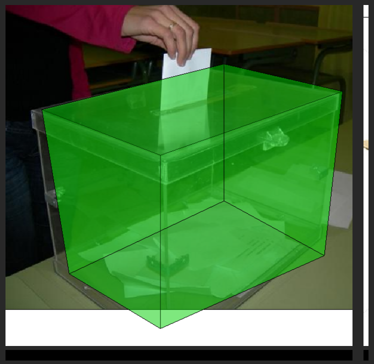
Hands, objects, and videotape:
recognizing object interactions from streaming wearable cameras

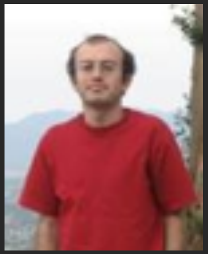
Deva Ramanan
UC Irvine



Students doing the work

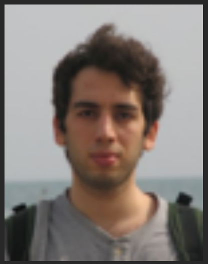
Greg Rogez

Post-doc, looking to come back to France!



Hamed Pirsiavash

Former student, now post-doc at MIT



Mohsen Hejrati

Current student

Motivation 1: integrated perception and actuation

Motivation 2: wearable (mobile) cameras

Google Glass

Outline

-Egocentric hand estimation

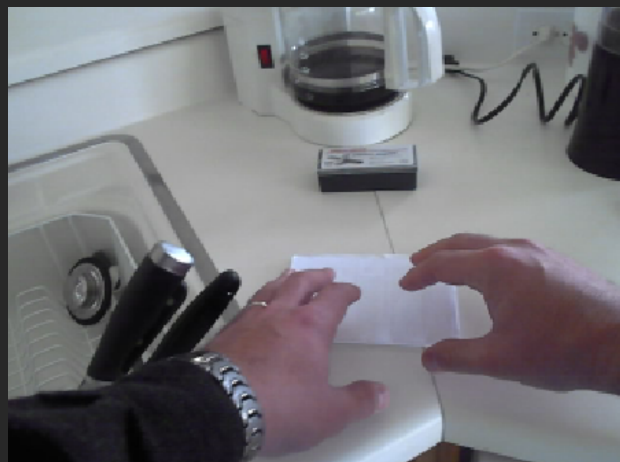
-Data analysis:
Analyze big temporal data

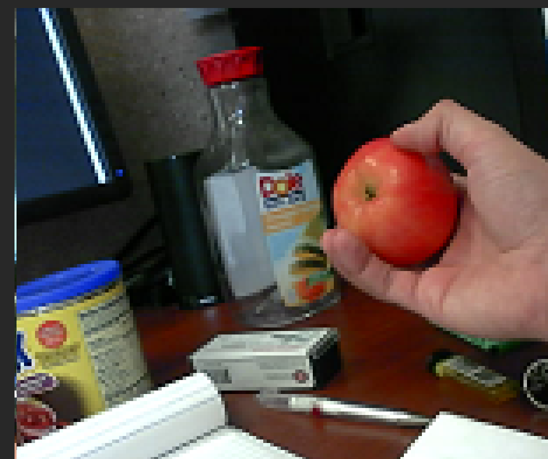
“Making tea”

-Functional prediction:
what can user do in scene?

Grab here

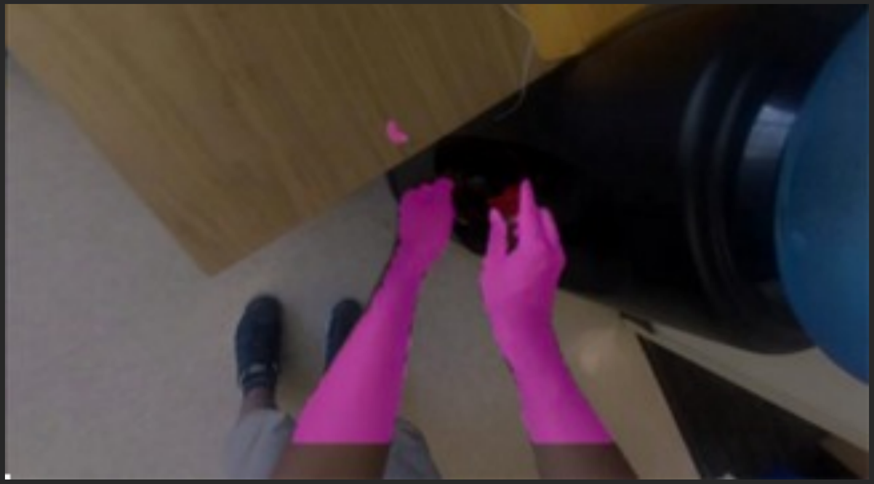
Egocentric hand pose estimation





Challenges: -hands have a higher (effective) DOFs than bodies
-self-occlusion due to egocentric viewpoint
-occlusions to objects

Past approaches



Skin-pixel classification: Li & Kitani, CVPR13, ICCV13

Motion segmentation: Ren & Gu, CVPR10, Fathi et al CVPR 11

Observation: RGB-D saves the day

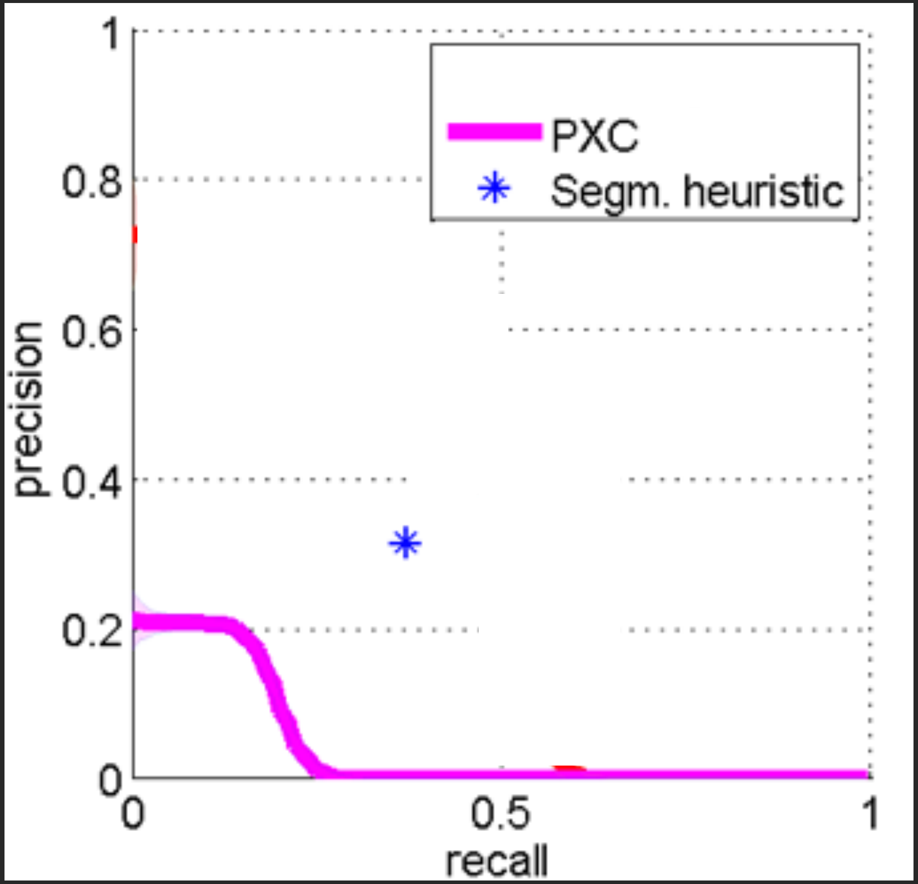
Produces accurate depth over “near-field workspace”

Mimic near-field depth from human vision (stereopsis)

TOF camera

Does depth solve it all?

Hand detection in egocentric views



PXC = Intel’s Perceptual Computing Software

Our approach

Make use of massive synthetic training set

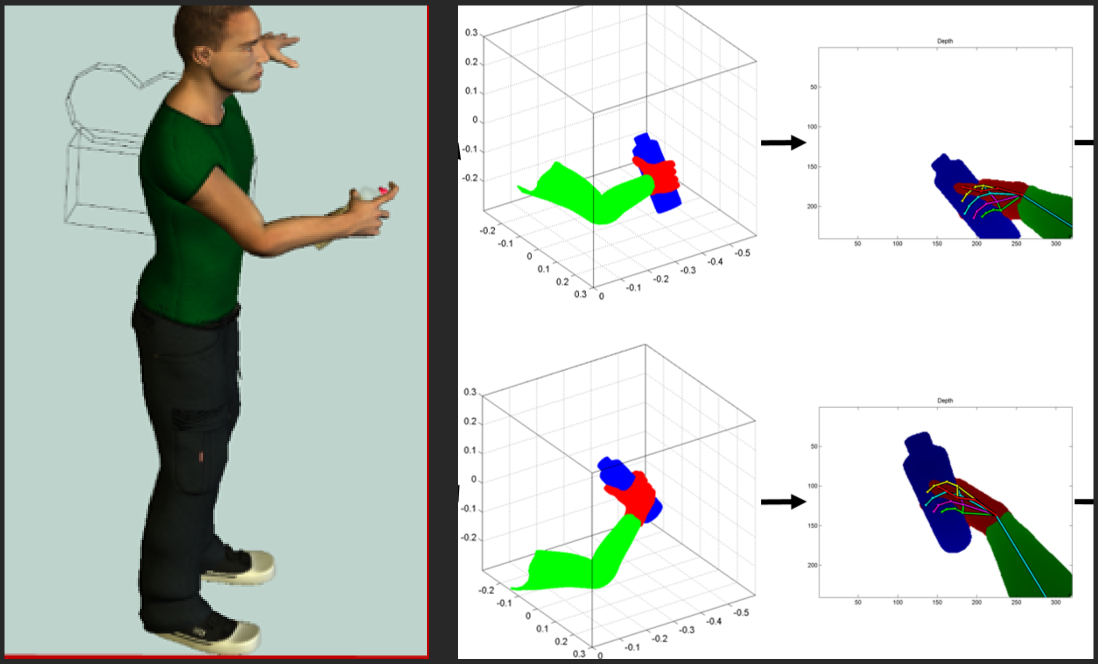
Mount avatar with virtual egocentric cameras

Use animation library of household objects and scenes

Our approach

Make use of massive synthetic training set

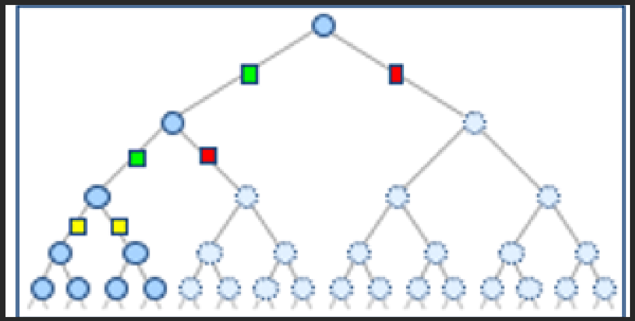
Mount avatar with virtual egocentric cameras



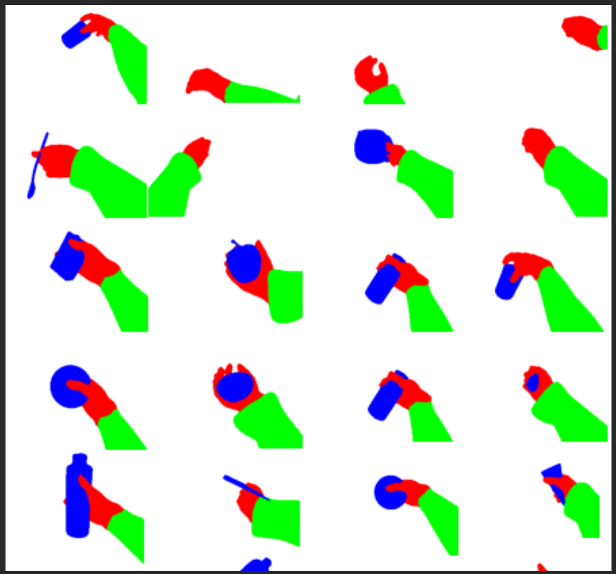
Use animation library of household objects and scenes

Naturally enforces “egocentric” priors over viewpoint, grasping poses, etc.

