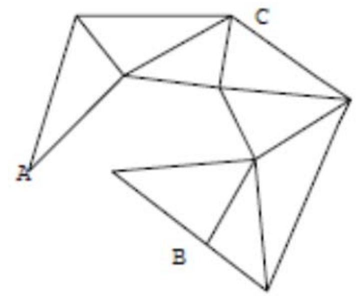
A* Talk

A.L. Yuille (UCLA)

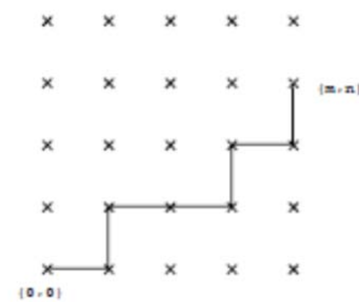
Three Examples of A*

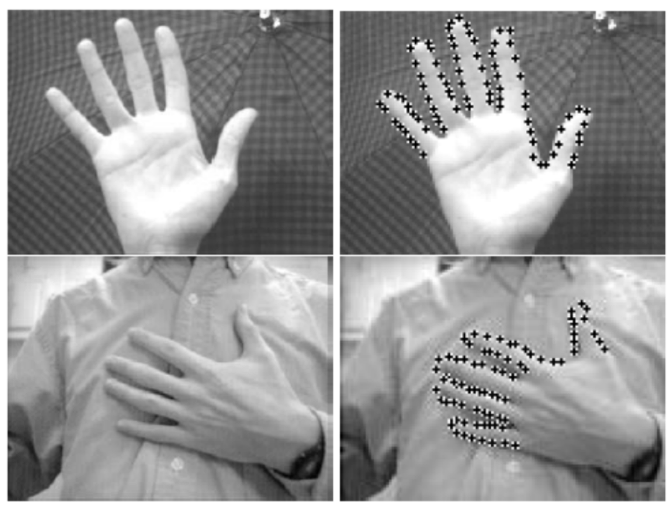


- Example 1: Interactive Segmentation.
- Example 2: Road Tracking – Geman and Jedynak 1996.
- Example 3: A* for Hierarchical Object Models (Kokkinos).

Example 1: Interactive Segmentation.

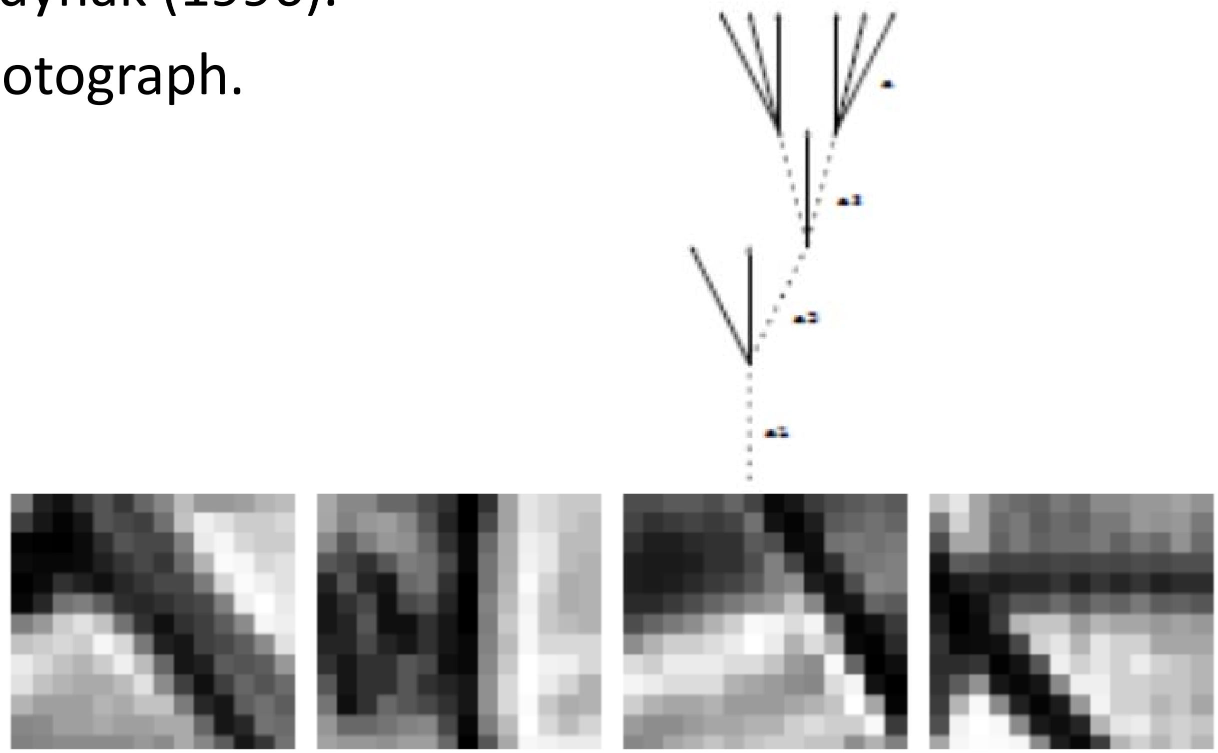
- Graph is the Image lattice.
- Two points on the graph are specified – A &B.
- Find shortest path between A& B.
- Now go to hand-written notes.





Example 2: Road Tracking

- Inspired by Geman and Jedynak (1996).
- Find a road in an aerial photograph.



Road Tracking

- Example:

How to search for the road?

- There are an exponential number of possible paths.
- You do not have time to search them all.
- You must select a search strategy that is efficient.
- Geman and Jedynak proposed a new strategy based on information theory.
- Their strategy is to search so as to maximize the expected gain in information – see Jedynak's talks on Wednesday.
- Coughlan and Yulle analyzed the algorithm – and showed that it was a variant of an inadmissible A* algorithm.

Third Example: Detecting Object in Image

- Hierarchical Models of Objects.
- A* over rules for combining subparts of objects to build a complete object.
- This is used to detect objects – cars – in images.

