understanding of Scene:

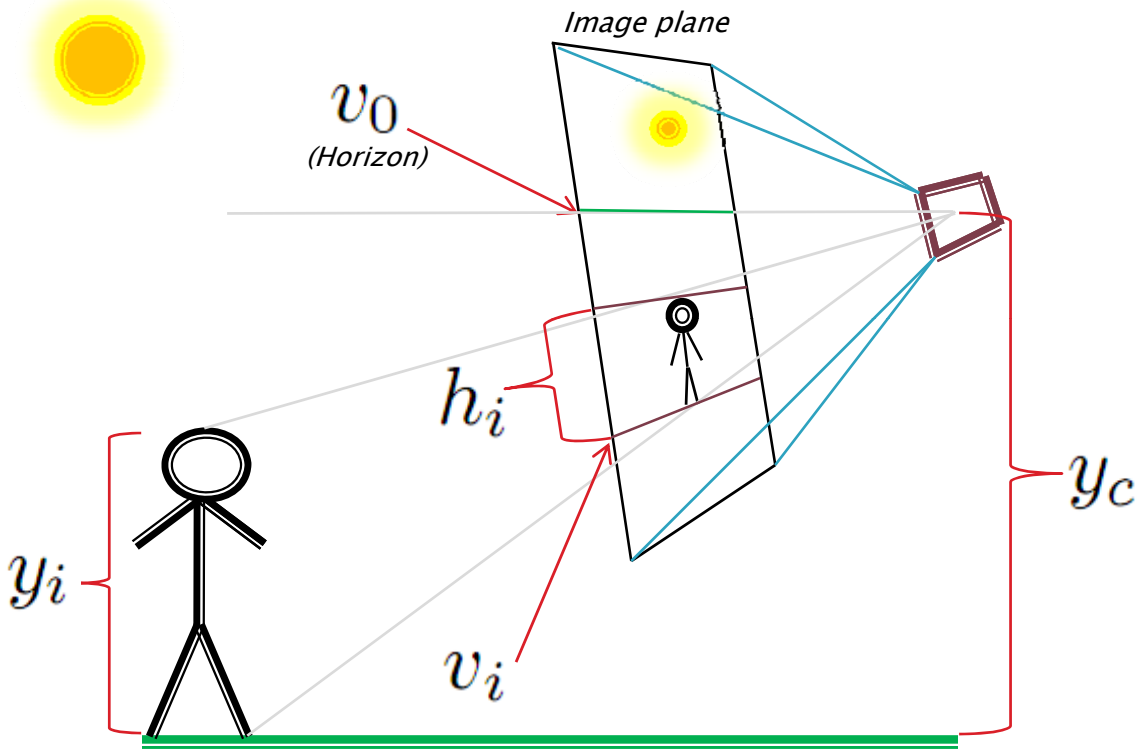
Without Perspective

With Perspective

(well, almost)