Recognition

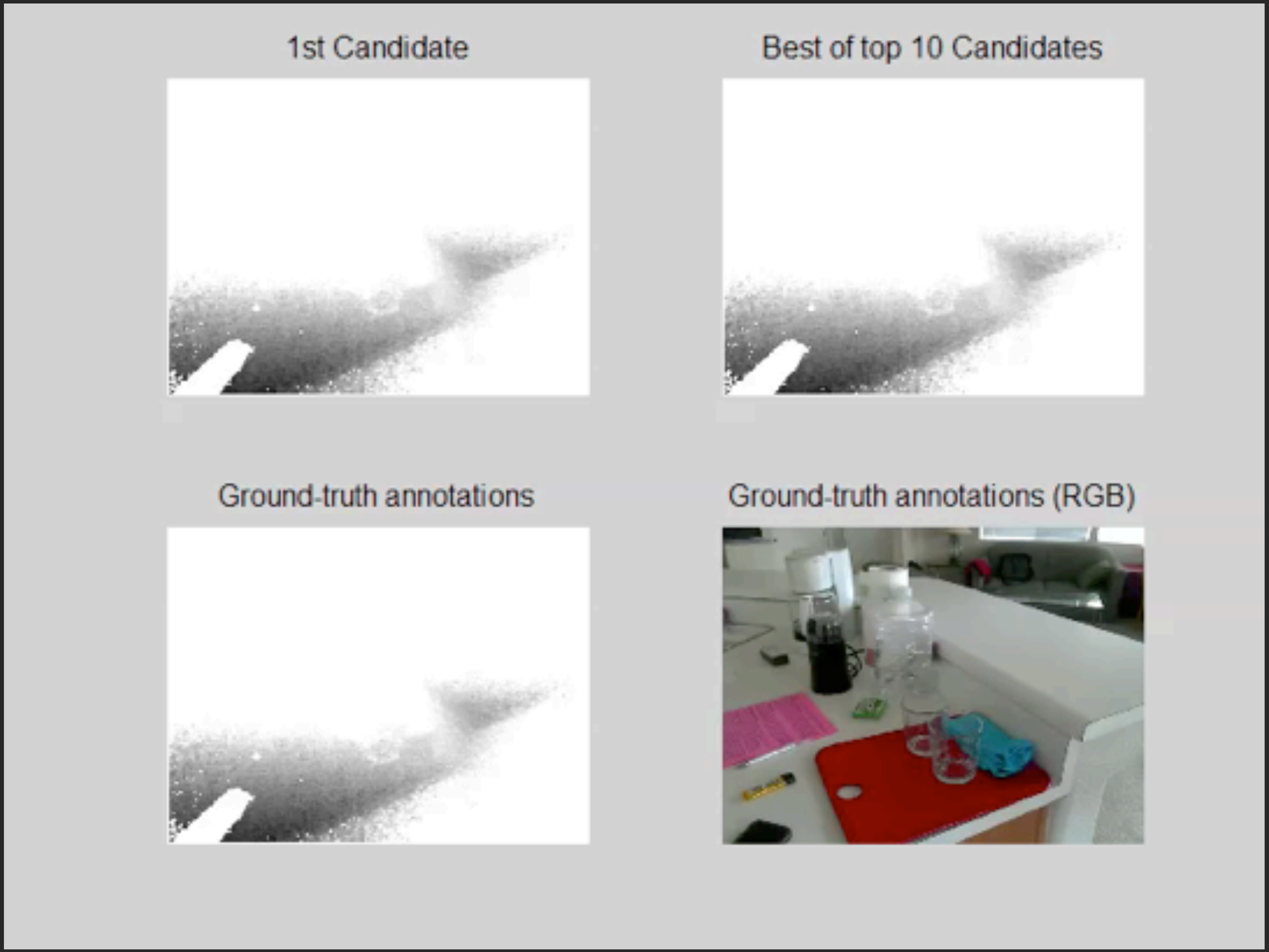


Decision / regression trees



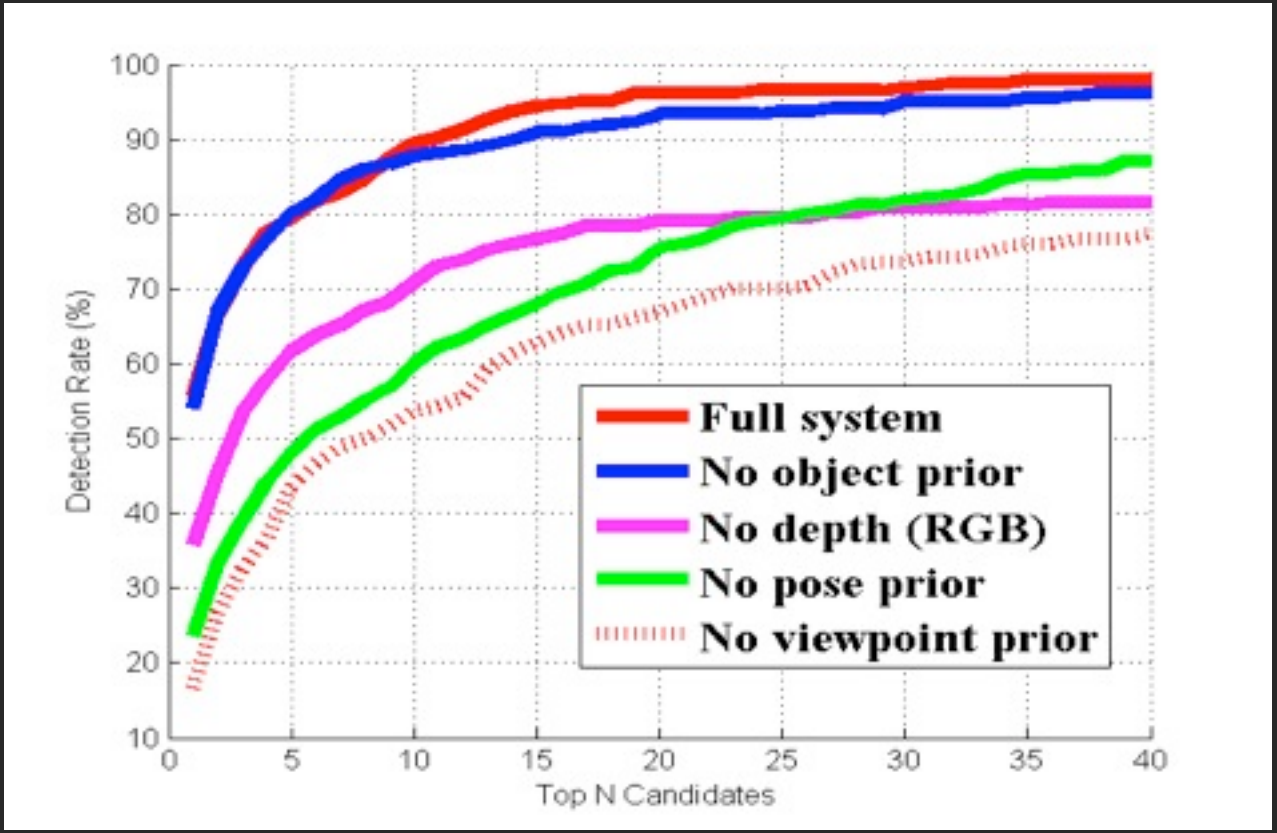
Nearest-neighbor on volumetric depth features

Results



Rogez et al, ECCV 14 Workshop on Consumer Depth Cameras

Ablative analysis

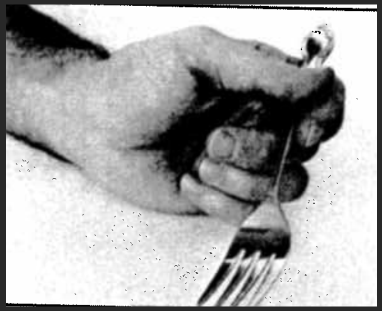


Depth & egocentric priors (over viewpoint & grasping poses) are crucial

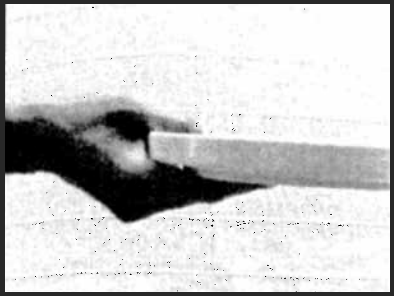
Ongoing work: hand grasp region prediction

Functionally-motivated pose classes

Disc grasp



Dynamic lateral tripod



Lumbrical grasp

(Though we are finding it hard to publish in computer vision venues!)

Outline

-Egocentric hand estimation

-Data analysis:
Analyze big temporal data

-Functional prediction:
what can user do in scene?

Grab here

Temporal data analysis

Challenges: -analyze large collections of temporal big-data vs YouTube clips

- some daily activities can take a long time (interrupted)
- some daily activities exhibit “internal structure” (more on this)

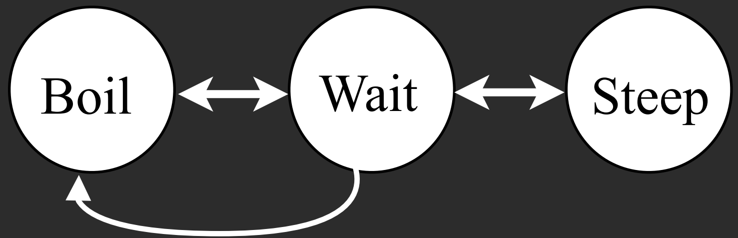
Start boiling water

Do other things (while waiting)

Pour in cup

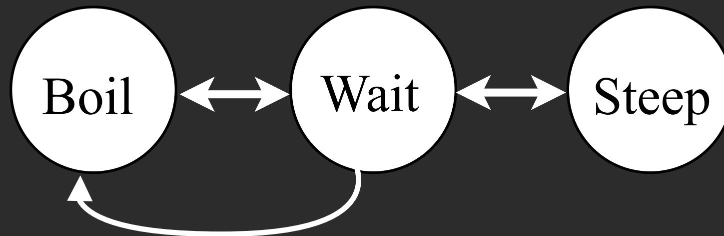
Drink tea

Classic models for capturing temporal structure



Markov models

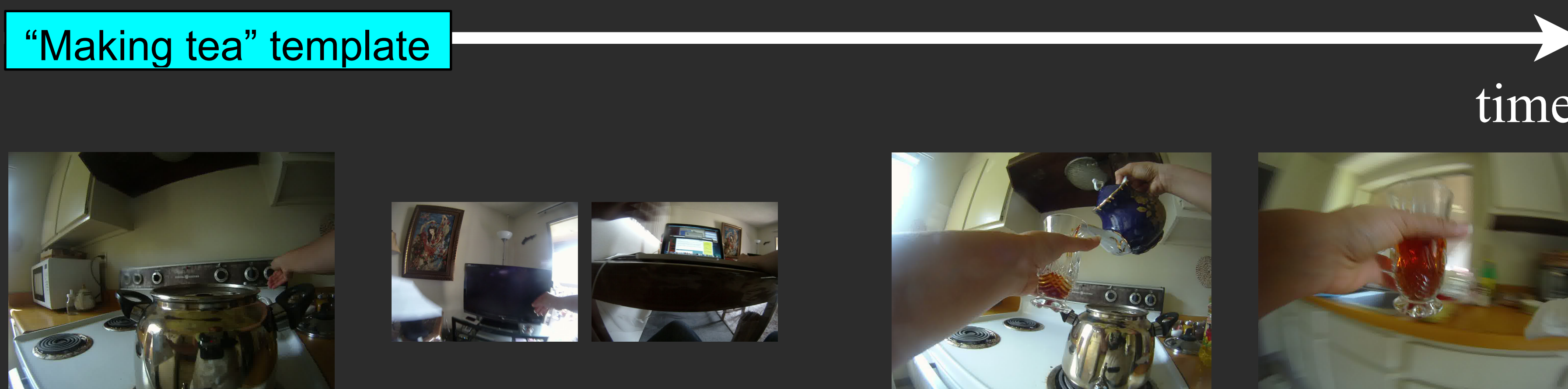
Classic models for capturing temporal structure



Markov models

... but does this *really* matter?

Maybe local bag-of-feature templates suffice

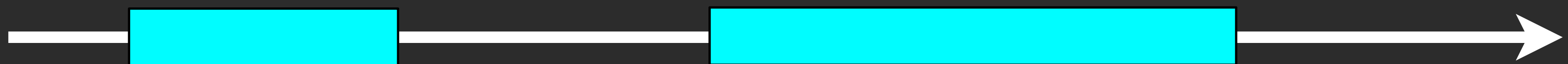


P. Smyth “Oftentimes a strong data model will do the job”

But some annoying details...

How to find *multiple* actions of differing lengths?
Can we do better that window scan of O(NL) + heuristic NMS ?

N = length of video
L = maximum temporal length

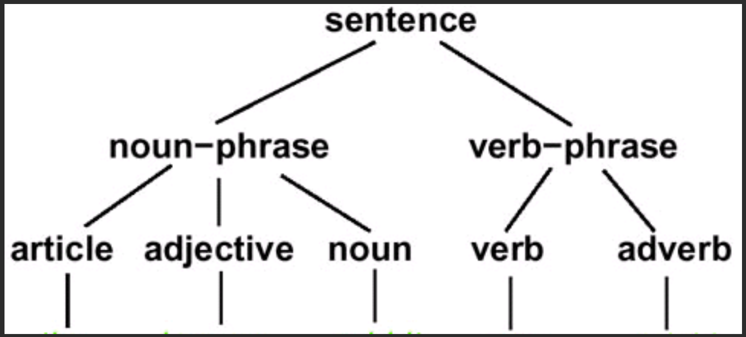






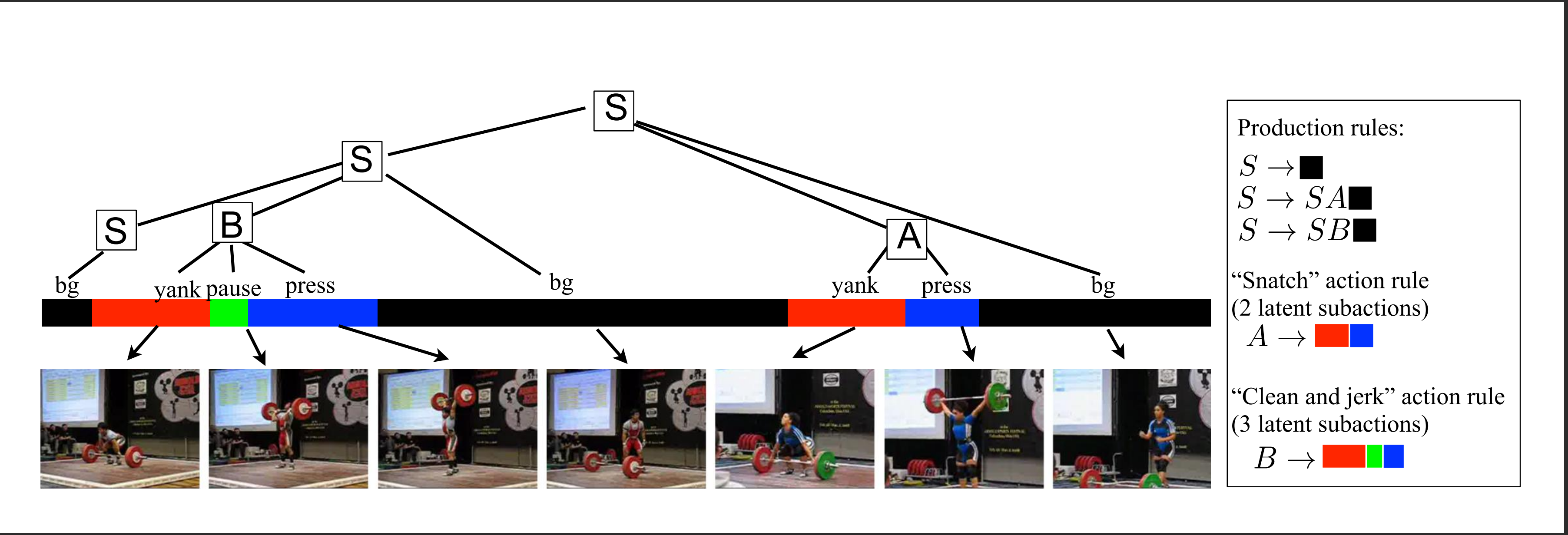
Insufficiently well-known fact

We can do all this for 1-D (temporal) signals with grammars



"The hungry rabbit eats quickly"

Application to actions



Context-free grammars (CFGs): surprisingly simple to implement but poor scalabity $O(N^3)$

Our contribution: many restricted grammars (like above) can be parsed in O(NL)

In theory & practice, no more expensive than a sliding window!

Real power of CFGs: recursion

e.g., rules for generating valid sequences of parenthesis

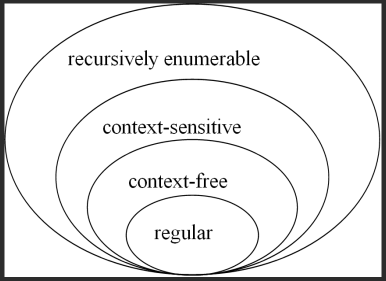
“((()())())()”

$$S \to \{\}$$

$$S \to (S)$$

$$S \to SS$$

If we don’t make use of this recursion, we can often make do with a simpler grammar.

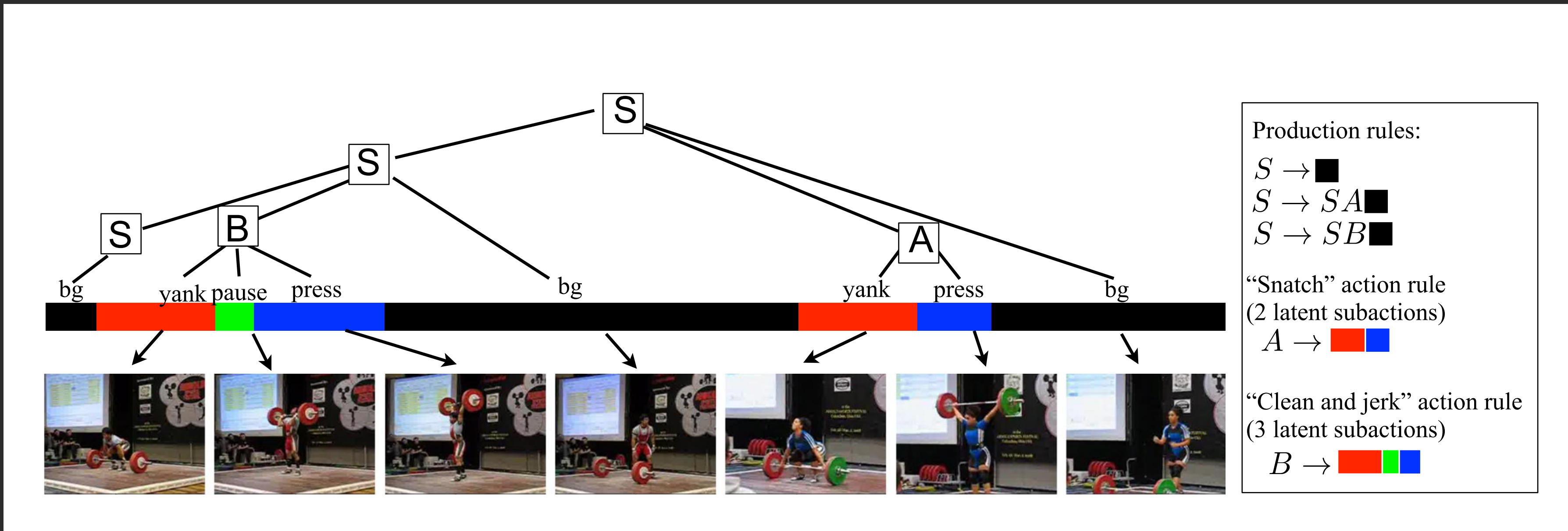


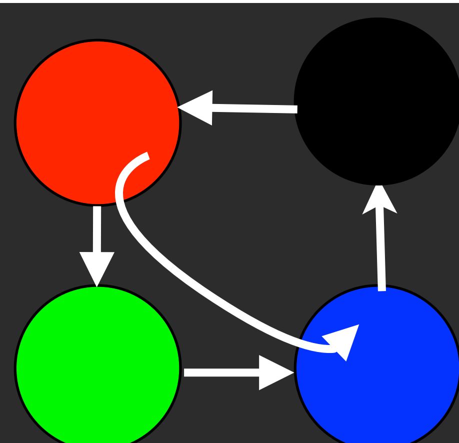
Regular grammar:

$$X \to uvw$$

$$X \to Yuvw$$

Intuition: compile regular grammar into a semi-markov model

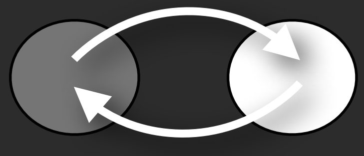




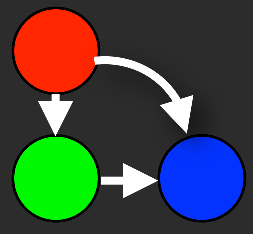
Semi-markov models = markov models with "counting" states

But aren't semi-markov models already standard?