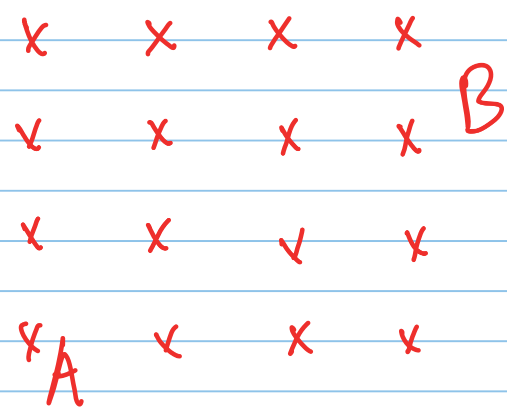
- Why hierarchical models? (more in week 3).
- More robust than “flat models” – can detect even if subparts of the object are missing (occluded or undetected).
- Ability to share parts (not used in this example).

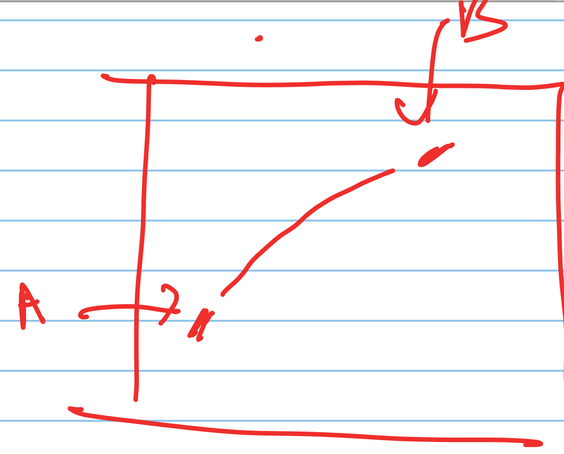
(i) # Example 1 : Interactive Segmentation

Find best curve between A & B.

Note Title 7/29/2013

Image lattice





Set X_A at A
X_B at B

Path $\underline{X}$: $X_A, X_1, \ldots, X_N, X_B$
where X_i, X_{i+1} are neighbors on the graph.

Best Curve
- Smoothest (geometry)
- edginess – evidence for edges on curve.

C ($X_A X_1, X_N X_B$)

(ii) <u>Energy Cost.</u> $E(\underline{X}) = \sum_{i=1}^{N} \phi_i(x_i) + \sum_{i=1}^{N-1} \phi_i(x_{i+1}, x_i)$

$+ \phi_A(x_A, x_1) + \phi_B(x_N, x_B)$.

Unary + Binary terms
(e.g. see Boykov,,)

- Binary term imposes geometry — e.g. the curve is smooth.
- Unary term imposes image term — e.g. the curve goes through pixels where there is evidence for an edge (e.g. $|\nabla I|$ is big)

<u>Task</u>: Solve $\hat{\underline{X}} = \arg\min_{\underline{X}} E[\underline{X}]$

We cannot search exhaustively over all curves — too many. (exponential)

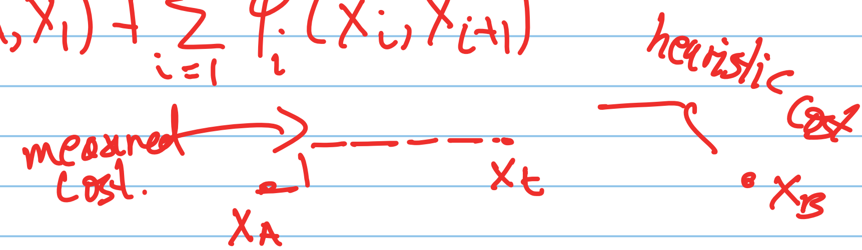
(iii)

A^* Algorithm.

Partial path $\underset{\sim}{X}^t = (X_A, X_1, \ldots X_t)$

has cost. $\sum_{i=1}^{t} \varphi_i(X_i) + \varphi_A(X_A, X_1) + \sum_{i=1}^{t-1} \varphi_i(X_i, X_{i+1})$

plus heuristic cost $f(X_t, X_B)$.



Apply A^*.

What heuristic cost? Depends on form of φ's.

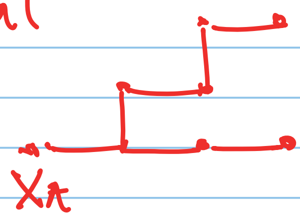
Admissible Heuristic: An underestimate of cost to get from X_t to X_B

$$f(X_t, X_B) \le \min_{\text{all path}} \left\{ \sum_{t+1}^{T} \varphi_i(X_i, X_{i+1}) + \varphi_T(X_T, X_B) + \sum_{t+1}^{T} \varphi_i(X_i) \right\}$$

(iv) In some cases, we can set $f(x_t, x_B) = 0$
$\forall \; \varphi \geq 0.$

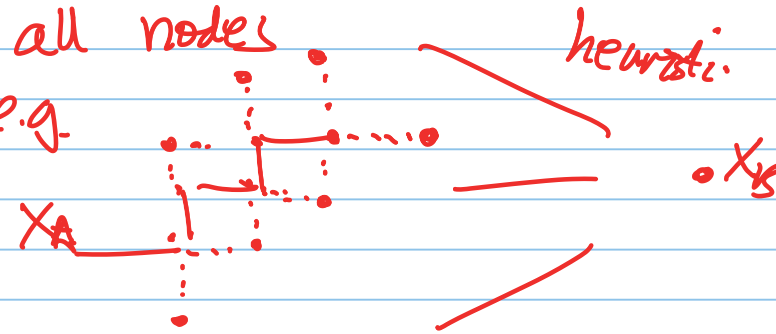
E.g., Dijkstra's Algorithm. // (Geiger & Liu, 1996)

Set of partial curves at time t.



can expand all nodes

e.g.



Which node to expand?

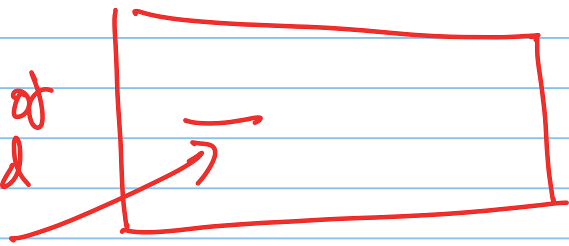
Node which minimizes Cost of Partial Curve + Cost of Heuristic.

Guaranteed to converge to minimum cost path (if heuristic is admissible)

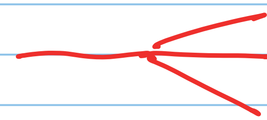
Note Title 7/29/2013

(1) **Example 2** : Find a road (highway) from Aerial Image
(Geman & Jedynak 1996)

Start of Road is given.