Paper Mario,
(c) Nintendo

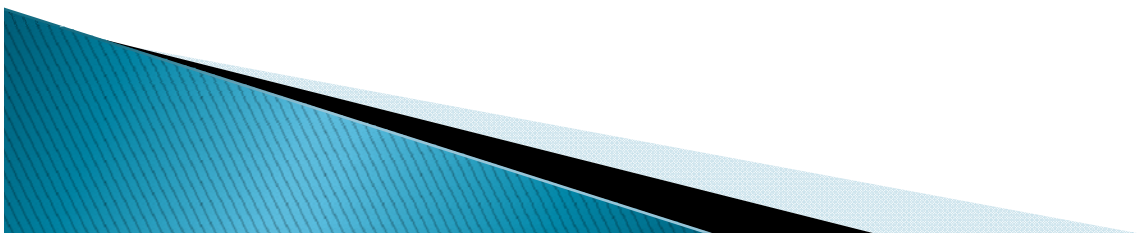
Determining object height



$$y_i = \frac{h_i y_c}{v_i - v_0}$$

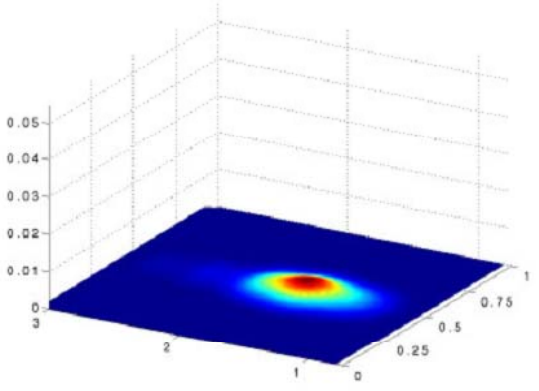
Putting Objects in Perspective

- Statistical Framework that allows simultaneous inference between:
  - Camera viewpoint
  - Object identities (Things)
  - Surface orientations (Geometry)

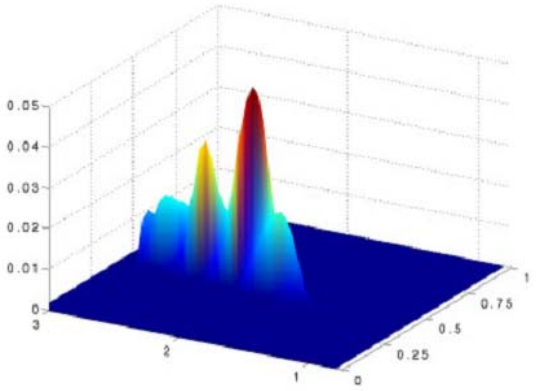


Camera

- Denoted by θ
- Only two parameters:
  - y_c : Height above ground plane
    - A priori height of 1.67m
  - v_0 : Vertical position of horizon line.
    - Initialized at 0.5



(e) Viewpoint: Prior



(f) Viewpoint: Full

[Hoiem 2006]

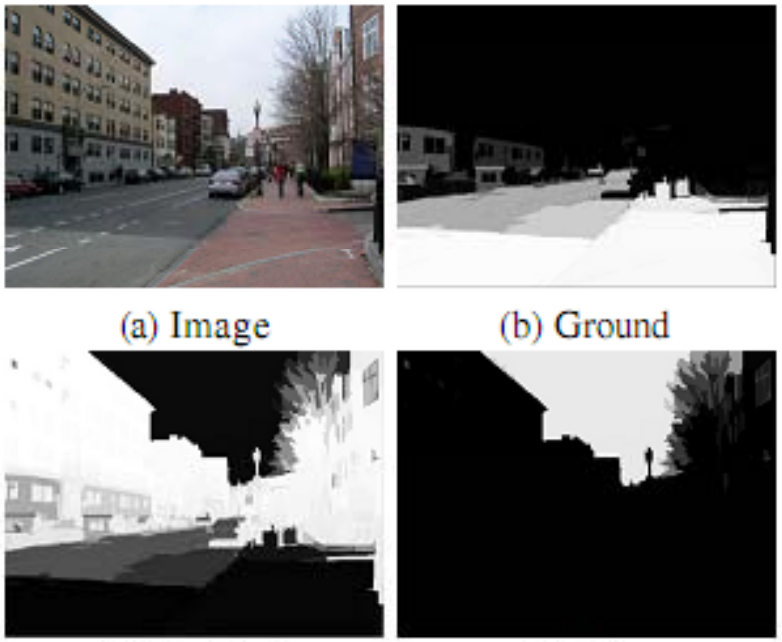
Things

- Detected by local object detector
- Based off of gist based object-detector of Murphy, Torralba, and Freeman

[Hoiem 2006]

Geometry

- Based on the previous work *Geometric Context from a Single Image*, also by Hoiem *et al*