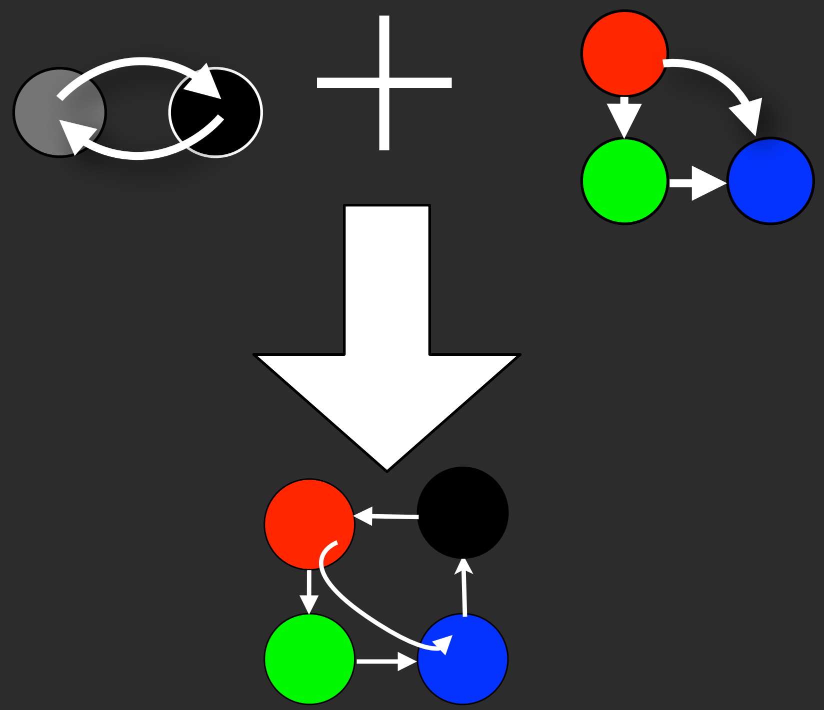
Action segmentation with 2-state semi-markov model:
(Shi et al IJCV10, Hoai et al CVPR11)



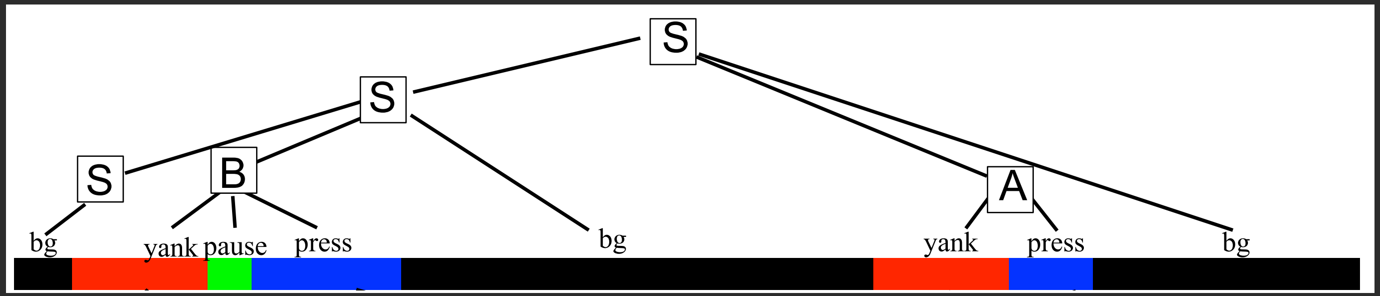
Model subactions with 3-state semi-markov model:
Tang et al CVPR12 (+ NMS?)



Our work

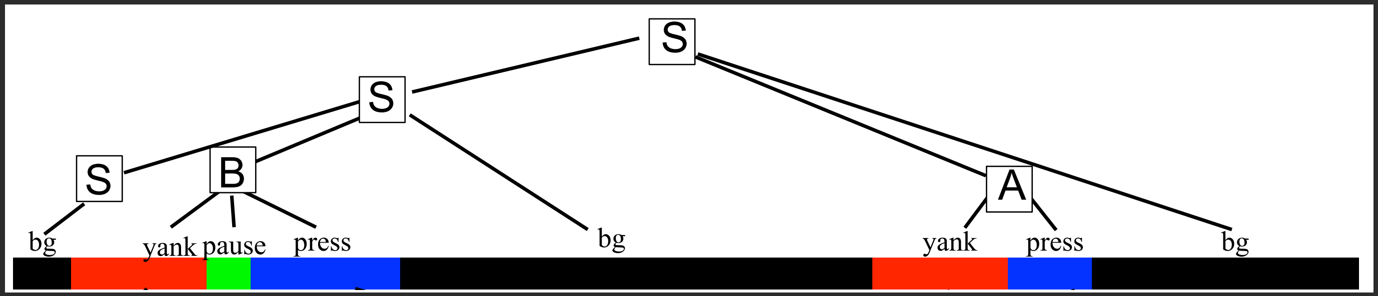


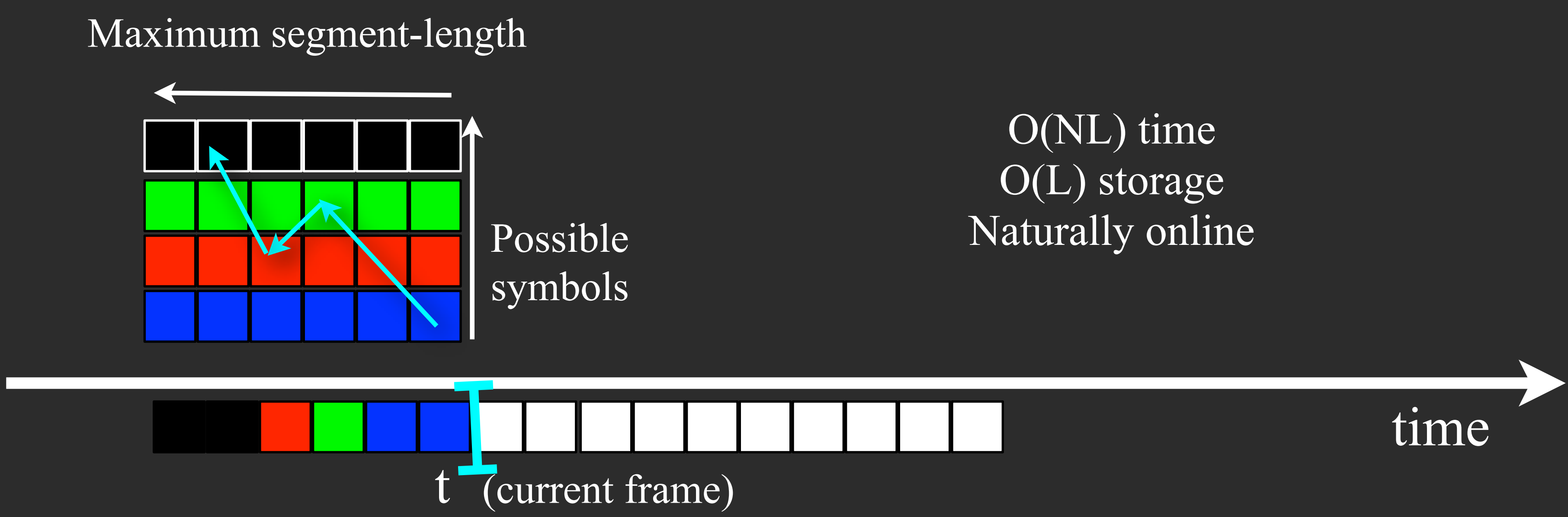
Single model enforces temporal constraints at multiple scales (actions, sub-actions)



Use production rules to implicitly manage additional dummy / counting states used by underlying markov model

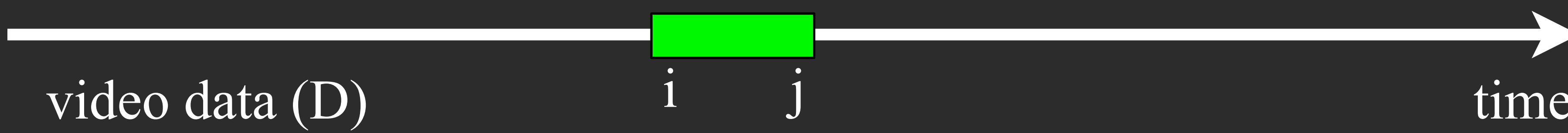
Inference





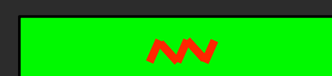
O(NL) time
O(L) storage
Naturally online

Scoring each segment



$$S(D, r, i, j) = \alpha_r \cdot \phi(D, i, j) + \beta_r \cdot \psi(j - i) + \gamma_r$$

α : data model 

β: prior over length of segment 

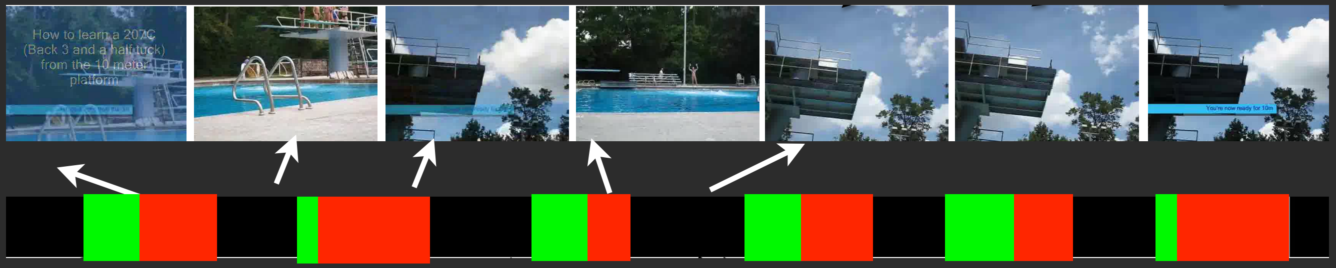
γ: prior of transition rule r = $X \rightarrow Y$ 

Learning

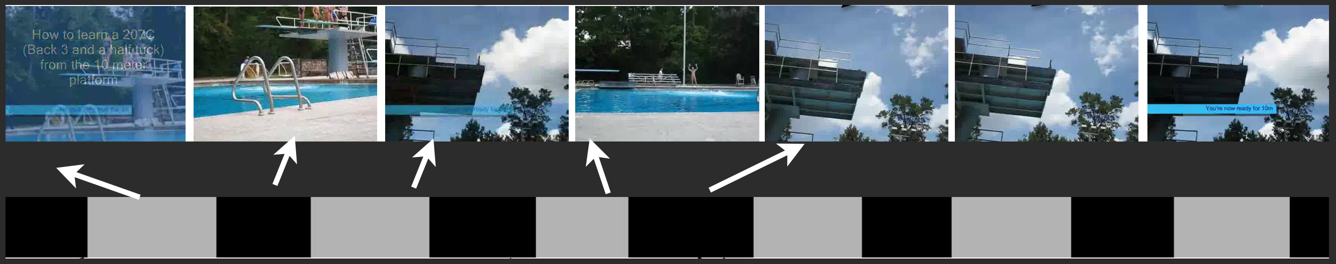
Score is linear in parameters
(segment data model α , segment length prior β, and rule transition prior γ)

Structured SVM solver

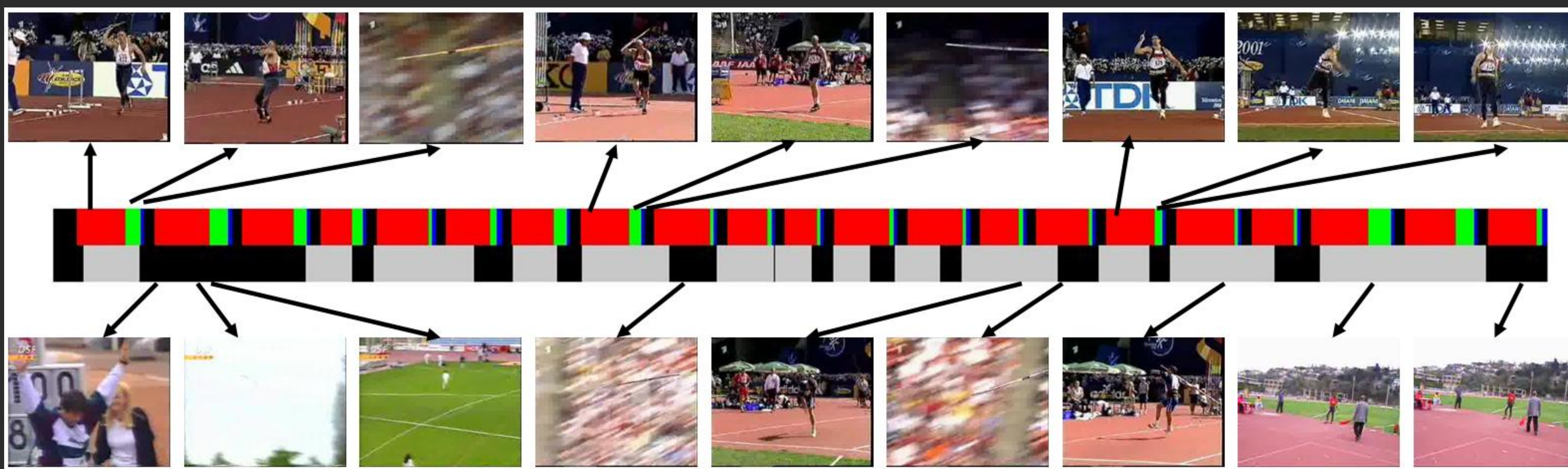
Supervised:



Weakly-supervised:
(Latent)

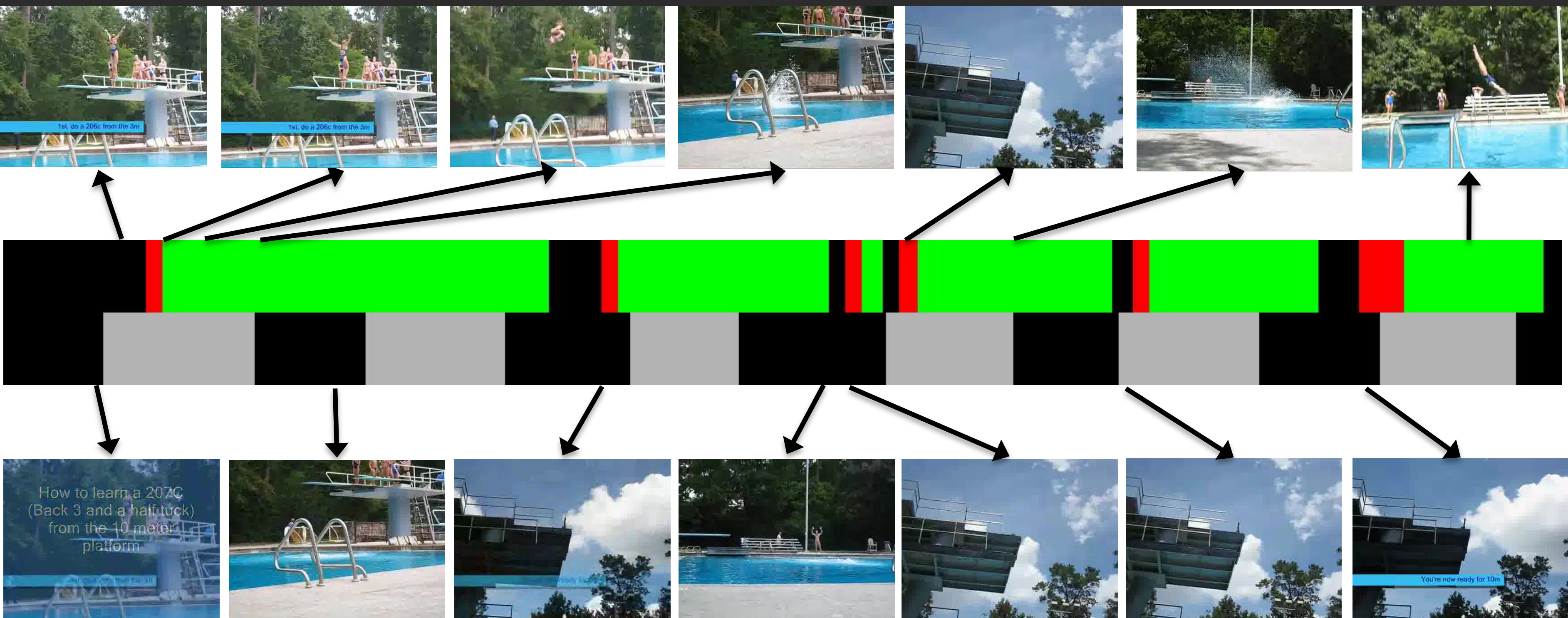


Results



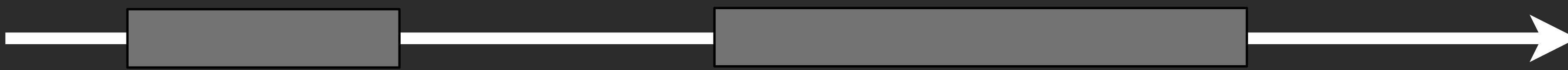
Latently inferred subactions appear to be run, release, and throw.