Model: road consists of segments.
(e.g. length 8 pixels)



The next segment is either: (i) left (at 15°)
(ii) straight
(iii) right (at 15°)

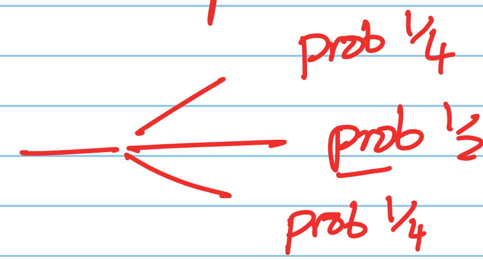
A road path is a sequence of N segments

e.g.  or  or . . .

There are an exponential no. of possible road paths
→ 3^N road paths

(2) Probability distribution over road paths.

or



representation $X = (x_1, \ldots x_N)$

→ position of first segment (specified)

$P(X_{t+1} \mid X_t)$ → e.g. if straight $P(\cdot \mid \cdot) = 1/2$

left $P(\cdot \mid \cdot) = 1/4$

right $P(\cdot \mid \cdot) = 1/4$

$$P(X) = \prod_{t=1}^{N-1} P(X_{t+1} \mid X_t)$$ Markov Property

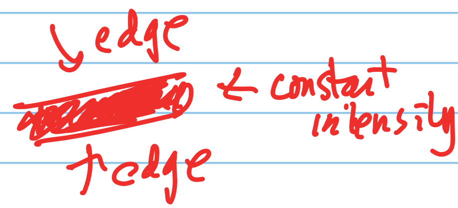
prob for the complete road pat

(3) Image Model.

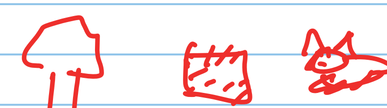
Set of all possible segments $a \in A$.

Test Y_a — image filter applied to segment a

E.g.

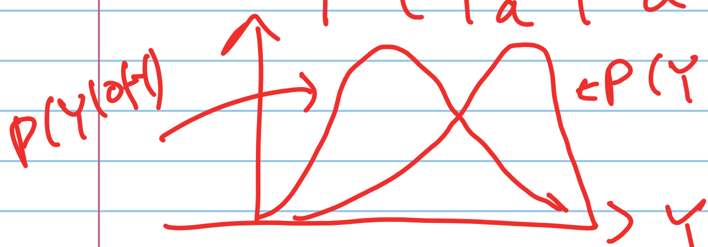


or anything else

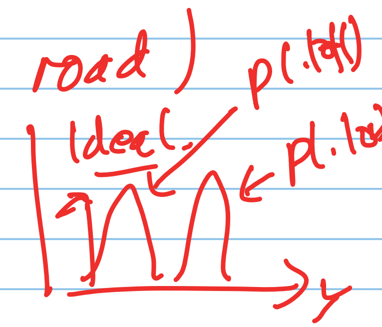


details of filter (unimportant)

$P(Y_a \mid a \text{ on road})$ $P(Y_a \mid a \text{ off road})$



Local cues are ambiguous
distribution overlap.



(4) Bayes Formulation.

$$P(Y|X) = \prod_{a \in A} P(Y_a | X_a)$$

$$P(X) = \prod P(X_{t+1} | X_t)$$

Assumes that the image filter responses are independent.

Want to maximize

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)} \propto P(Y|X)P(X)$$

($\propto$: proportional to)

Solve:

$$X^* = \text{ARG} \max_X P(X|Y)$$

(5) Cannot solve $\hat{X} = \underset{X}{\text{ARG MAX}}\ P(Y|X)P(X)$ by exhaustive search.

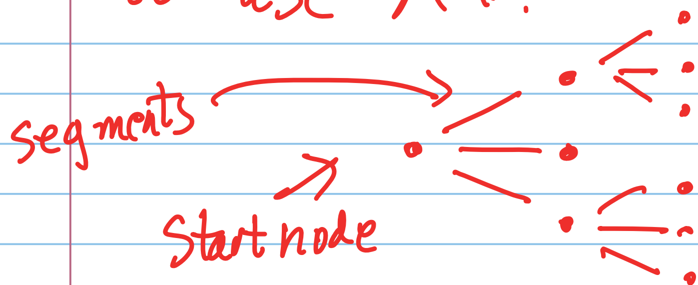
There are too many possibilities → 3^N road paths.

Can't evaluate all these paths, can't even store them.

$$\hat{X} = \underset{X}{\text{ARG MAX}} \{\log P(Y|X) + \log P(X)\}$$

or $$\hat{X} = \underset{X}{\text{ARG MIN}} \{-\log P(Y|X) - \log P(X)\}$$

(6) The form of $P(X)$ and $P(Y|X)$ enables us to use A*.



Note: this is like Prof. Korf's road search problem with GPS

End node

Note: In G&J, there is no end node. So we pretend that we have one.

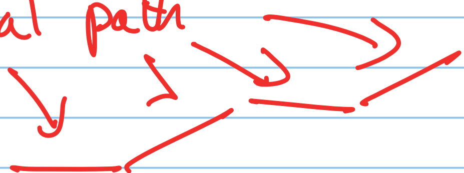
What is cost for each segment?

Two terms: (i) output of image filter. $-\log \frac{P(Y_a \mid a \text{ on road})}{P(Y_a \mid a \text{ off road})}$

(ii) geometry (depends on previous node) $-\log P(X^{t+1} \mid X^t)$ X^{t+1} is segment a.

(7) To apply A* we need a heuristic

partial path



Score $-\sum_{t=1}^{T} \log P(X_{t+1} \mid X_t) - \sum_{a=1}^{T} \log \frac{P(Y_a \mid a \text{ on road})}{P(Y_a \mid a \text{ off road})}$

Heuristic → lower bound of the rest of the path.

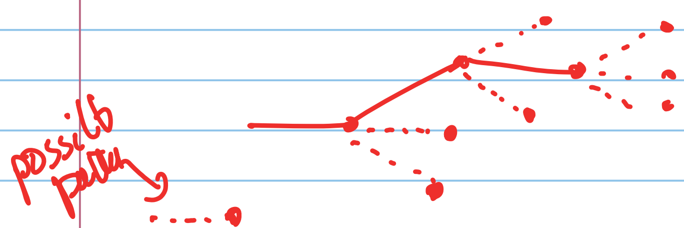
Need to lower bound $\min \left\{ -\sum_{t=T+1}^{n} \log (X_{t+1} \mid X_t) - \sum_{a=T+1}^{n} \log \frac{P(Y_a \mid a \text{ on road})}{P(Y_a \mid a \text{ off road})} \right.$

(8) **Special Case.**

One idea: let the heuristic be $(N-M)c$, where N is total length of the road, M is the no. of segments we have travelled so far, and c is a constant. ← cost so far.