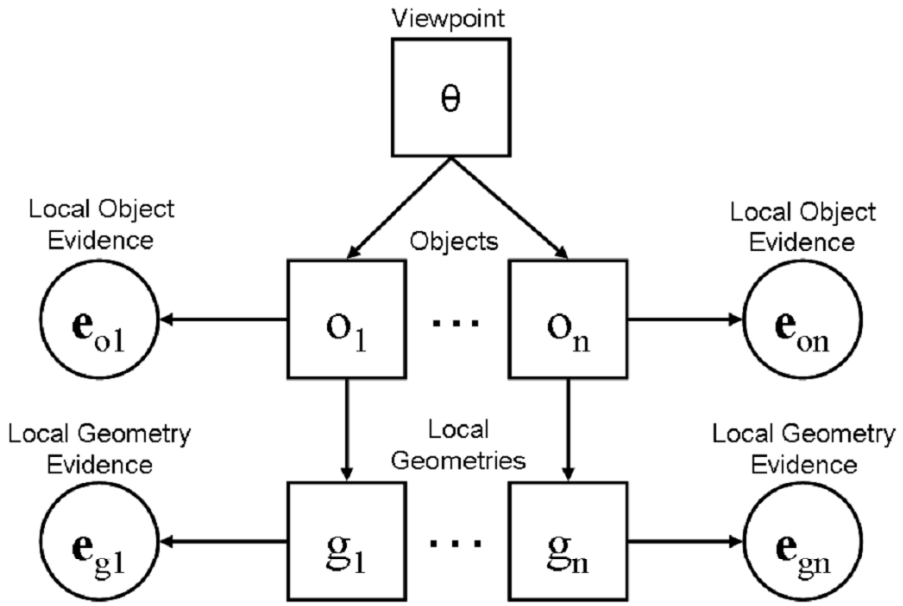


(a) Image (b) Ground

(c) Vertical (d) Sky

[Hoiem 2006]

Model Breakdown



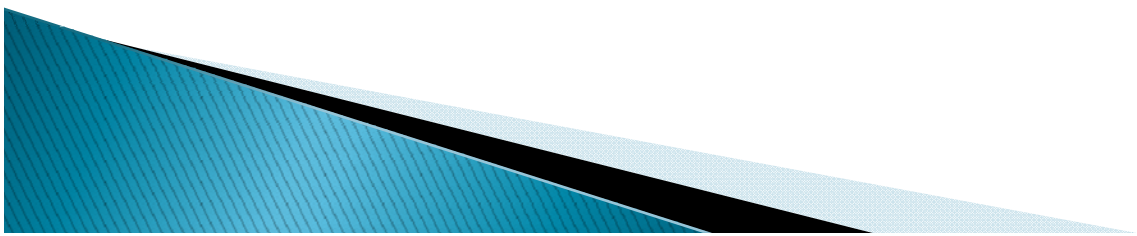
$$P(\theta, \mathbf{o}, \mathbf{g}|\mathbf{e}) \propto P(\theta) \prod_i P(o_i|\theta) \frac{P(o_i|\mathbf{e}_o)}{P(o_i)} P(g_i|o_i) \frac{P(g_i|\mathbf{e}_g)}{P(g_i)}$$

[Hoiem 2006]

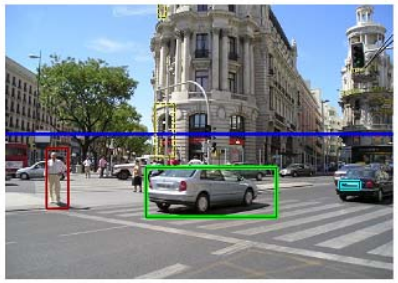
Experiment

- Test set consists of 422 random outdoor images from the LabelMe dataset.
- The images contain 923 cars and 720 pedestrians.

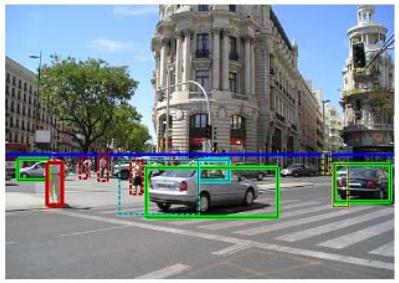
- 60 images have no cars or pedestrians
- 44 have only pedestrians
- 94 have only cars
- 224 have both cars and pedestrians