Results

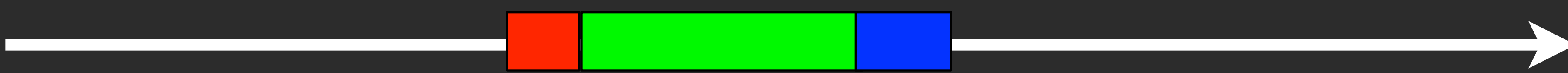


Latently inferred subactions appear to be bend and jump.

Baselines

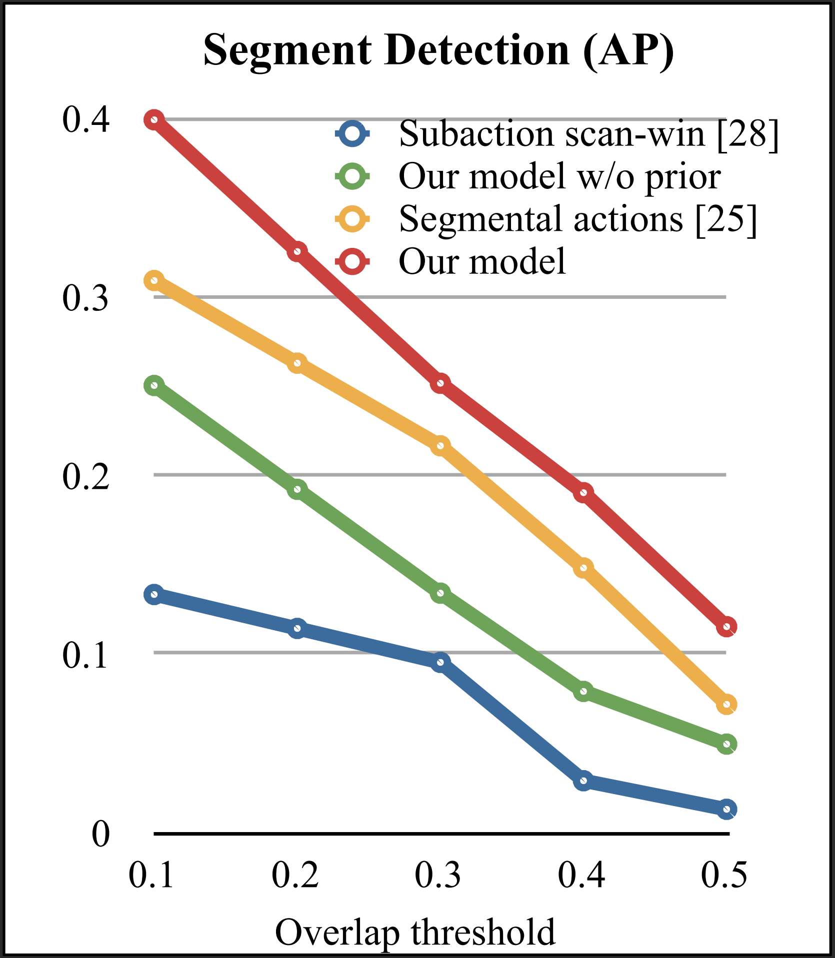


Action segmentation with 2-state semi-markov model:
(Shi et al IJCV10, Hoai et al CVPR11)



Model subactions with 3-state semi-markov model:
Tang et al CVPR12 + NMS

Results for action segment detection (AP)



Pirsiavash & Ramanan, CVPR14

Outline

-Egocentric hand estimation

-Data analysis:
Analyze big temporal data

“Making tea”

-Functional prediction:
what can user do in scene?

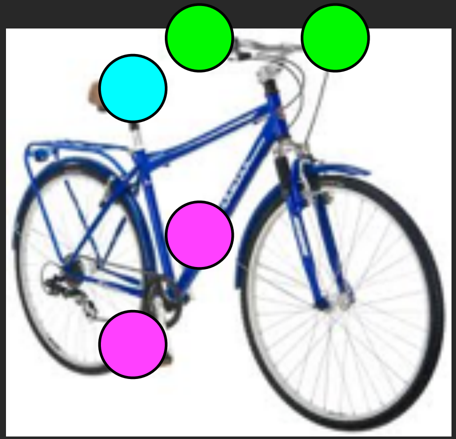
Grab here

Object touch (interaction) codes

Label object surfaces with body parts that come in contact with them

hands
mouth

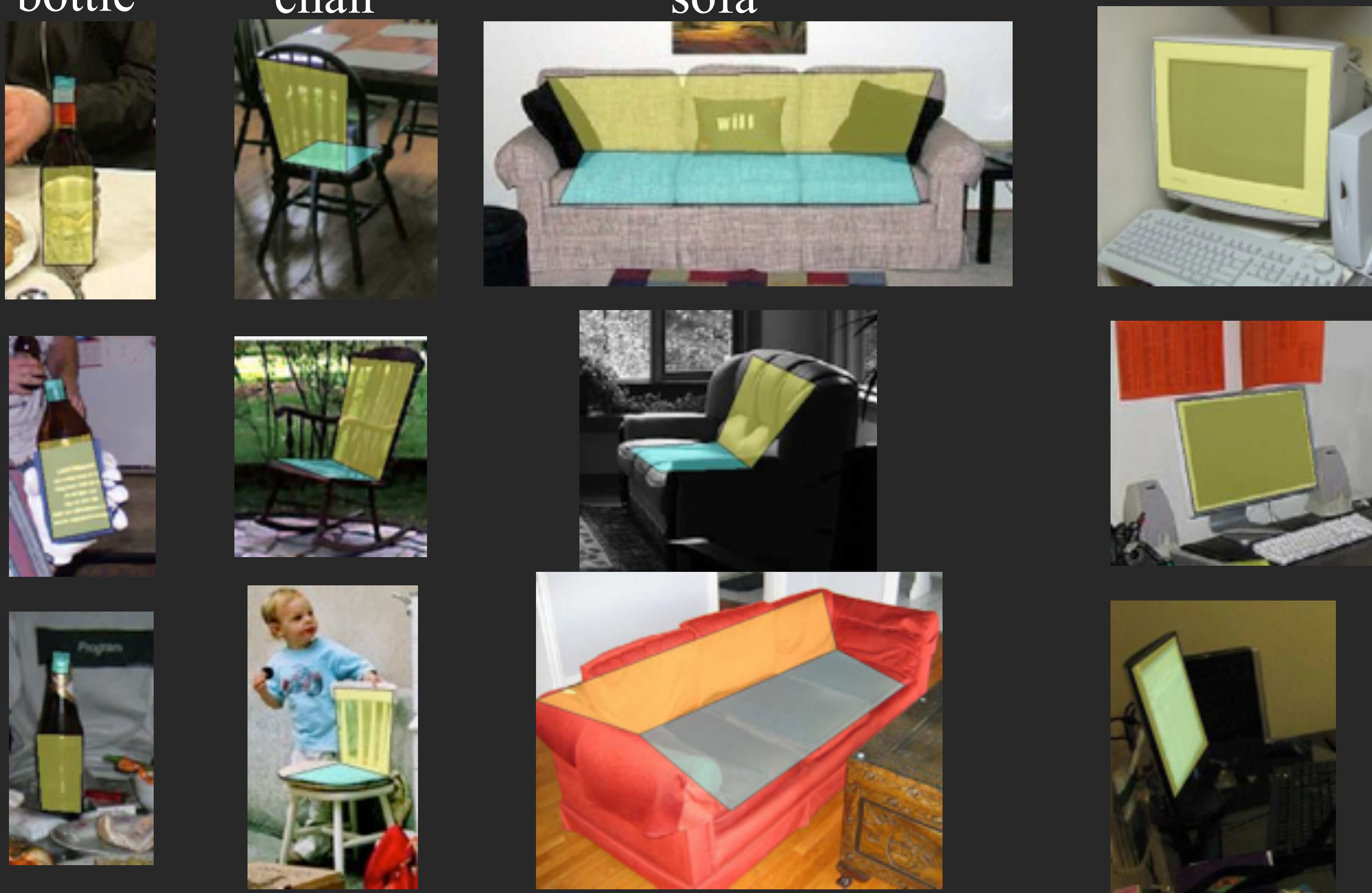
arms
back
bum



hands
bum
feet

Dataset of interaction region masks

bottle chair sofa monitor



Alternate perspective

Prediction of functional landmarks

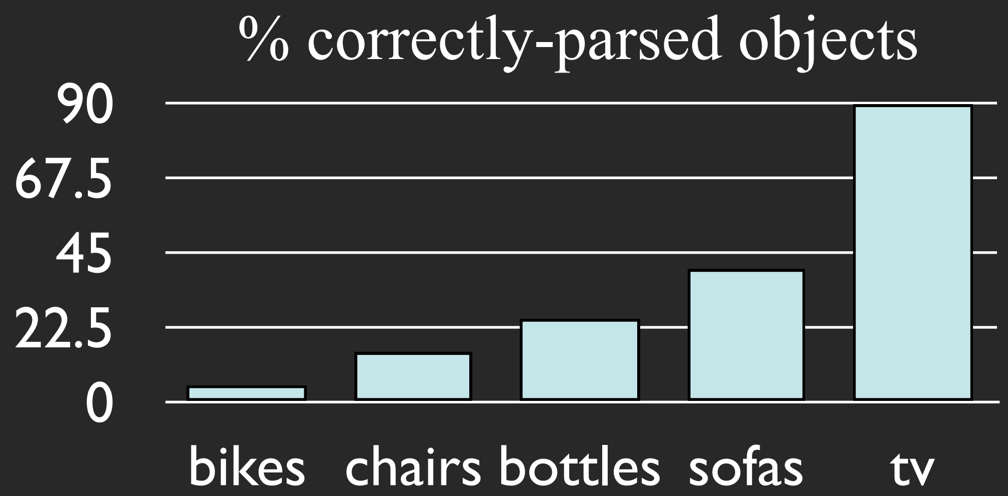
How hard is this problem?

Benchmark evaluation of several standard approaches

Blind regression (from bounding box coordinates)
Regression from part locations
Bottom-up geometric labeling of superpixels
Nearest neighbor matching + label transfer
...

Desai & Ramanan "Predicting Functional Regions of Objects"
Beyond Semantics Workshop, CVPR13

Some initial conclusions



-Difficulty varies greatly per object

Harder to ride a bike than sit on sofa (or watch TV)!

Blind prediction of bottle & TV regions works just as well as anything else

-Nearest neighbor + label transfer is the winner

Simple and works annoyingly well (though considerable room for improvement)

Strategic question

How to build models that produce detailed 3D landmark reports for general objects?

Recognition by 3D Reconstruction

Input:
2D image

Output:
3D shape
camera viewpoint

Overall approach: “brute-force” enumeration of 3D hypotheses

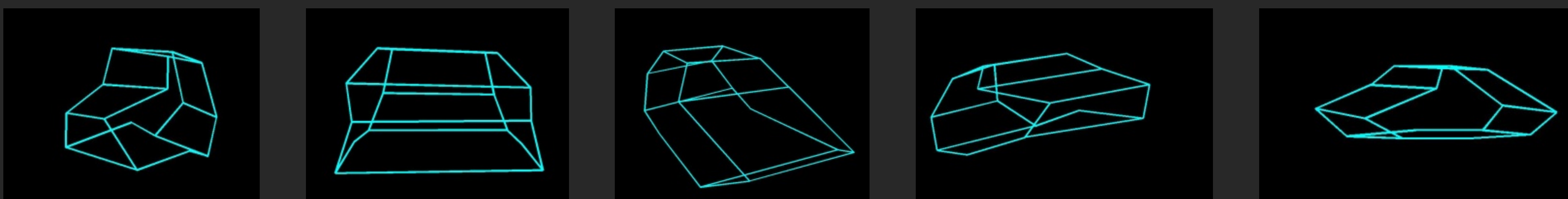
Enumerate hypotheses θ = (shape,viewpoint) and rendered HOG images $w(\theta)$

Find one that correlates best with query image

A model of 3D shape and viewpoint

1) 3D shape of object = linear combinations of 3D basis shapes

$$B = \sum_i \alpha_i B_i$$



2) Standard perspective camera model

$$p(\theta) \sim C(R, t, f)B$$

$$\theta = (\alpha, R, t, f)$$

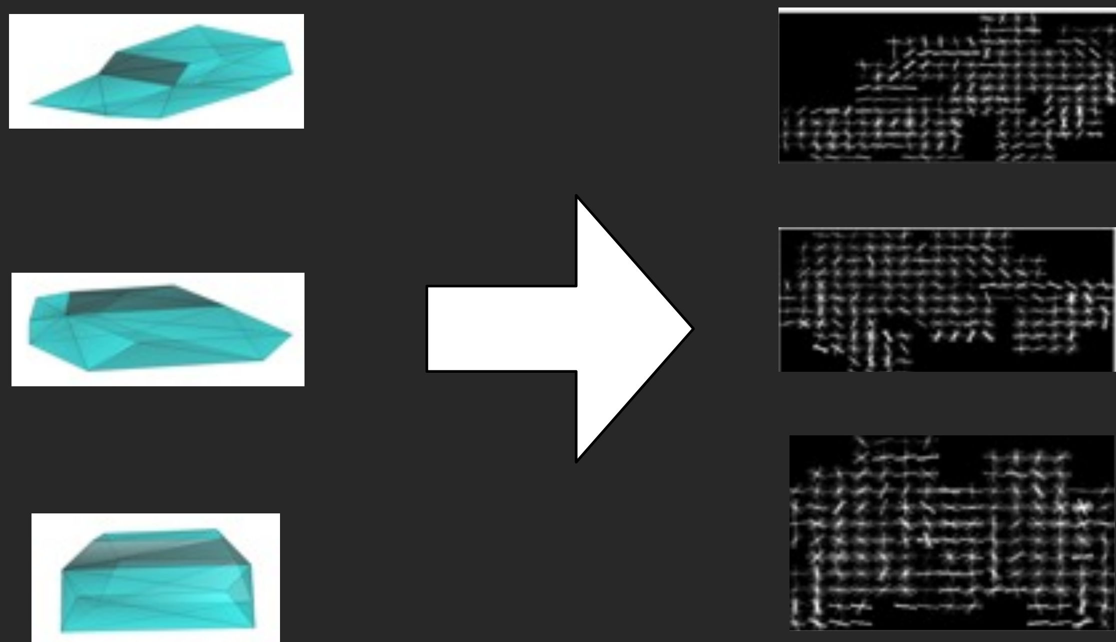
(shape coefficients, camera rotation, translation, focal length)