Our cost is

$$-\sum_{t=1}^{M}\left\{\log P(X^{t+1}|X^{t}) + \log P(Y^{t}|X^{t})\right\} + (N-M)c \quad \leftarrow \text{heuristic cost}$$



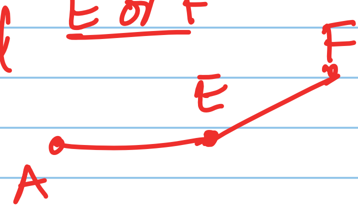
$$= -\sum_{t=1}^{M}\left\{\log P(X^{t+1}|X^{t}) + \log(Y^{t}|X^{t}) + c\right\} + Nc \quad \leftarrow$$

Note: independent of path and of M.

(9) This choice of heuristic means that we do not need to have an end node. → i.e. drop the NC term.
we can continue for ever.

Intuition → expand a road path by adding a segment
compute its extra cost $-\log P(X^{M+1} | X^M) - \log (Y^n | X^n)$
subtract c

Expand E or F



Path AEF is longer than path AE (by one segment).

But subtracting c penalizes the length.

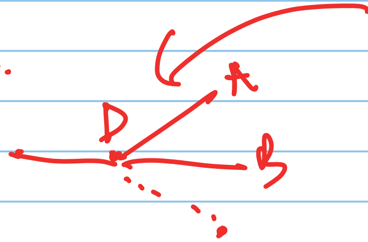
Hence

(10) What choice of heuristic?

If admissible, then $-C \leq \min_{X^{t+1}, X^t} \{ -\log P(X^{t+1}|X^t) - \log P(Y^t|X^t) \}$,

But, this is bad for this example.

Why, because it means we are not satisfied with any segment.



Expand this segment.

measure $-\log P(X^{t+1}|X^t) - \log P(Y^t|X^t)$

add heuristic $-C$.

get result ≤ 0

So do not expand A further. Instead expand B, (same result)

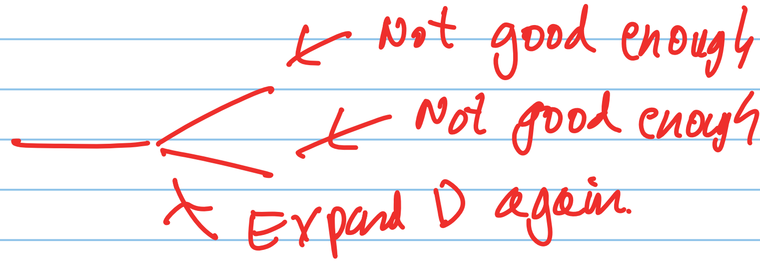
then expand D.

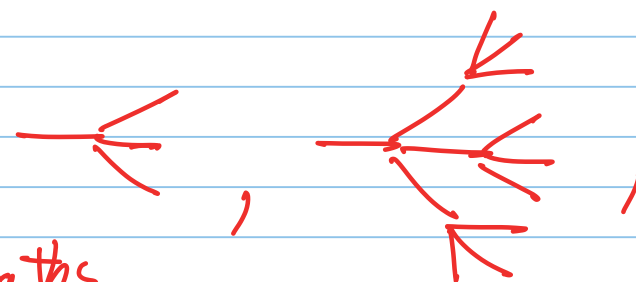
(11) Problem with this heuristic → reduces to breadth-first search.

Intuition → we are never satisfied with the local measurement -

$-\log P(x_{t+1}|x_t) - \log P(y_{t+1}|y_t)$ because it is worse than c.

− So we go back and expand an earlier path.





This requires searching 3^N paths.

Exponential — impossible

(12) This is an unusual case → Not standard A*

What to do?

- Inadmissible Heuristic.

Cannot guarantee convergence to best solution

But can make statistical guarantees (Coughlan and Yuille)

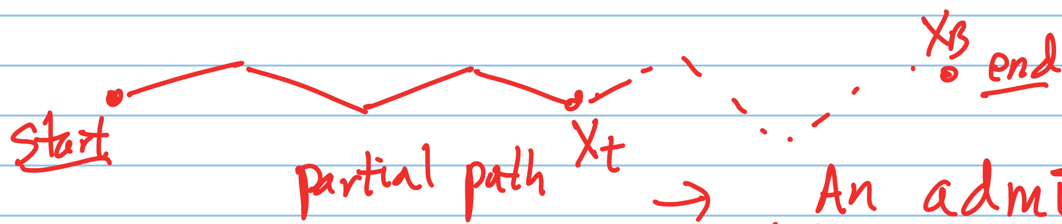
- With high probability we can find a path which is close to the true path in O(N) time.

Like Probably Approximately Correct (PAC) theorems in Machine Learning (Vapnik, Valiant, McAllester)

(13) <u>How to do this?</u> <u>Brief Sketch.</u>

Problem Formulation assumes probability distributions $P(X)$ and $P(Y|X)$.

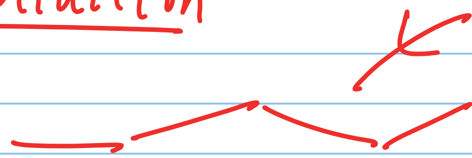
Can use these distributions to analyze the algorithms.



→ An admissible heuristic gives a lower bound of cost from X_t to X_B.

<u>Instead</u>, specify a heuristic cost for each segment (not a lower bound) and compute the probability that the algorithm wastes time exploring false paths.

(14) Some Intuition.



← cost of segment is a random variable ξ.

There is a probability $P(\xi \mid \text{on road})$

probability $P(\xi \mid \text{off road})$

specified by $P(X), P(Y|X)$

We have a score $\varphi(\xi)$ for each segment

(specified by $-\log \frac{P(\cdot|on)}{P(\cdot|off)} - \log P(\,|\,)$).