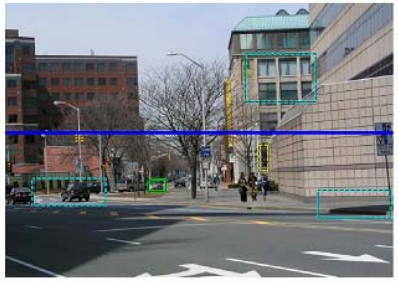
Sample Results



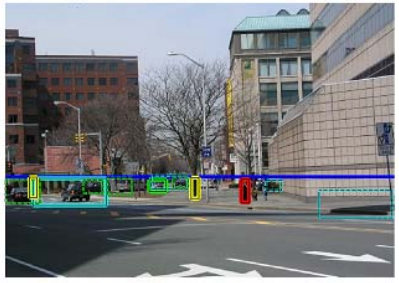
(a) Local Detection



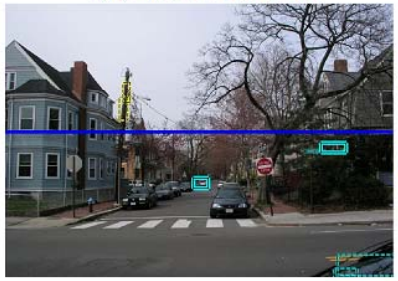
(a) Full Model Detection



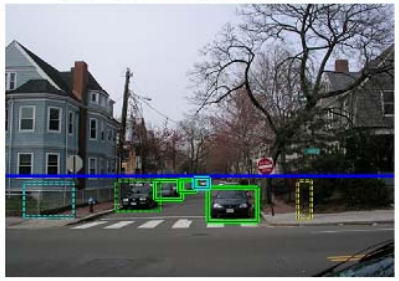
(c) Local Detection



(c) Full Model Detection



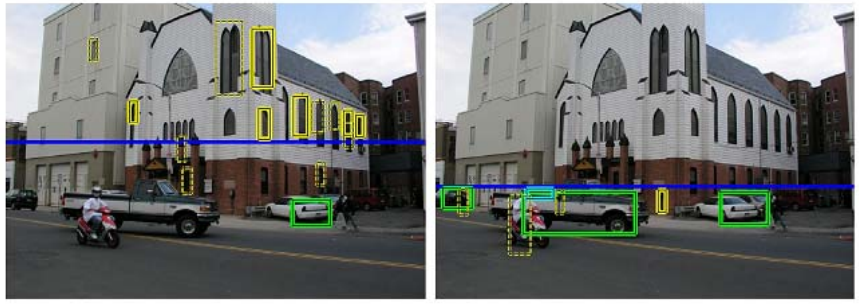
(e) Local Detection



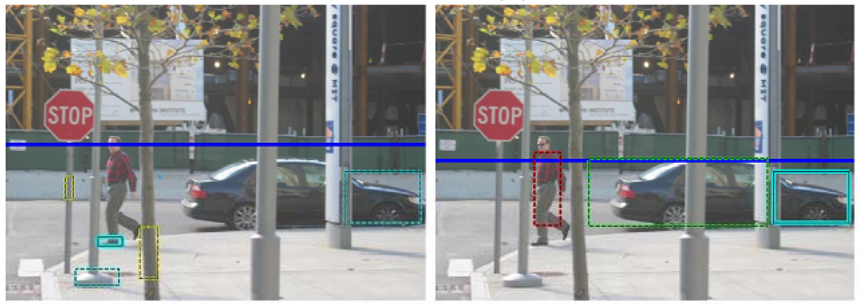
(e) Full Model Detection

[Hoiem 2006]

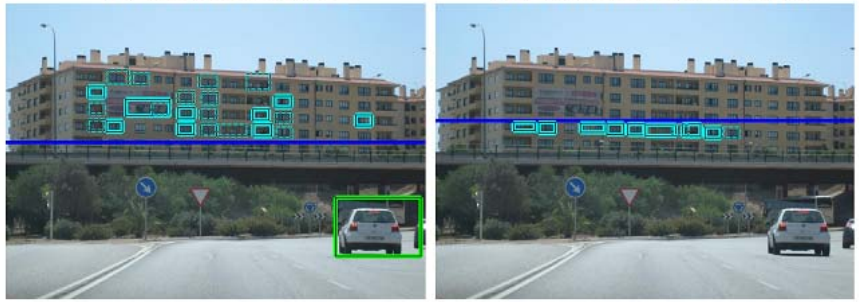
Sample Results



(b) Local Detection (b) Full Model Detection



(d) Local Detection (d) Full Model Detection



(f) Local Detection (f) Full Model Detection

[Hoiem 2006]