View & shape-specific templates

θ_1 $w(\theta_1)$

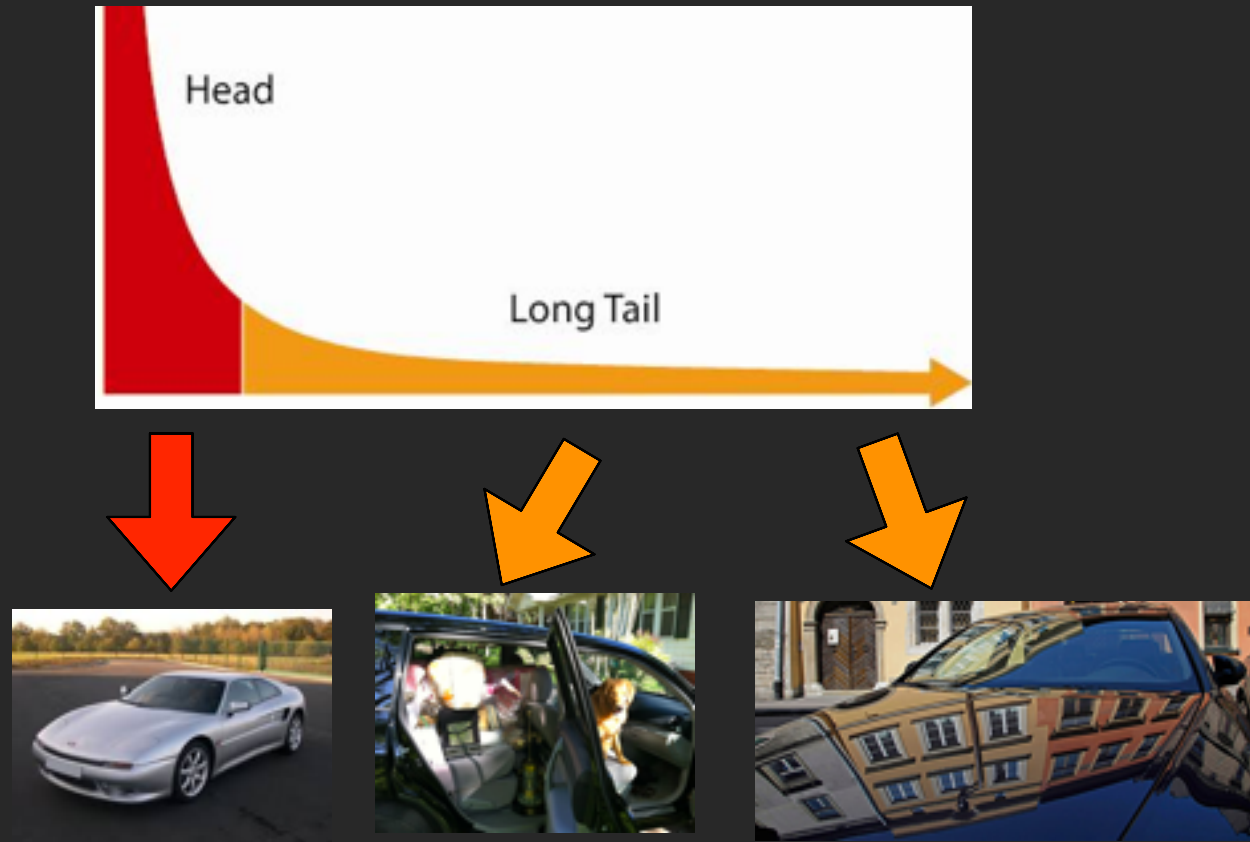
θ_2 $w(\theta_2)$

θ_3 $w(\theta_3)$



Treat each θ_n as unique subcategory (e.g., side-view SUVs) and learn template for it

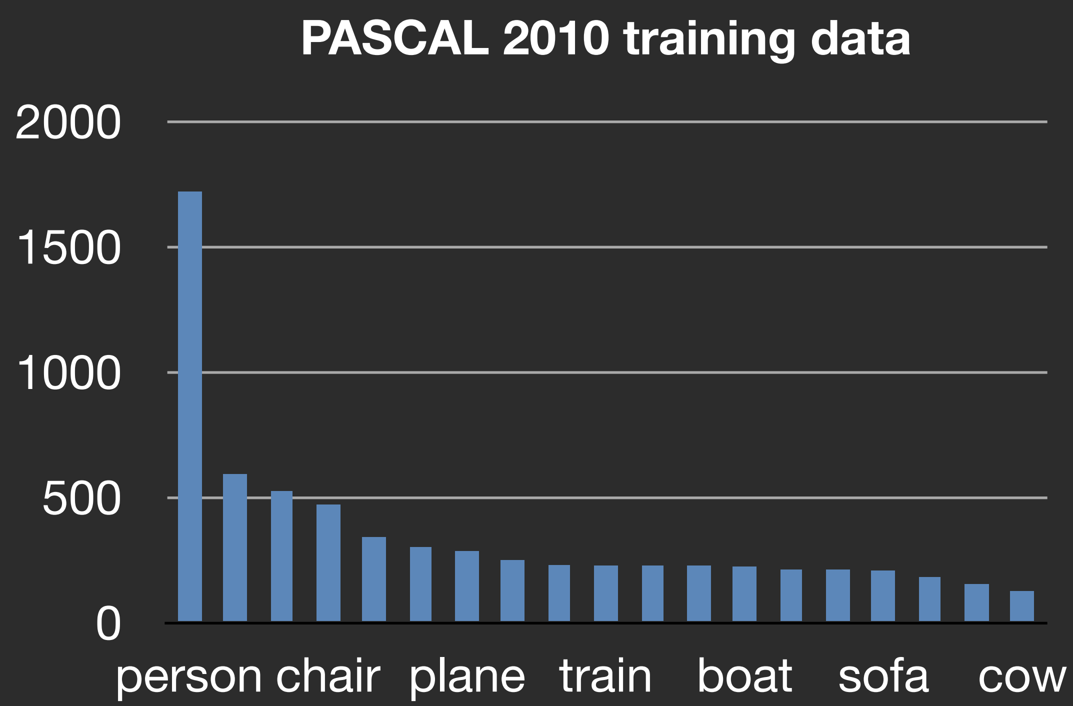
Challenge: rare shapes & views



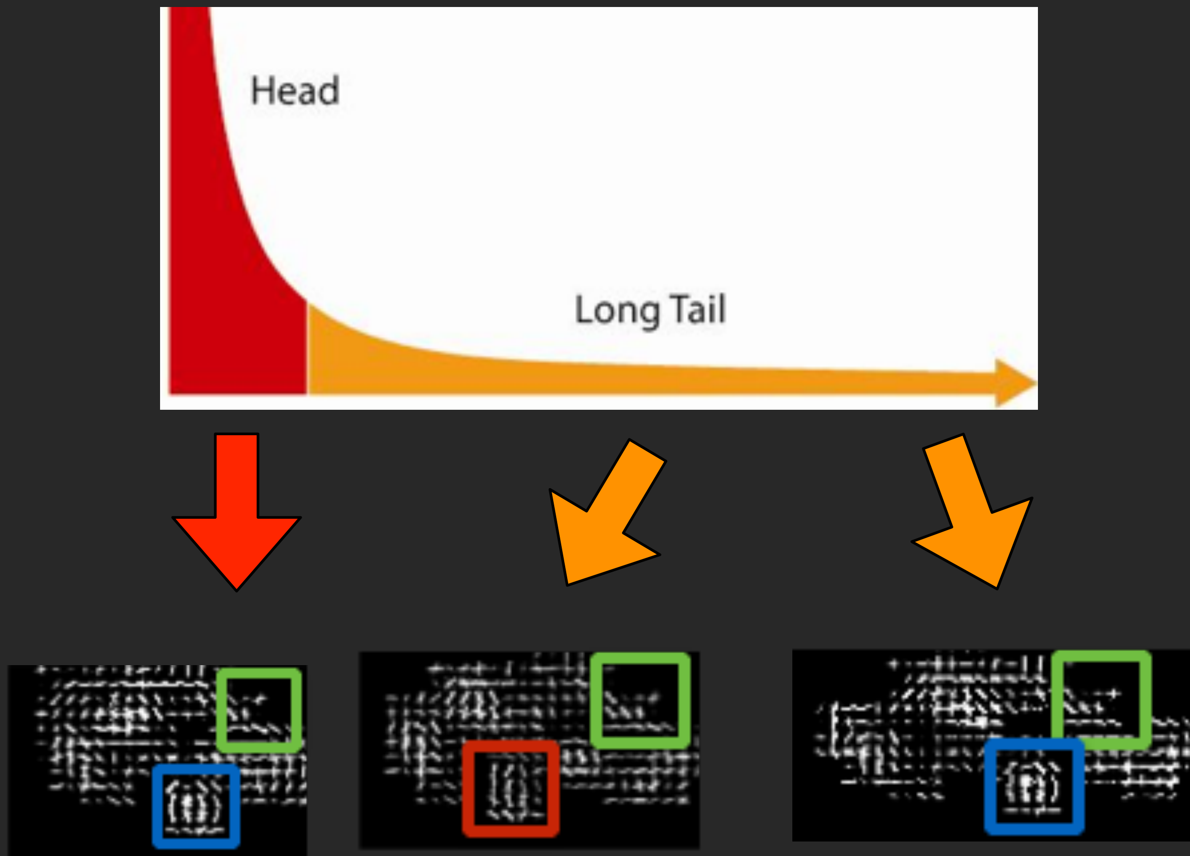
We need lots of templates, but will likely have little data of 'rare' car views

Zhu, Anguelov, & Ramanan "Capturing long-tail distributions of object subcategories" CVPR14

Long tail distributions of categories (cf. LabelMe)



Soln: share information with parts



Use ‘wheels’ from common views/shapes to help model rare ones

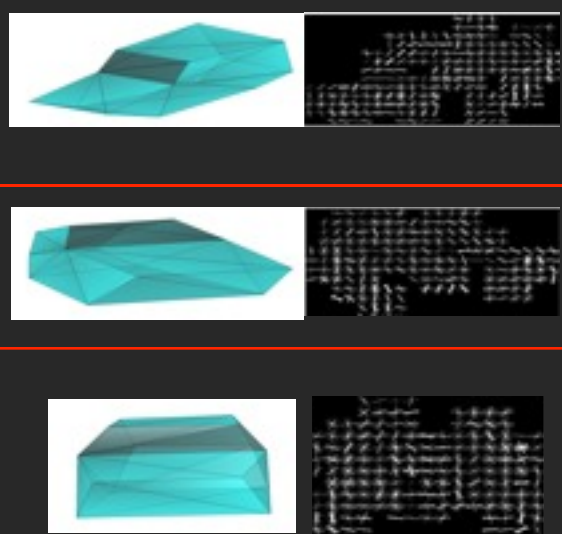
Some formalities

Cast recognition and reconstruction as a maximization problem

θ_1

θ_2

θ_3

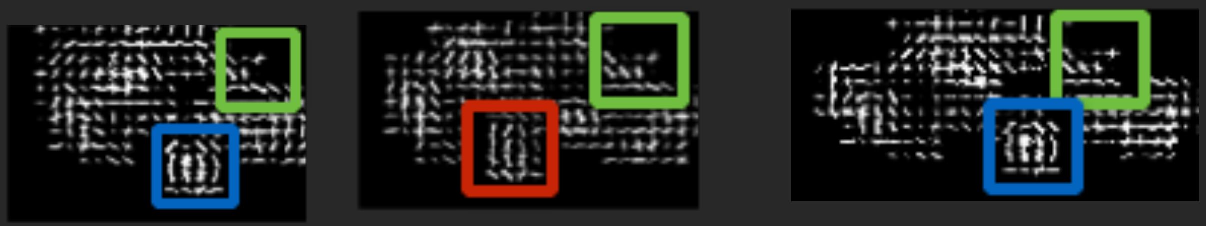


$\vdots$

$$S(I, \theta) = w(\theta) \cdot I$$

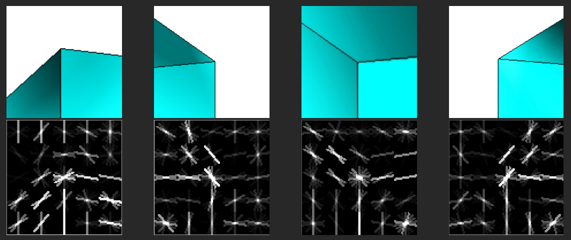
$$\theta^* = \arg\max_{\theta \in \Omega} S(I, \theta)$$

Templates with shared parts

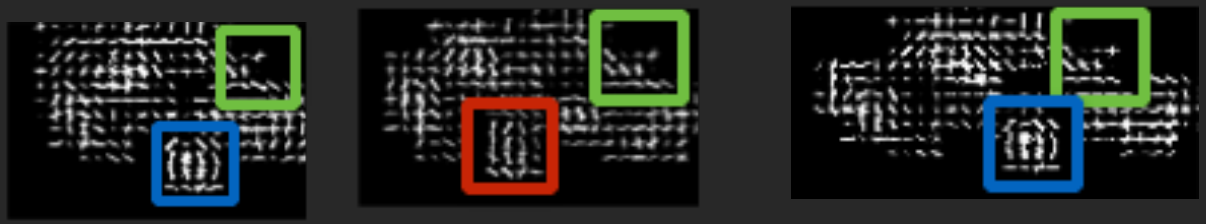


$$S(I, \theta) = \sum_{i \in V(\theta)} w_i^{m_i(\theta)} \cdot \phi(I, p_i(\theta))$$

$$\left.\begin{array}{ll} V: & \text{set of visible parts} \\ m_i: & \text{local mixture of part i} \\ p_i: & \text{pixel location of part i} \end{array}\right\} \text{all depend on } \theta$$



Templates with shared parts



$$S(I,\theta) = \sum_{i \in V(\theta)} w_i^{m_i(\theta)} \cdot \phi(I, p_i(\theta))$$

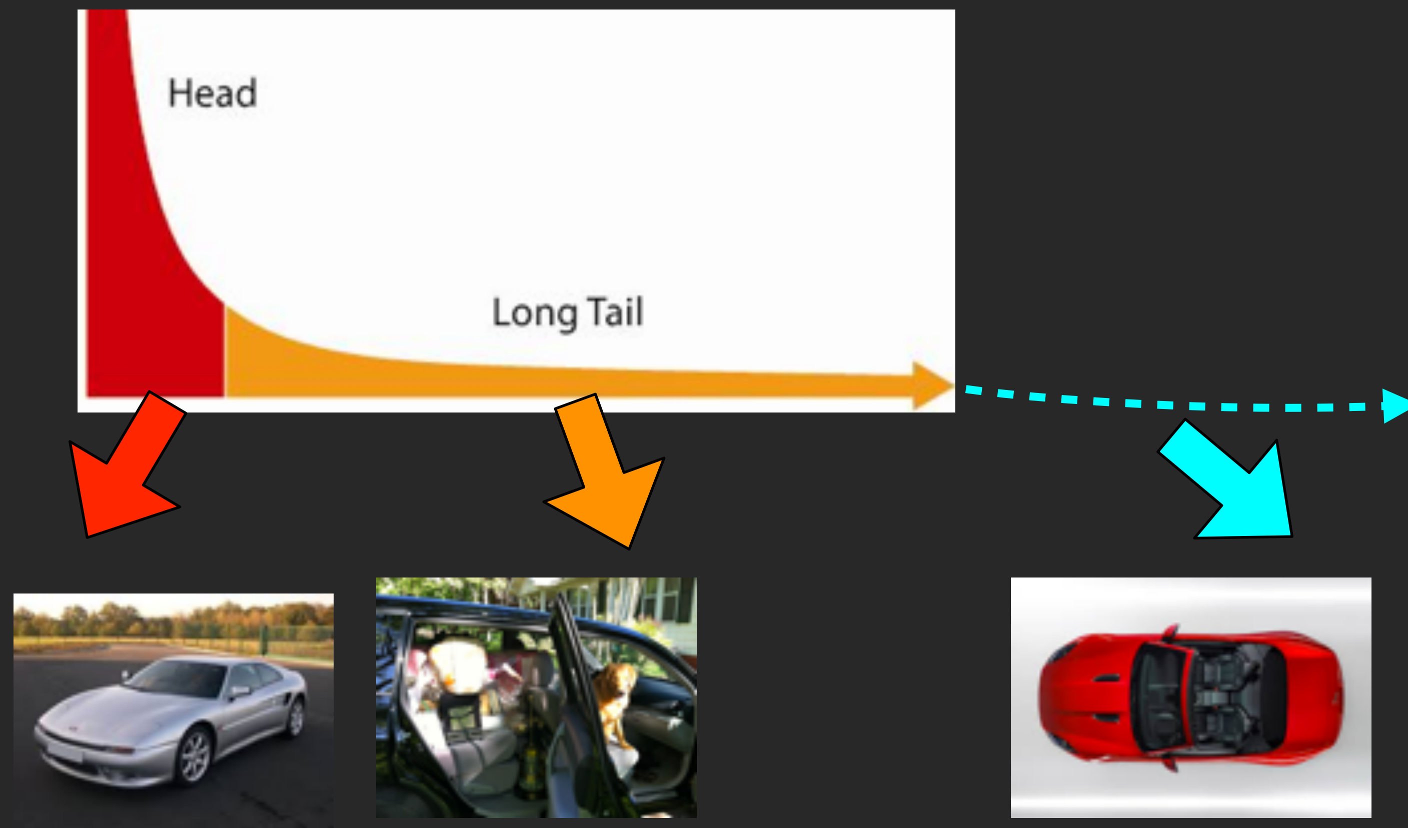
$$\theta^* = \arg\max_{\theta \in \Omega} S(I,\theta)$$

How do we define set of valid $\theta \in \Omega$?