Admissible Heuristic is $\leq (N-M) \min_{\xi} \varphi(\xi)$

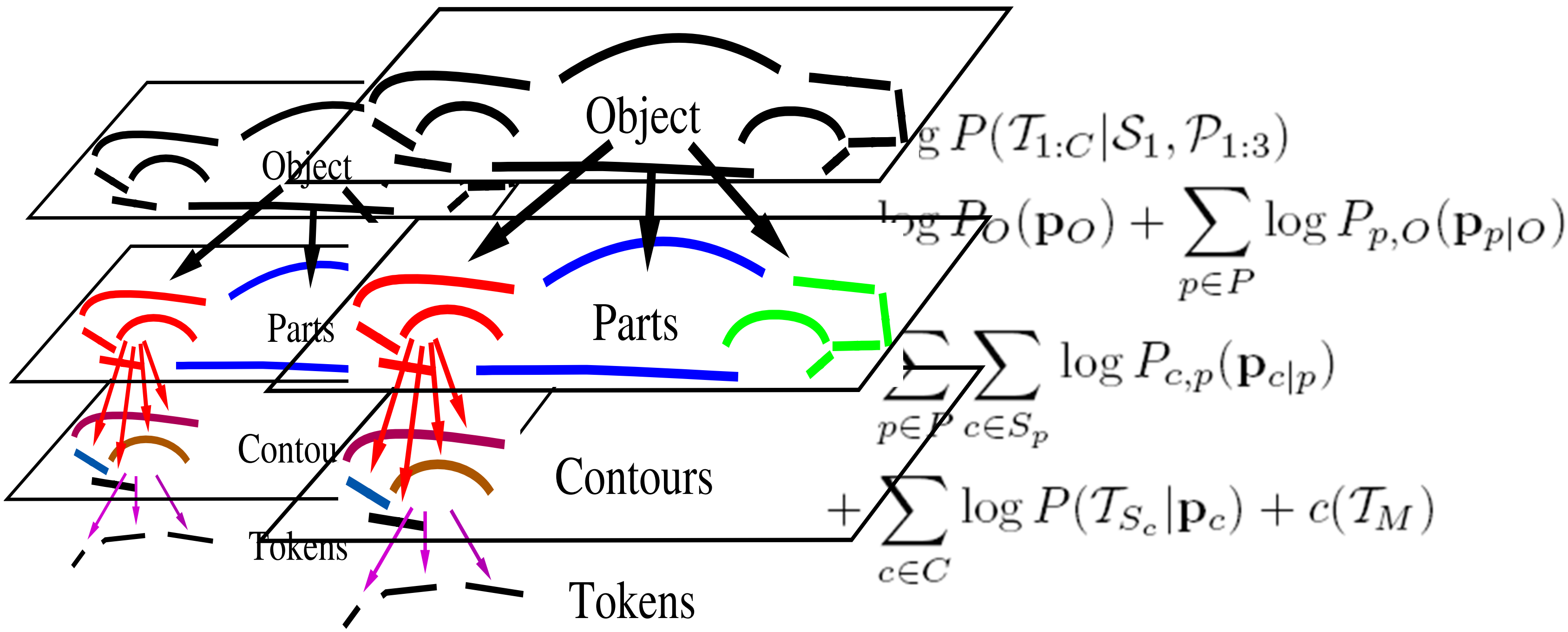
But it is very unlikely that we will have this cost

If $N-M$ is large, law of large numbers says that it will be closer to expected value $(N-M)\langle \varphi(\xi) \rangle$.

Example 3

I. Kokkinos

Hierarchical Compositional Models



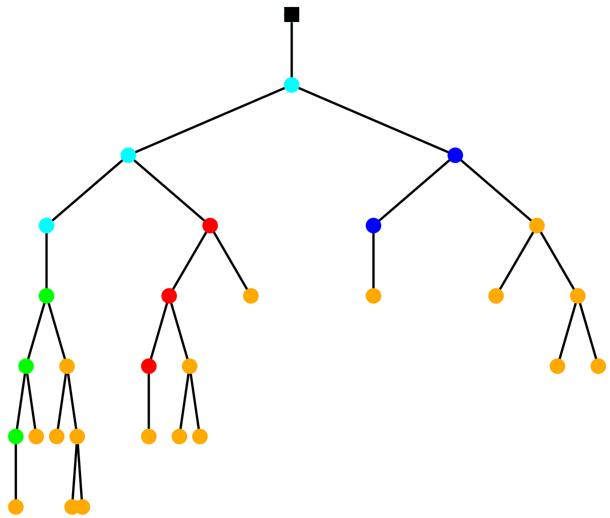
- Top-down view: object generates tokens
- Bottom-up view: object is composed from tokens