Results:

	Cars			Pedestrians		
	1FP	5FP	10FP	1FP	5FP	10FP
+Geom	6.6%	5.6%	7.0%	7.5%	8.5%	17%
+View	8.2%	16%	22%	3.2%	14%	23%
+GeomView	**12%**	**22%**	**35%**	**7.2%**	**23 %**	**40%**

Table 1. Modeling viewpoint and surface geometry aids object detection. Shown are percentage reductions in the missed detection rate while fixing the number of false positives per image.

[Hoiem 2006]

Results

	Mean	Median
Prior	10.0%	8.5%
+Obj	7.5%	4.5%
+ObjGeom	7.0%	3.8%

Table 2. Object and geometry evidence improve horizon estimation. Mean/median absolute error (as percentage of image height) are shown for horizon estimates.

[Hoiem 2006]

Results

	Horizon	Cars (FP)		Ped (FP)	
Car	7.3%	5.6	7.4	—	—
Ped	5.0%	—	—	12.4	13.7
Car+Ped	3.8%	5.0	6.6	11.0	10.7

Table 3. Horizon estimation and object detection are more accurate when more object models are known. Results shown are using the full model in three cases: detecting only cars, only pedestrians, and both. The horizon column shows the median absolute error. For object detection we include the number of false positives per image at the 50% detection rate computed over all images (first number) and the subset of images that contain both cars and people (second number).

[Hoiem 2006]

Results

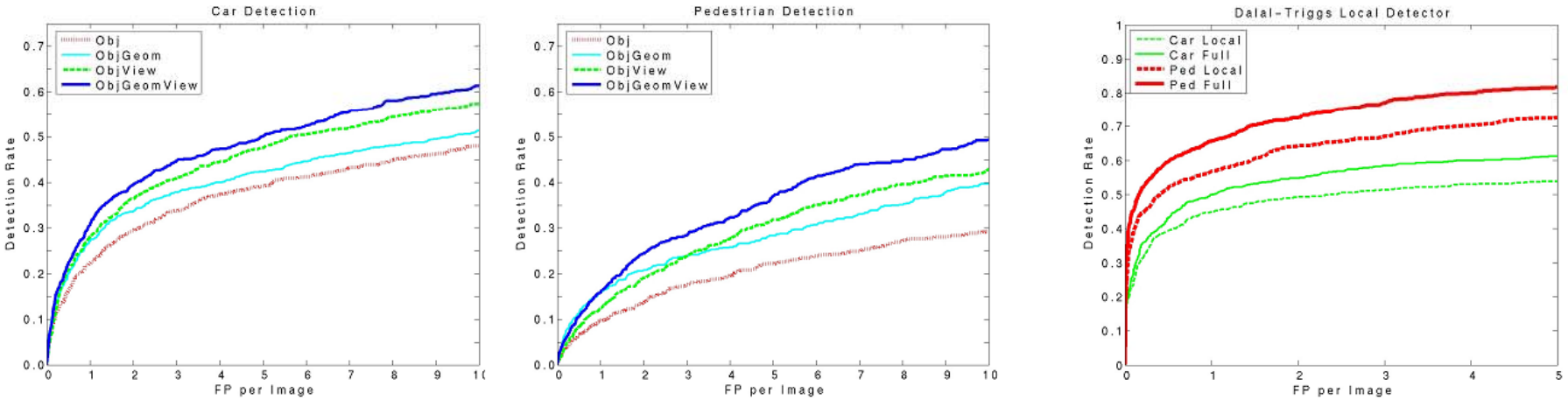
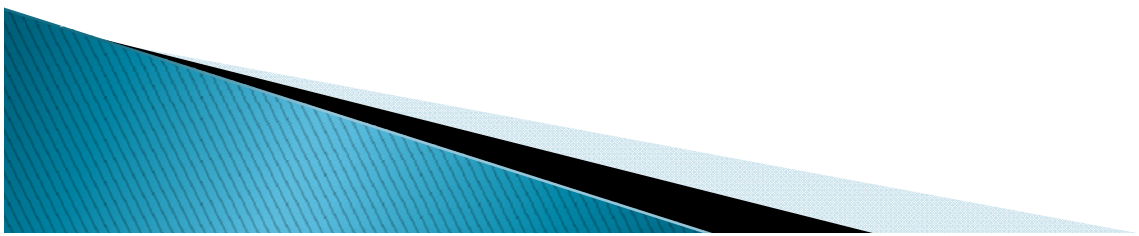