One option: just use set of shapes/views observed in training set

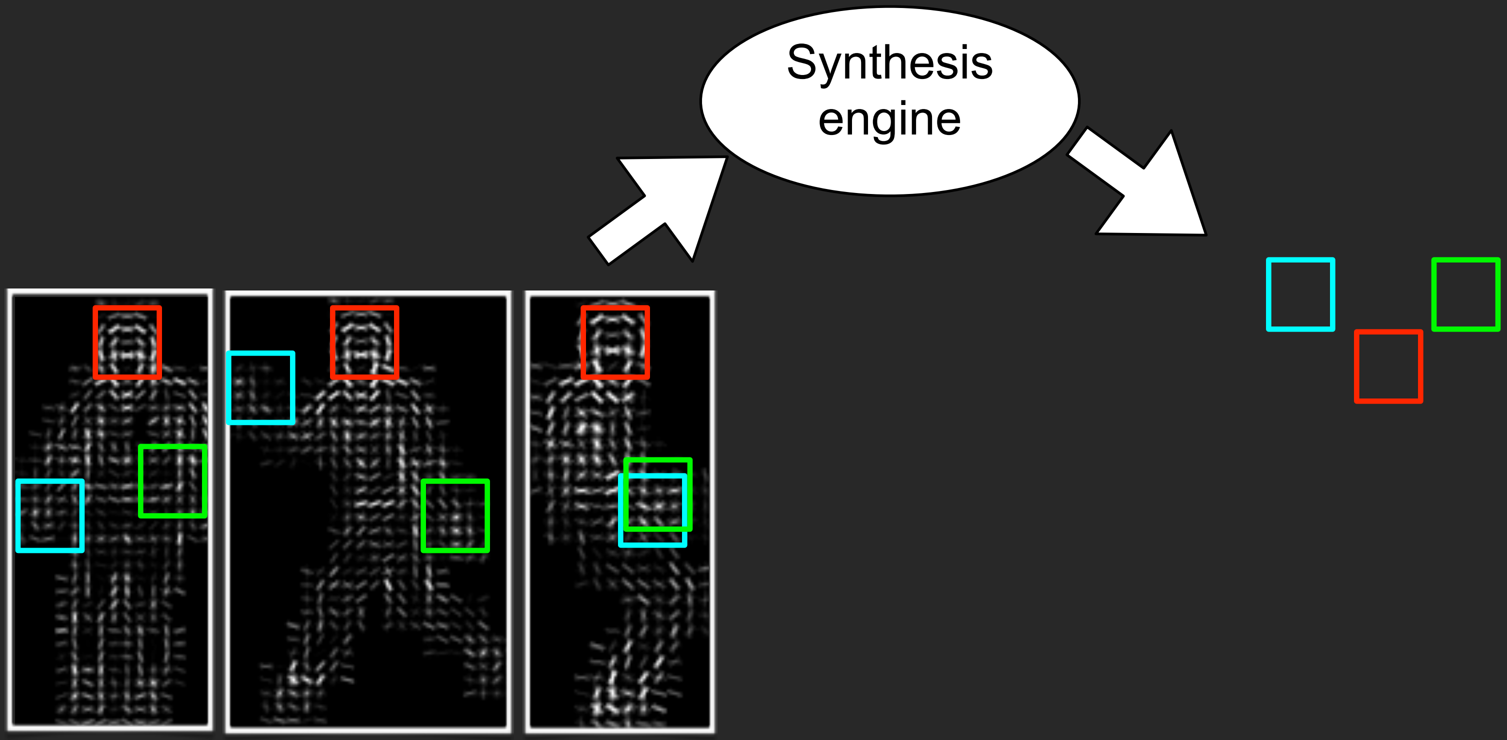
Sharing

Helps address "one-shot" learning (subcategory seen at least once)

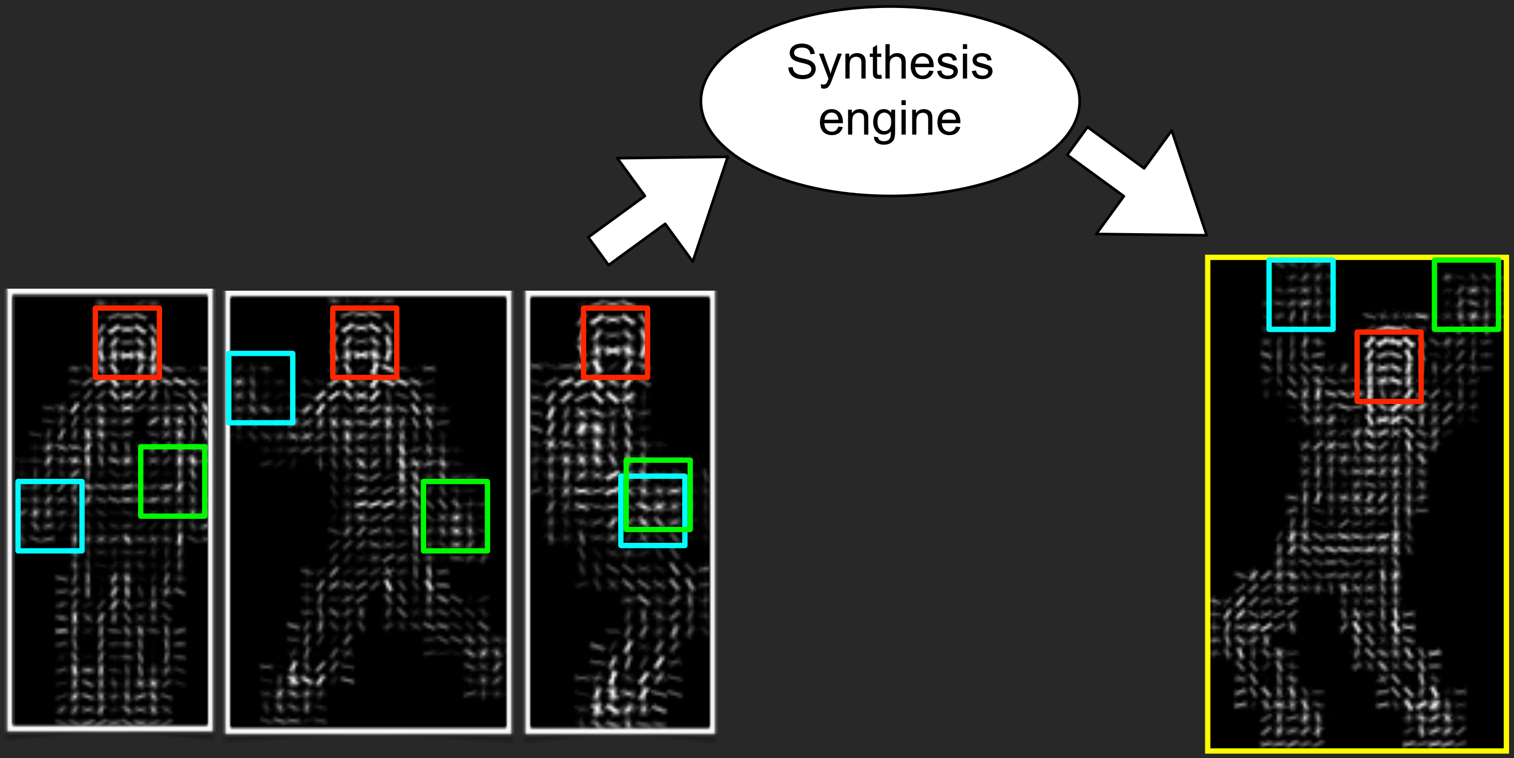


What about shapes/views that are never seen ("zero-shot" learning)?

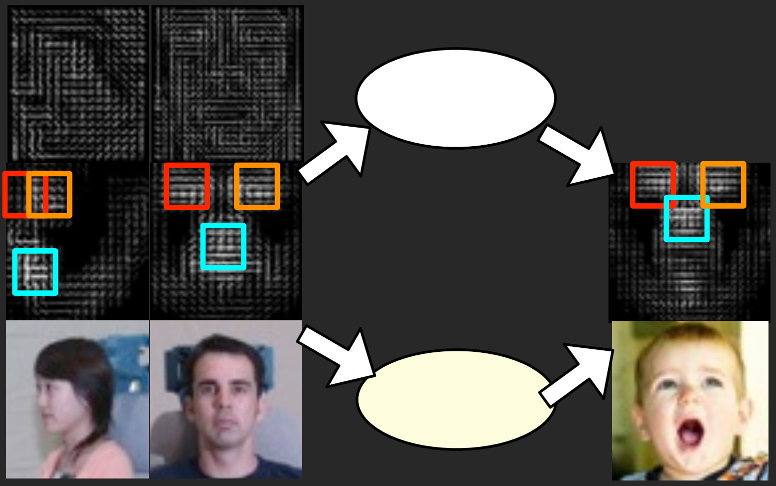
Shape synthesis

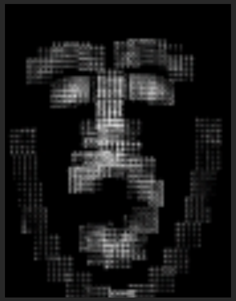


Shape synthesis



Sharing versus synthesis



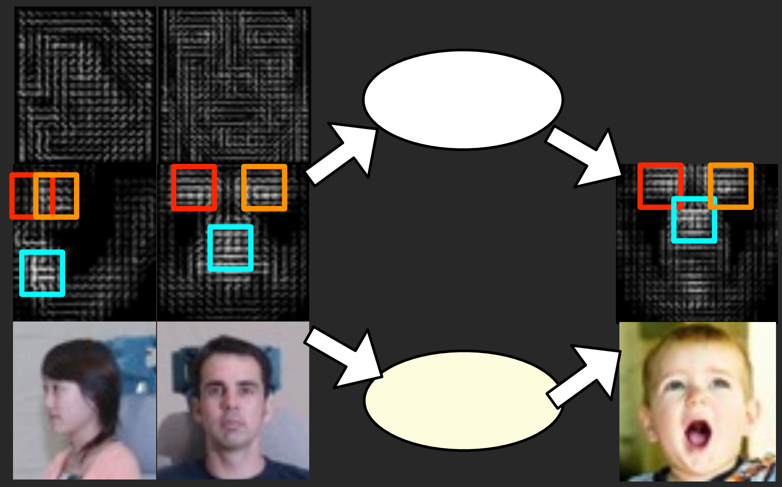


Part models perform implicit shape synthesis

Zhu et al, BMVC 2012

+ Don’t need to pre-synthesize
- Limited to simplistic shape models with efficient inference (stars, trees, springs,...)

Sharing versus synthesis



Part models perform implicit shape synthesis

+ Don’t need to pre-synthesize
- Limited to simple shapes with efficient computation (trees, springs,...)

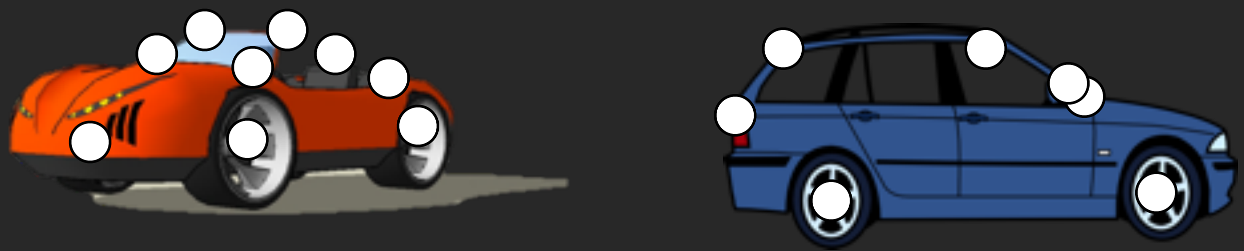
Instead, lets *explicitly* synthesize shapes with a graphics engine

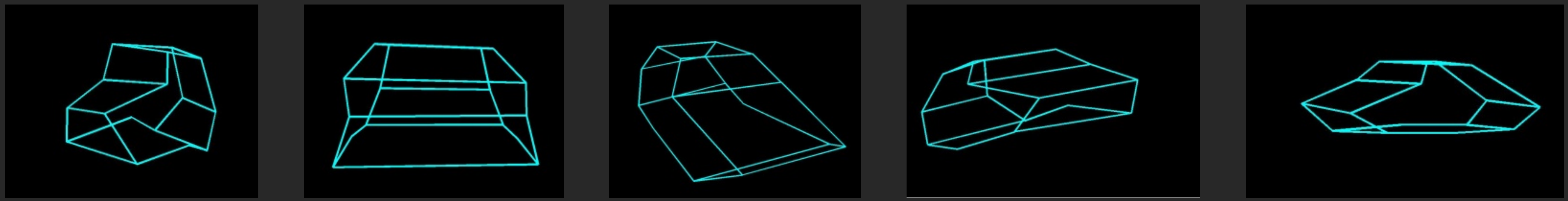
+ Can synthesize arbitrary shapes (e.g. 3D)
- Need to pre-synthesize millions of shapes

Aside: learning a 3D synthesis engine from 2D keypoints

- Stack all 2D landmarks into a large matrix; in noise-free case, it must be rank 3K (K=# of basis shapes)

- Learn shape basis with rank-based non-rigid SFM (Torresani et al CVPR01)