Compositional Detection

- View production rules as composition rules

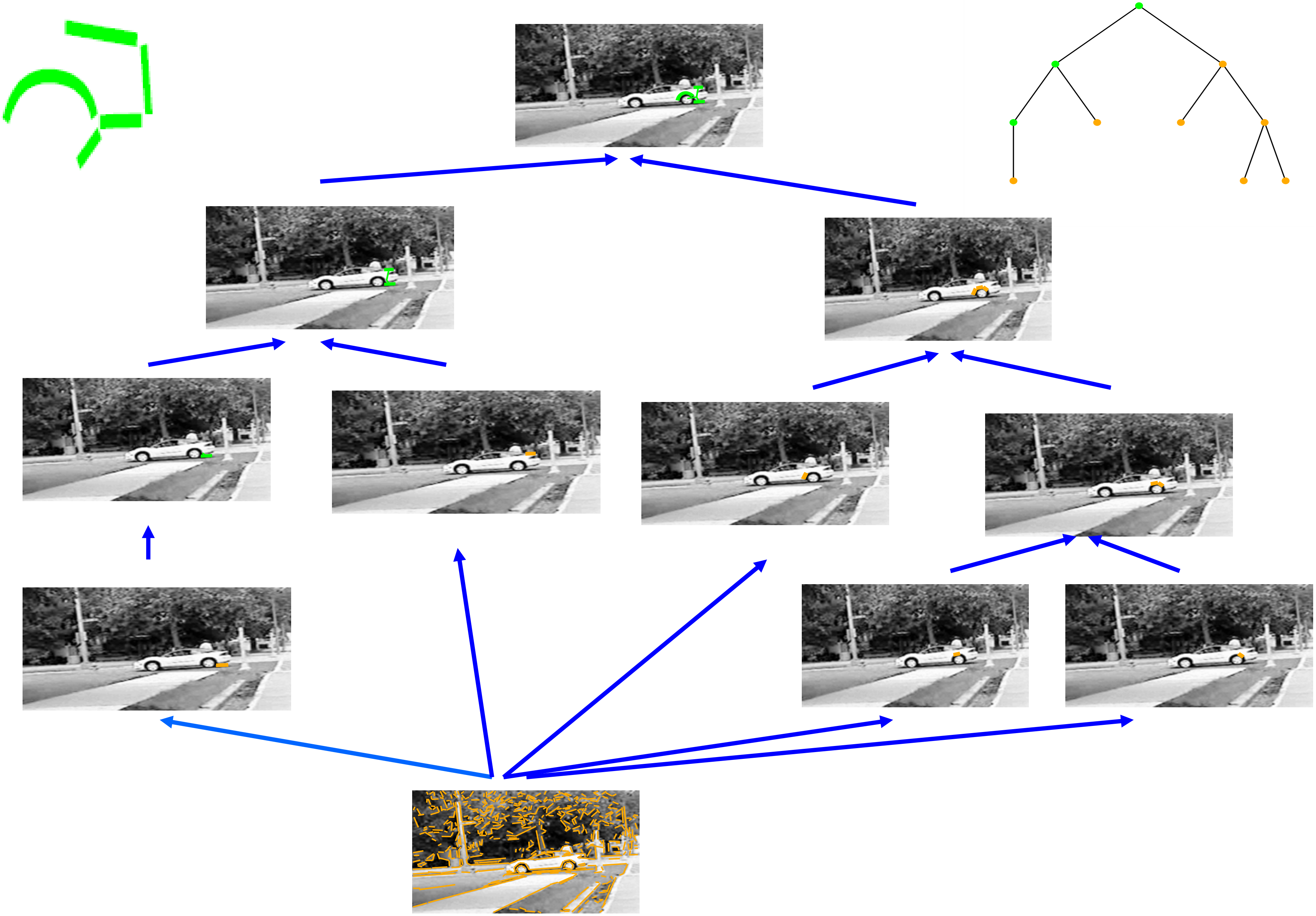
$$(\mathbf{p}_{p_1}, \ldots, \mathbf{p}_{p_n}) \rightarrow \mathbf{p}_O$$

- Build a parse tree for the object



- Requires
  - Composition rules
  - Prioritized search

Composition of the `Back' Structure



Composition as Climbing a Lattice

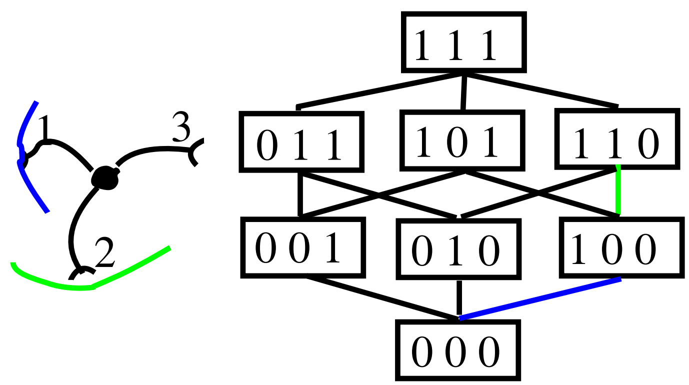
- Introduce vector indicating instantiated substructures

$$I(S) \quad = \quad [1,0,1]\,, \quad S = (S_1, -, S_3)$$

  - partial ordering among structures

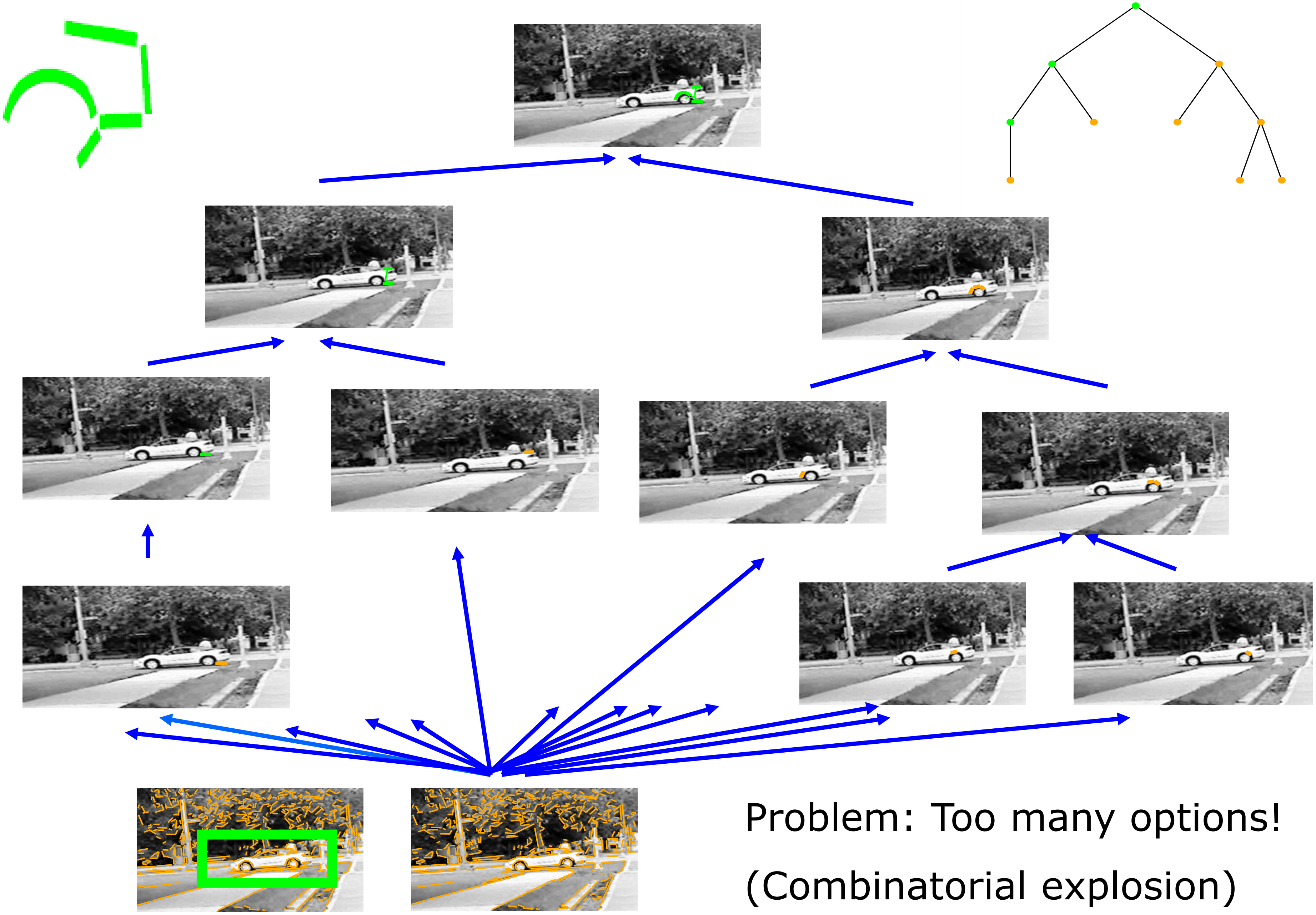
$$S^i \preceq S^j \leftrightarrow I_k(S^i) \leq I_k(S^j) \quad \forall k$$