Figure 6. Considering viewpoint and surface geometry improves results over purely local object detection. The left two plots show object detection results using only local object evidence (Obj), object and geometry evidence (ObjGeom), objects related through the viewpoint (ObjView), and the full model (ObjViewGeom). On the right, we plot results using the Dalal-Triggs local detector [6].

[Hoiem 2006]

Result Highlights

- Including viewpoint and surface geometry estimates nets 20% reduction in false negatives.
- Reduces horizon estimation error by 3%.
- Including more object types improves performance.



Caveats

- Elevation:
  - Has trouble with unusual object placement, because it assumes everything is on the ground plane.

Kill Bill, Miramax Films

Caveats (cont)

- Ground Slope:
  - "If the ground is sloped, as in Figure 2, the coordinates and parameters are computed with respect to that slope, and the relationship between viewpoint and objects in the image still holds."

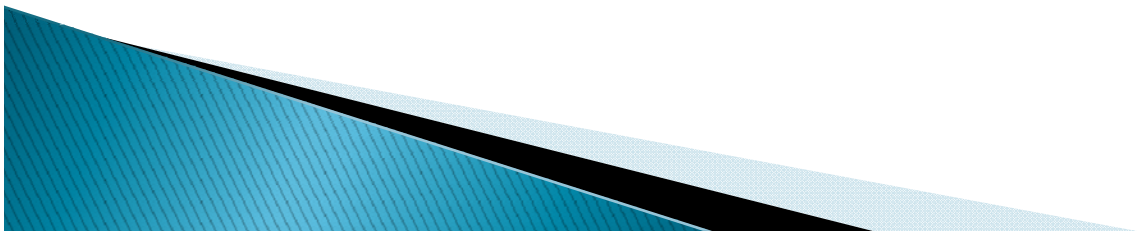
  Maybe for cars, but people stand upright regardless of local slope!

- Assumes things are 2D billboards.

Figure 2(a) [Hoiem 2006]

Thoughts

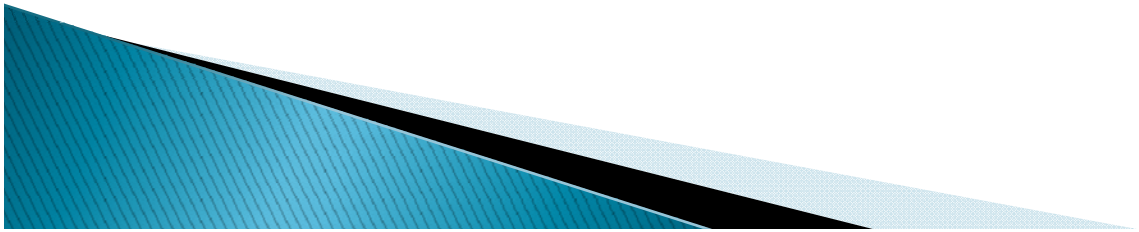
- Combine perspective context and stuff context?
- Estimate angle of observed object for better viewpoint estimation?



References

- Learning Spatial Context: Using Stuff to Find Things, by G. Heitz and D. Koller, ECCV 2008.

- Putting Objects in Perspective, by D. Hoiem, A. Efros, and M. Hebert, CVPR 2006.



Thanks!