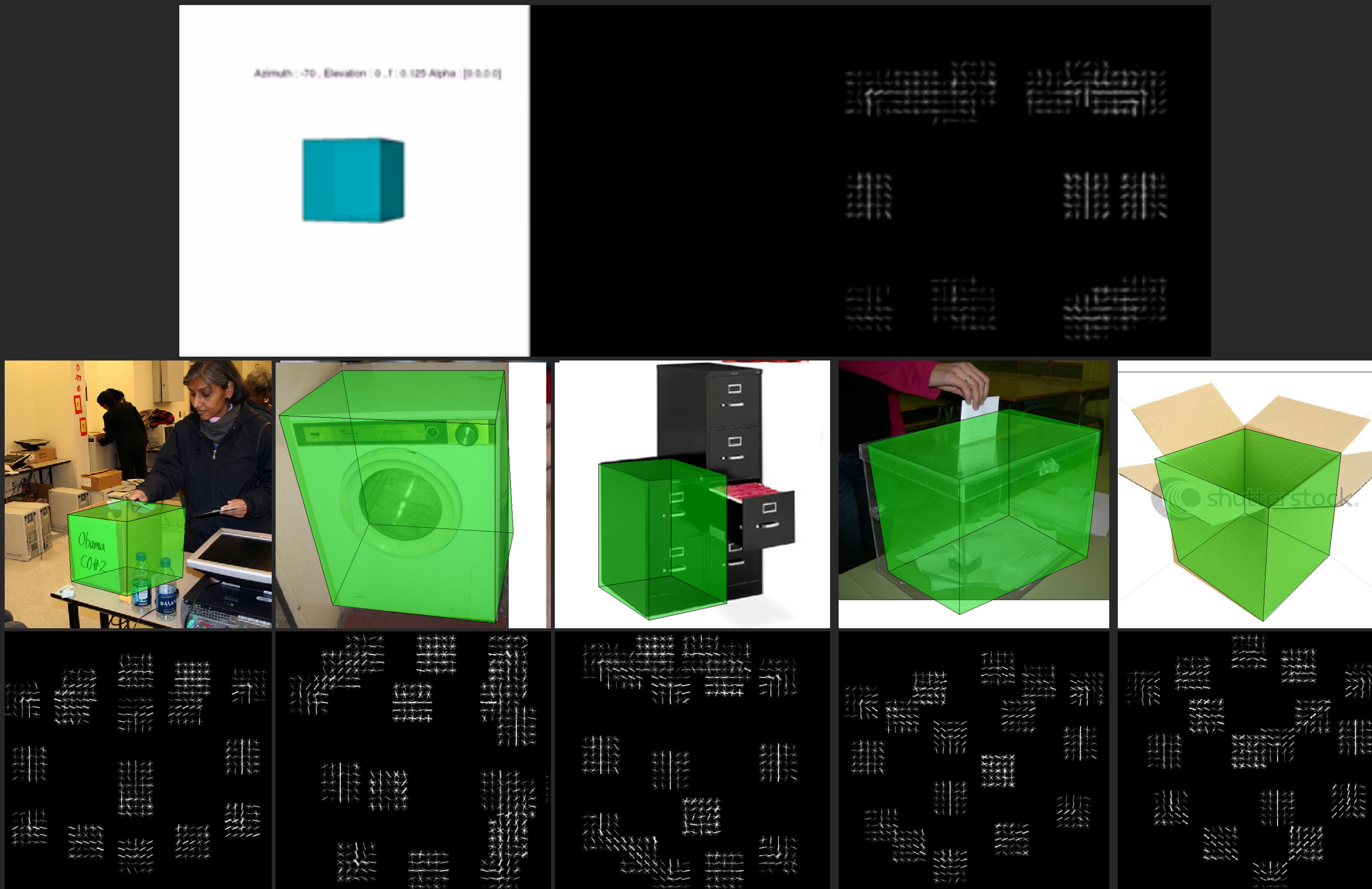


Hejrati & Ramanan, NIPS12

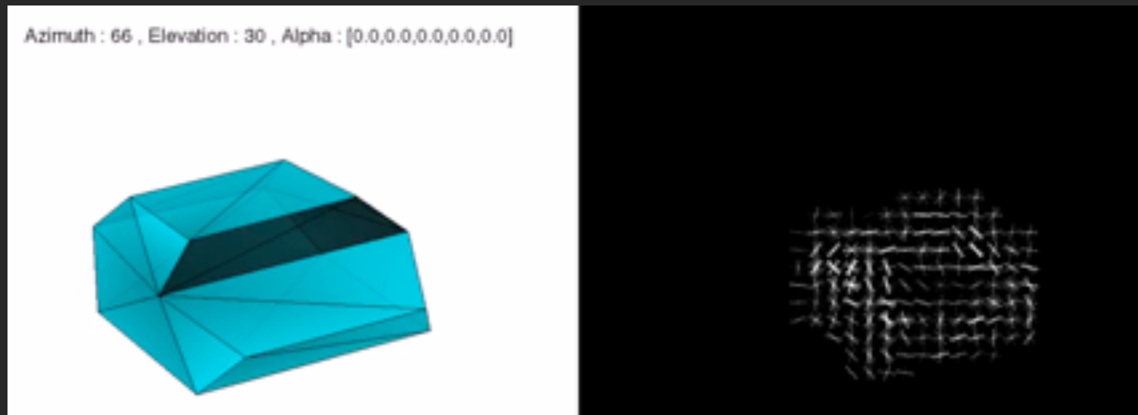
Explicit set of synthesized templates

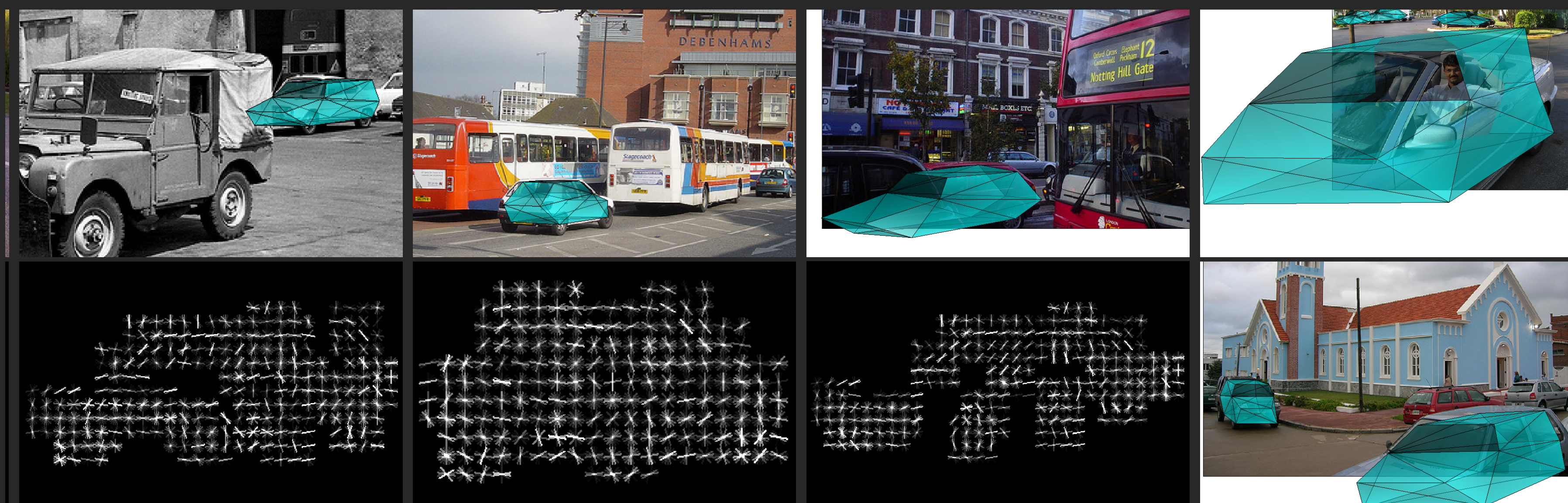
(Most shapes never seen during training)

Example detections

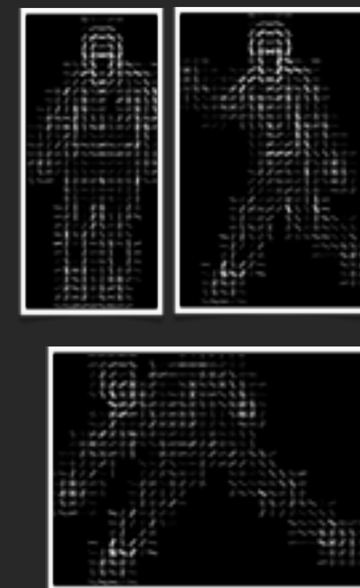


Car detection + reconstruction



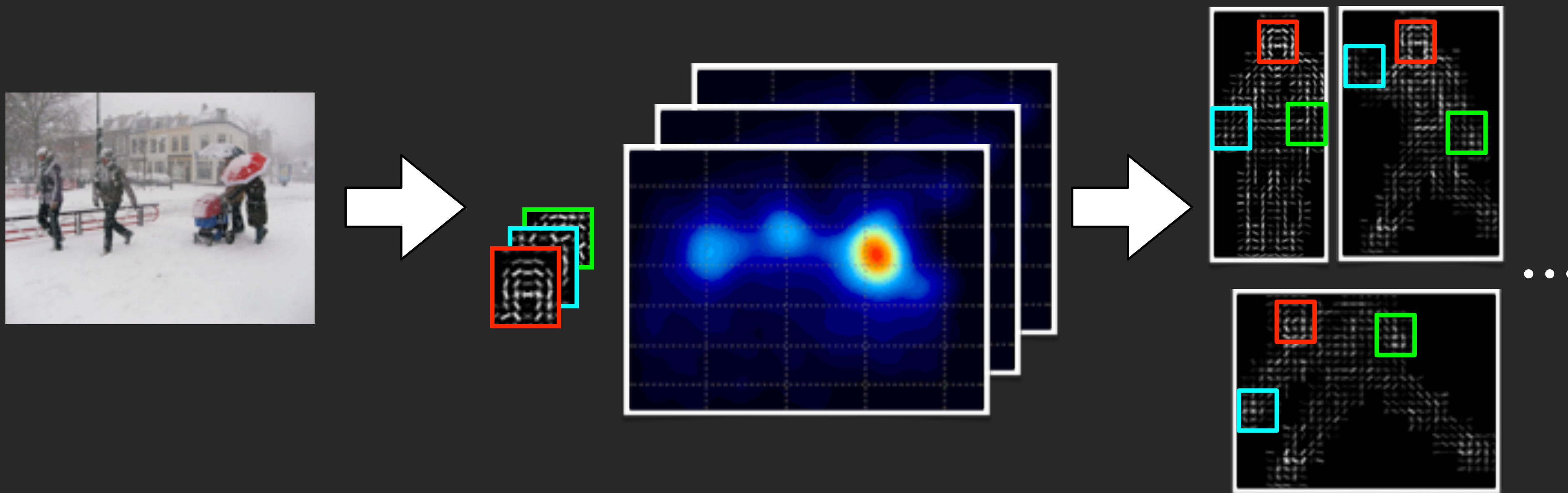


Inference



...

Inference



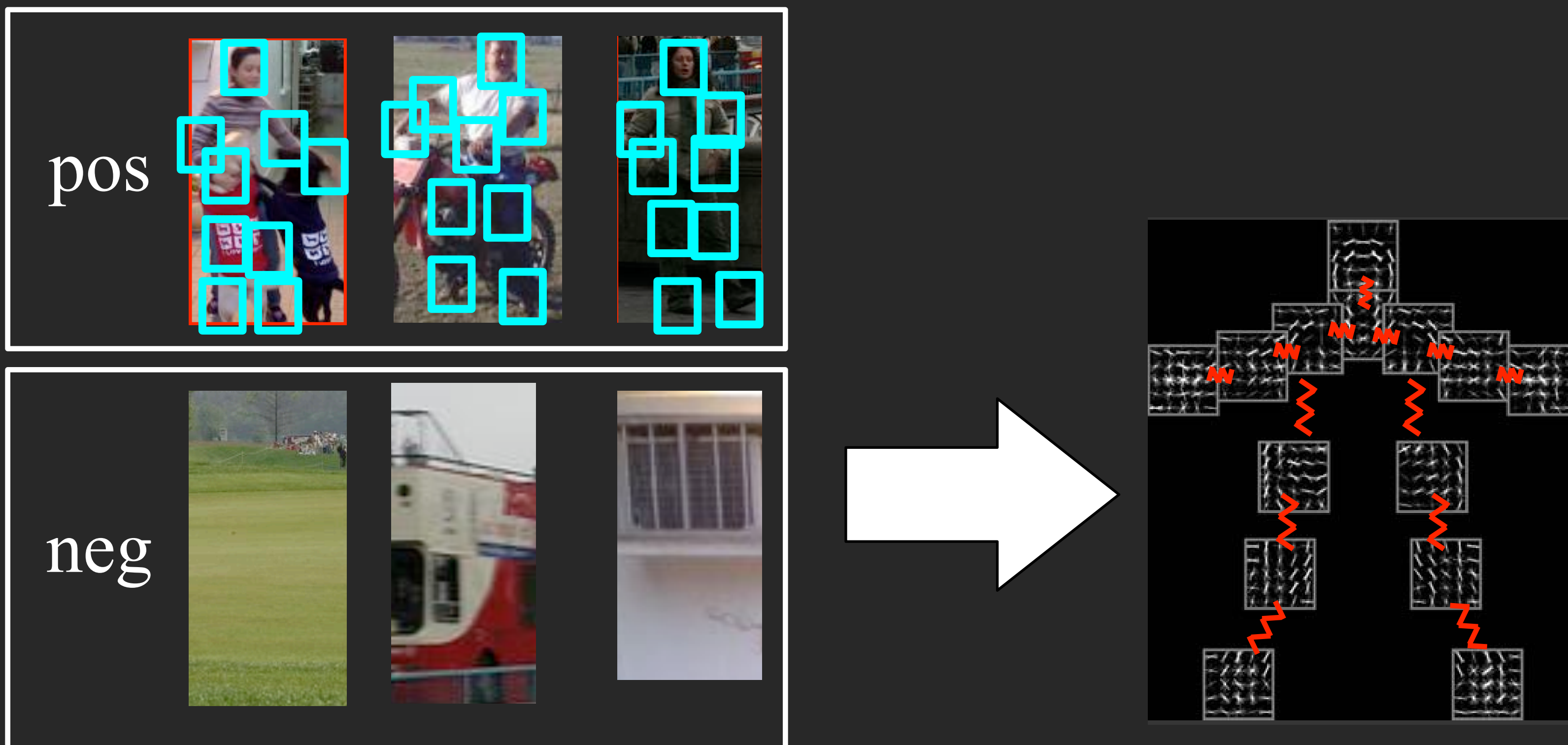
(1) Pre-compute tables of part responses

(2) Score each template with lookup table (LUT) queries

With efficient LUTs, (1) is bottleneck

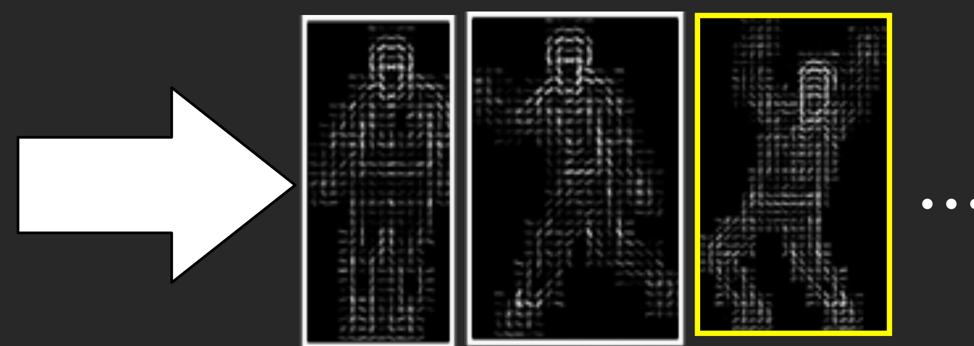
Supervised learning

$$S(I,\theta) = w \cdot \Phi(I,\theta), \quad \theta \in \Omega$$



Supervised learning

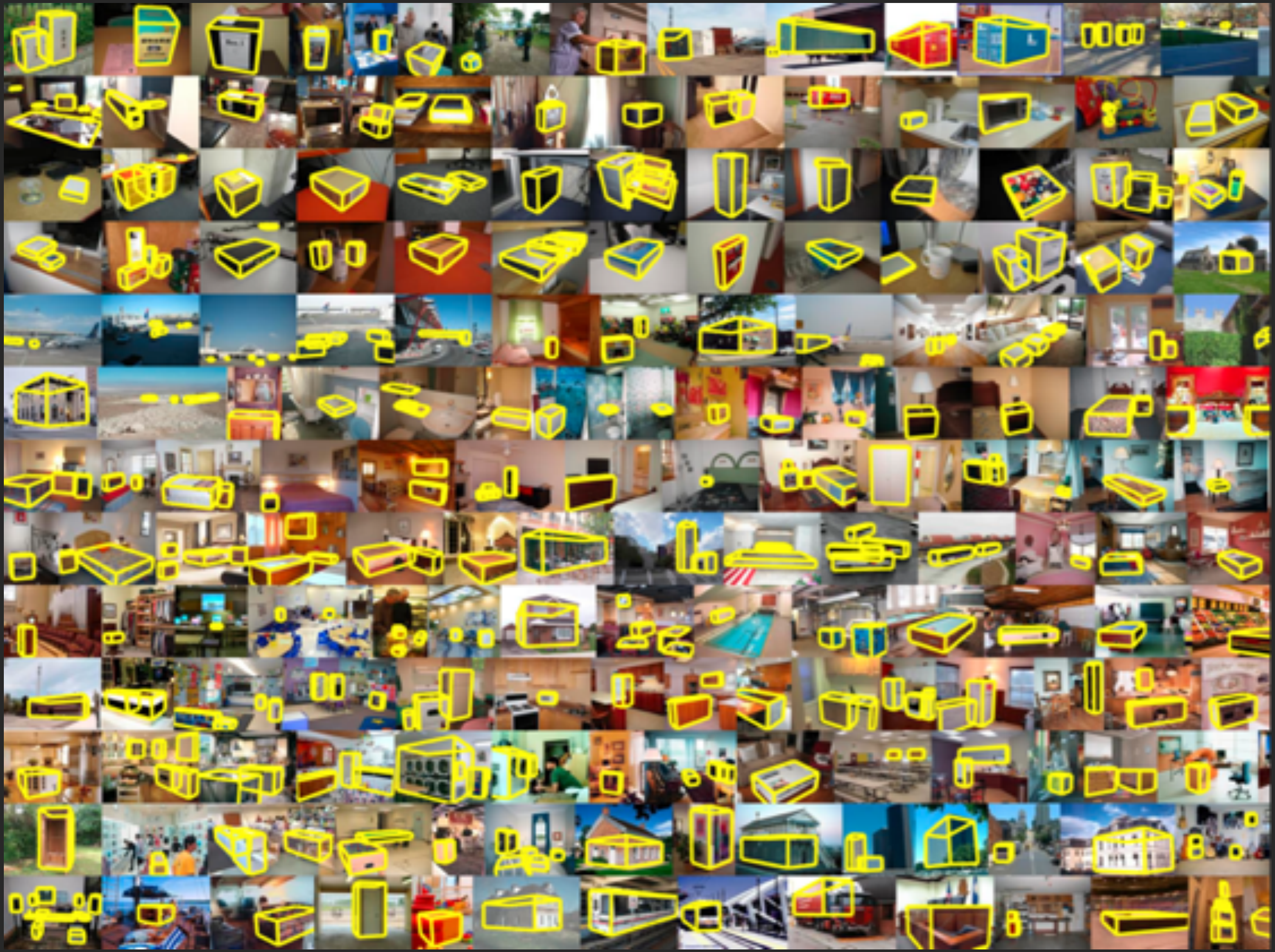
$$S(I,\theta) = w \cdot \Phi(I,\theta), \quad \theta \in \Omega$$