- Hasse Diagram for 3-partite structure



  - By acquiring a substructure, the structure climbs upwards

Composition of the `Back' Structure



Problem: Too many options!

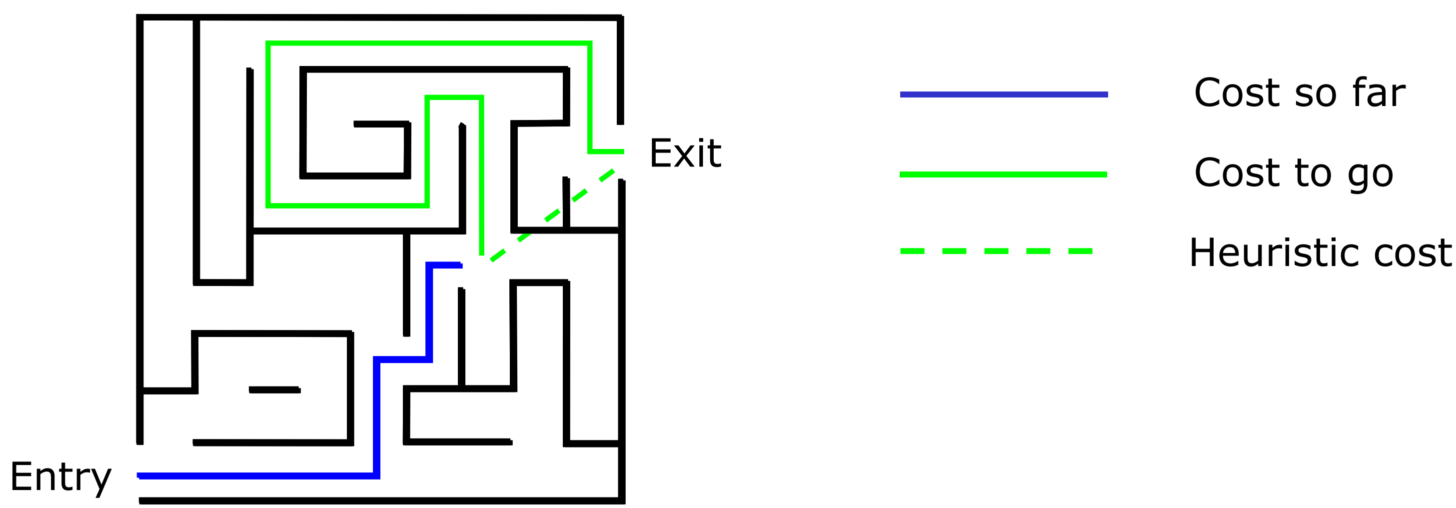
(Combinatorial explosion)

Analogy: Building a puzzle

- Bottom-Up solution: Combine pieces until you build the car
  - Does not exploit the box' cover
- Top-Down solution: Try fitting each piece to the box' cover.
  - Most pieces are uniform/irrelevant
- Bottom-Up/Top-Down solution:
  - Form car-like structures, but use cover to suggest combinations.

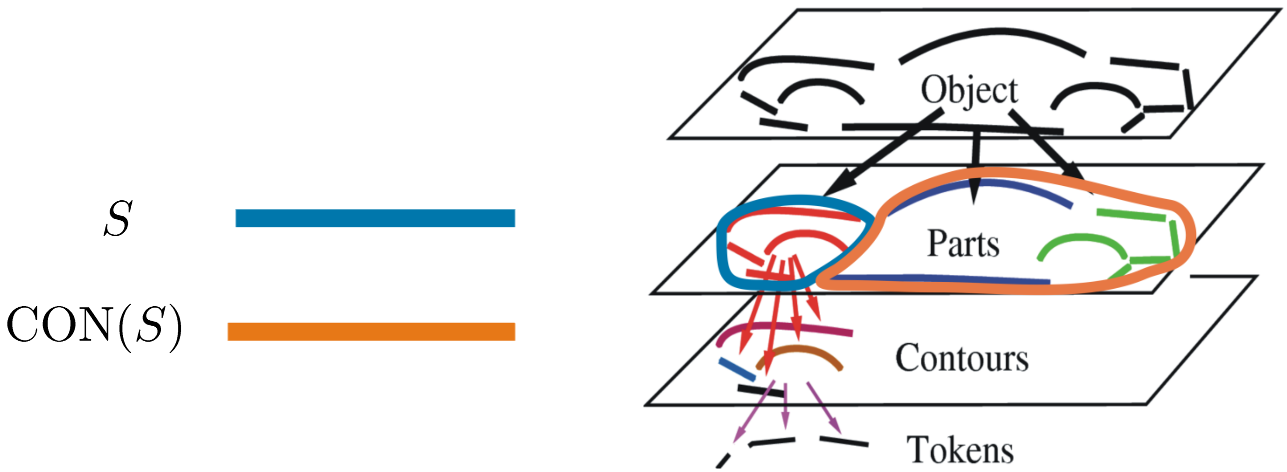
Best First Search

- Dijkstra's Algorithm
  - Prioritize based on `cost so far'
  - For parsing: Knuth's Lightest Derivation
- A* Search
  - Consider `cost to go'
  - Approximate with **heuristic** cost



‘Cost to go’ for Parsing

- The Generalized A* Architecture, Felzenszwalb & McAllester
- Context: complement needed to get to the goal.