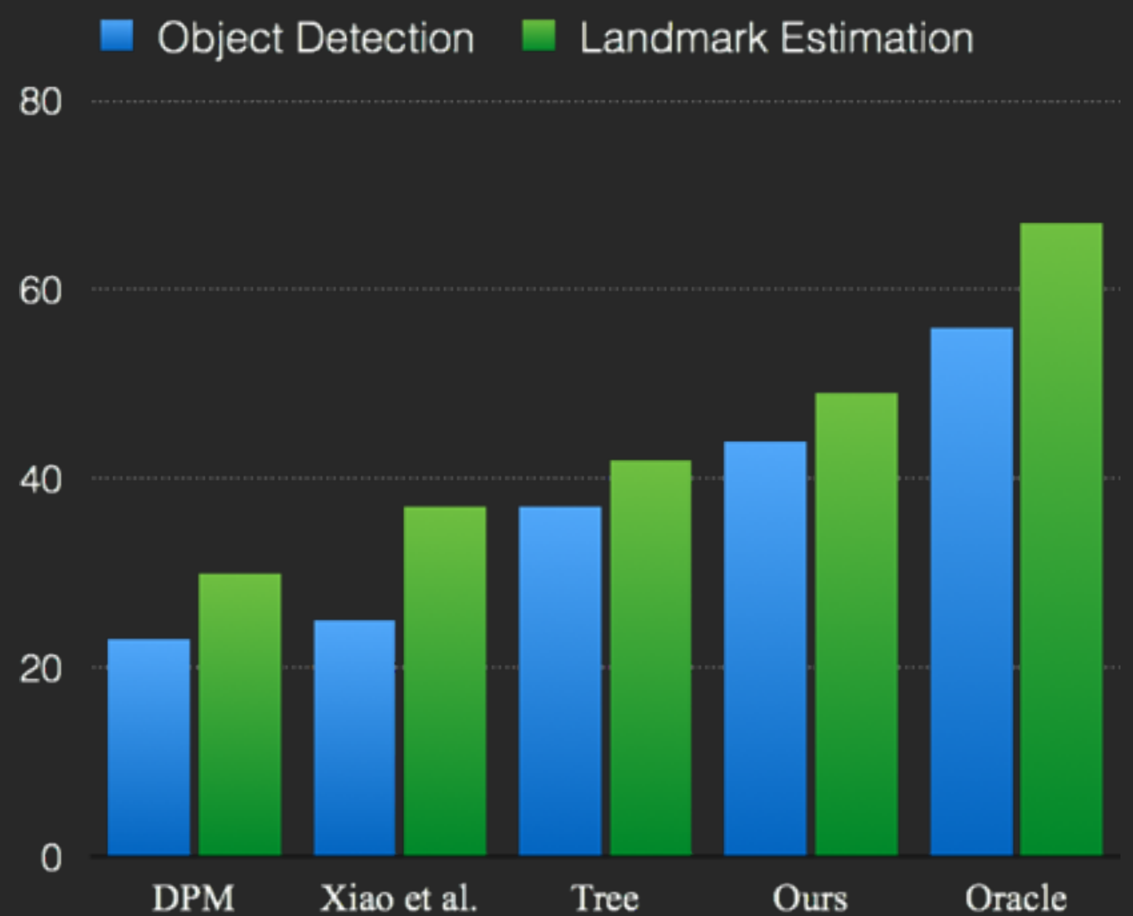
Learn classifiers for never-before-seen templates with synthesis

(Apply sparse learning tricks to deal with large set of negatives)

Evaluation - SUN Primitive dataset

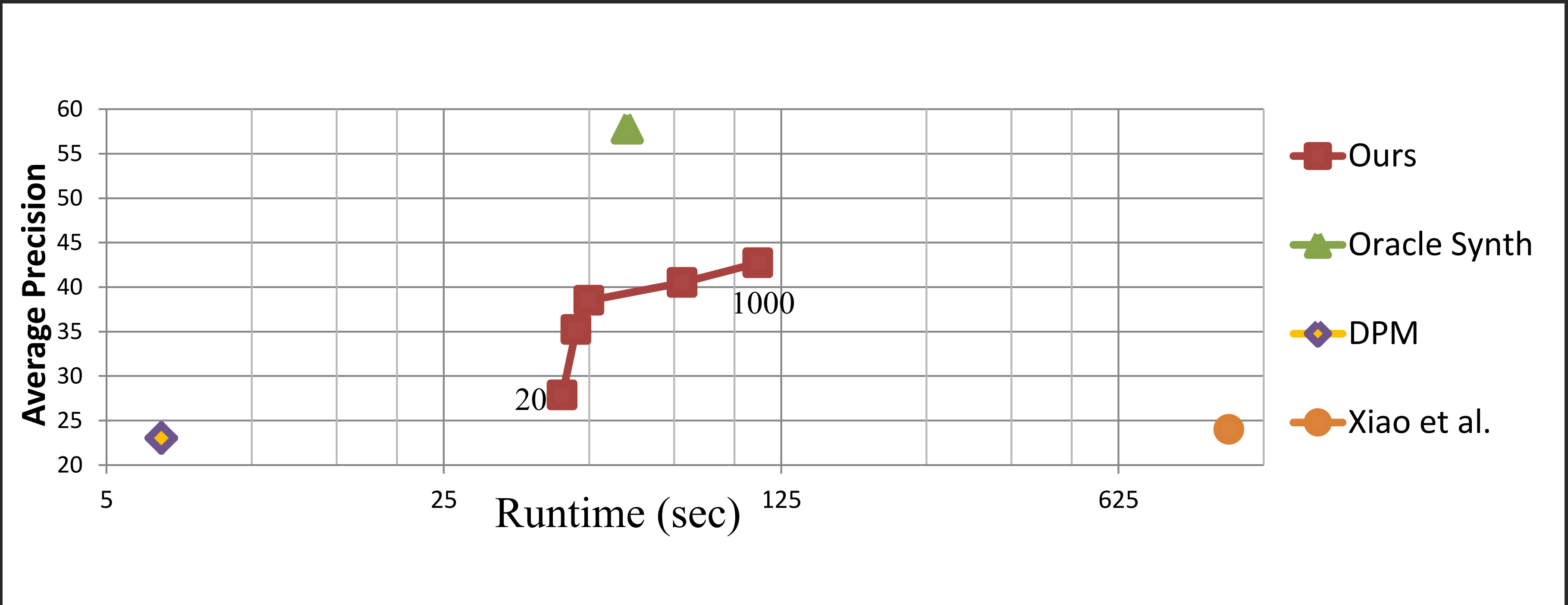


Quantitative performance



voc-re5 NIPS12

Quantitative performance

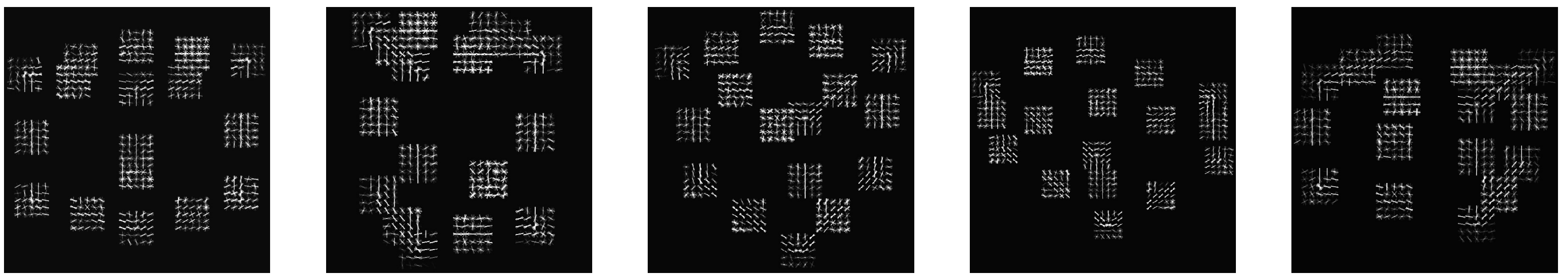


Tune Ω (set of quantized 3D parameters) to a fixed size by vector quantization

$$|\Omega| = \{20,50,100,500,1000\}$$

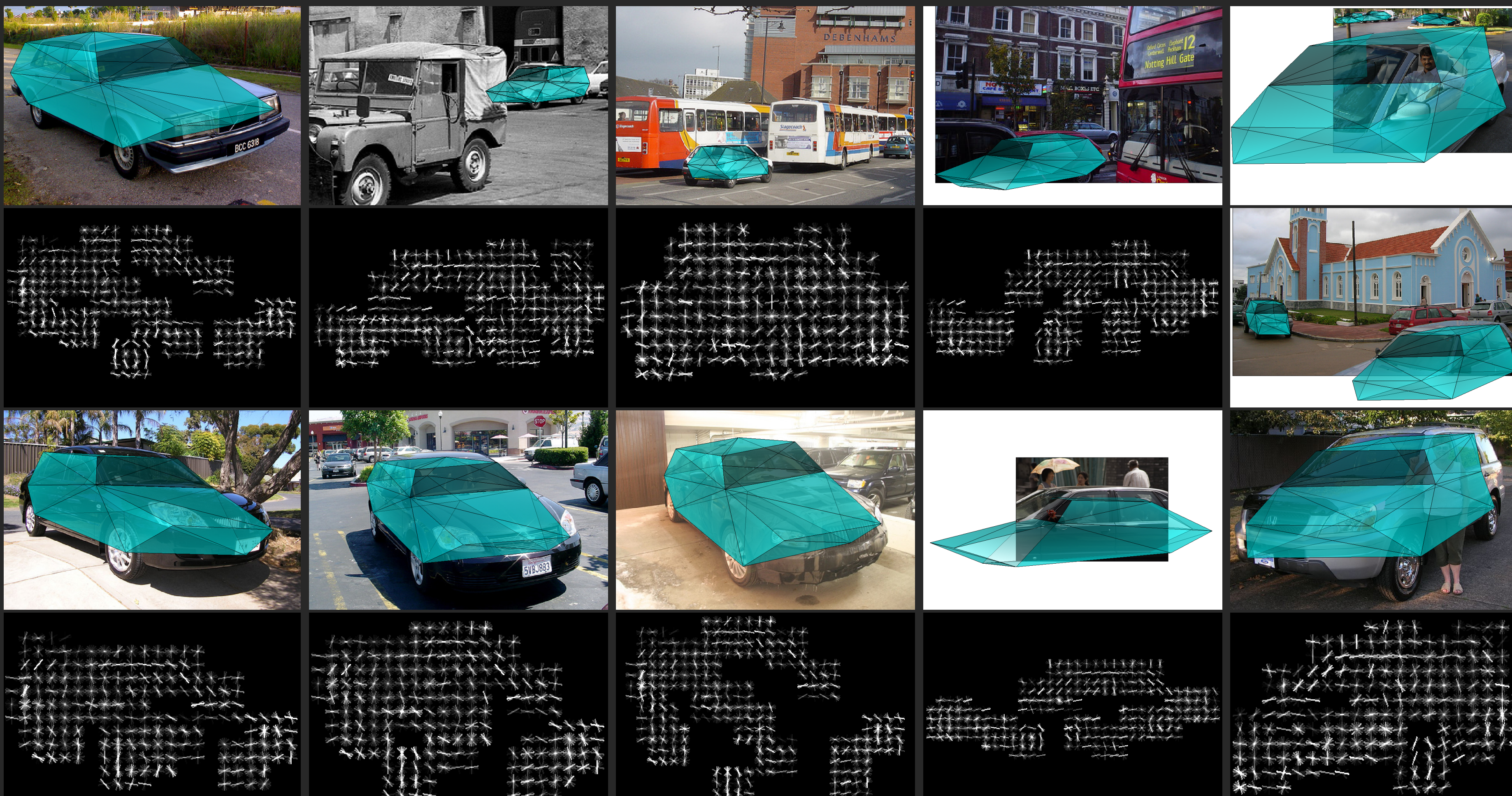
Anytime recognition + 3D reconstruction

Search through Ω in a coarse-to-fine fashion

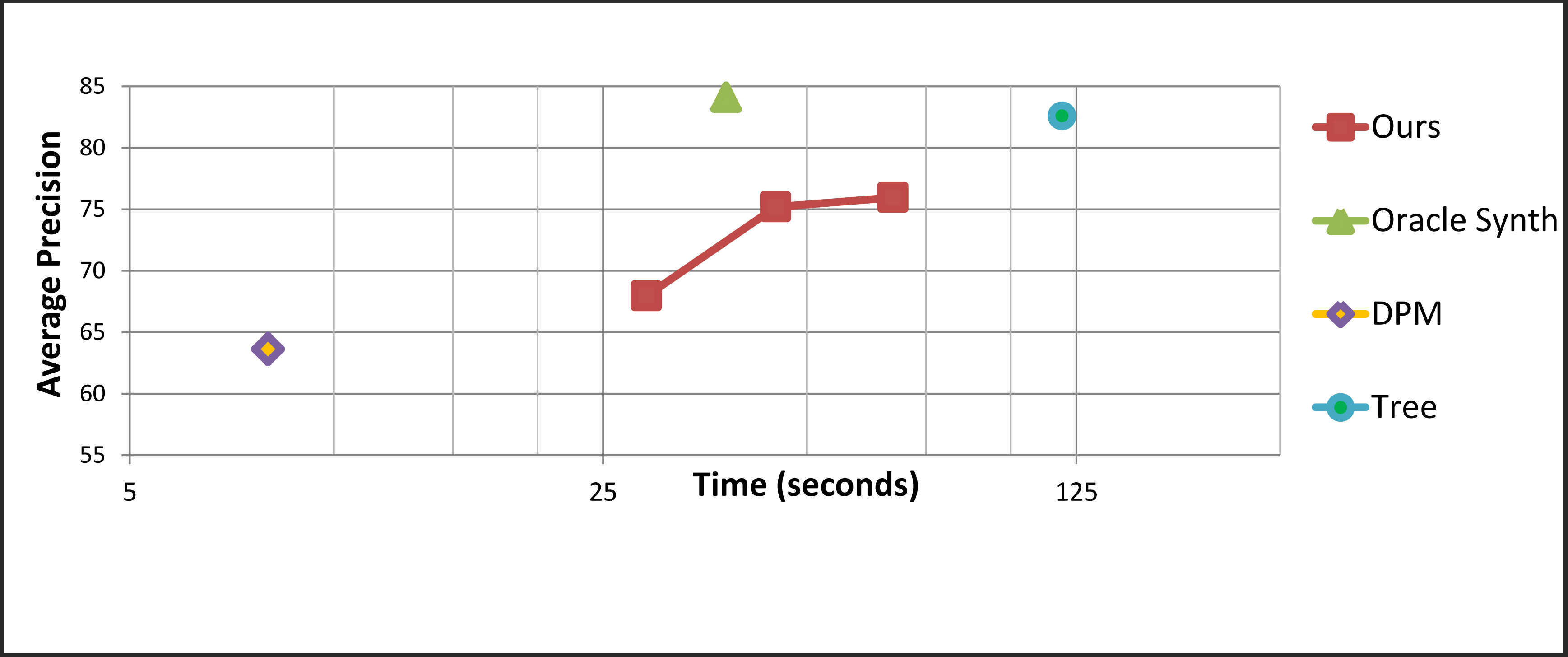


	θ_1	θ_2	θ_3	θ_4	θ_5	$\cdot$
Order	3	47	1	8	5	$\cdot$

Car recognition/reconstruction results



UCI Car



Average Precision
85
80
75
70
65
60
55
5
25
125
Time (seconds)
Ours
Oracle Synth
DPM
Tree

A look back

-Egocentric hand estimation

-Data analysis:
‘big’ temporal data

-Recognition as
3D reconstruction