- Recursive derivation of contexts.

$$\mathrm{CON}(goal) = 0$$

$$(S_1 = w_1, S_2 = w_2) \rightarrow (S_3 = w_3)$$

$$(S_1 = w_1, S_2 = w_2, S_3 = w_3, \mathrm{CON}(S_3) = w_c) \rightarrow \begin{array}{l} (\mathrm{CON}(S_1) = w_3 + w_c - w_1) \\ (\mathrm{CON}(S_2) = w_3 + w_c - w_2) \end{array}$$

Heuristics for Parsing: Context Abstractions

- A* requires lower bound of derivation cost
- Derive context in coarser domain (*abstraction*)
  - Lower bound cost on fine domain

$$\mathrm{Cost}(\mathrm{CON}(Abs(S_3))) \leq \mathrm{Cost}(\mathrm{CON}(S_3))$$

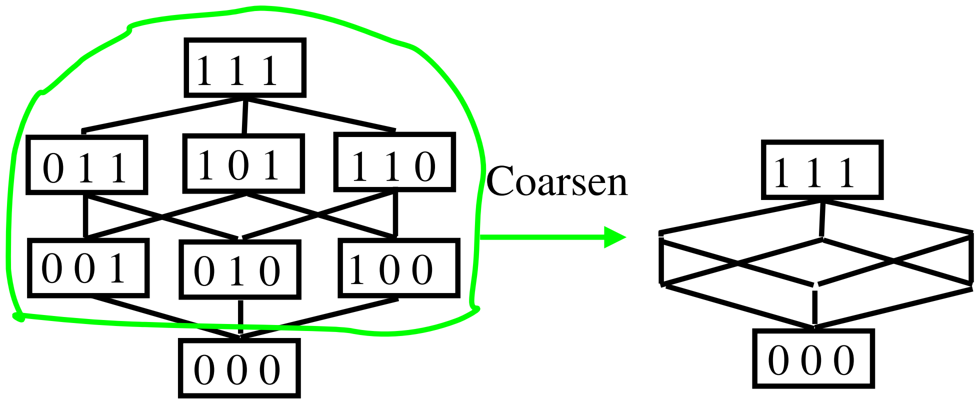
- Use it to prioritize search

KLD: $(S_1 = w_1, S_2 = w_2) \rightarrow_{w_3} (S_3 = w_3)$

A* : $(S_1 = w_1, S_2 = w_2, \mathrm{CON}(Abs(S_3)) = w_h) \rightarrow_{w_3 + w_h} (S_3 = w_3)$

Abstractions via Structure Coarsening

- Coarsening: identify nodes of Hasse diagram



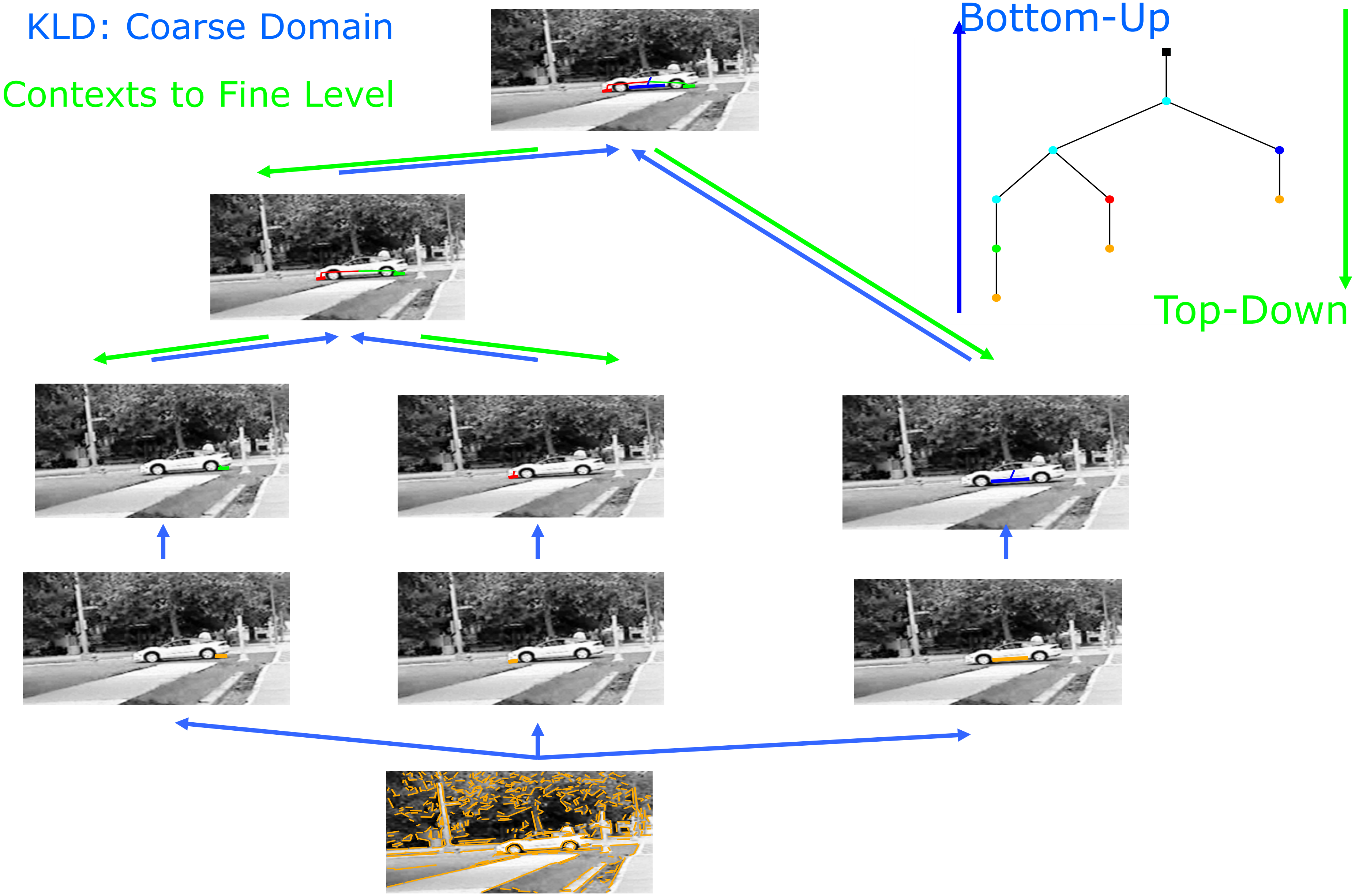
1 part suffices

- Lower bound composition cost $\sum_{p \in P} \log P_{p|O}(\mathbf{p}_{p|O})$

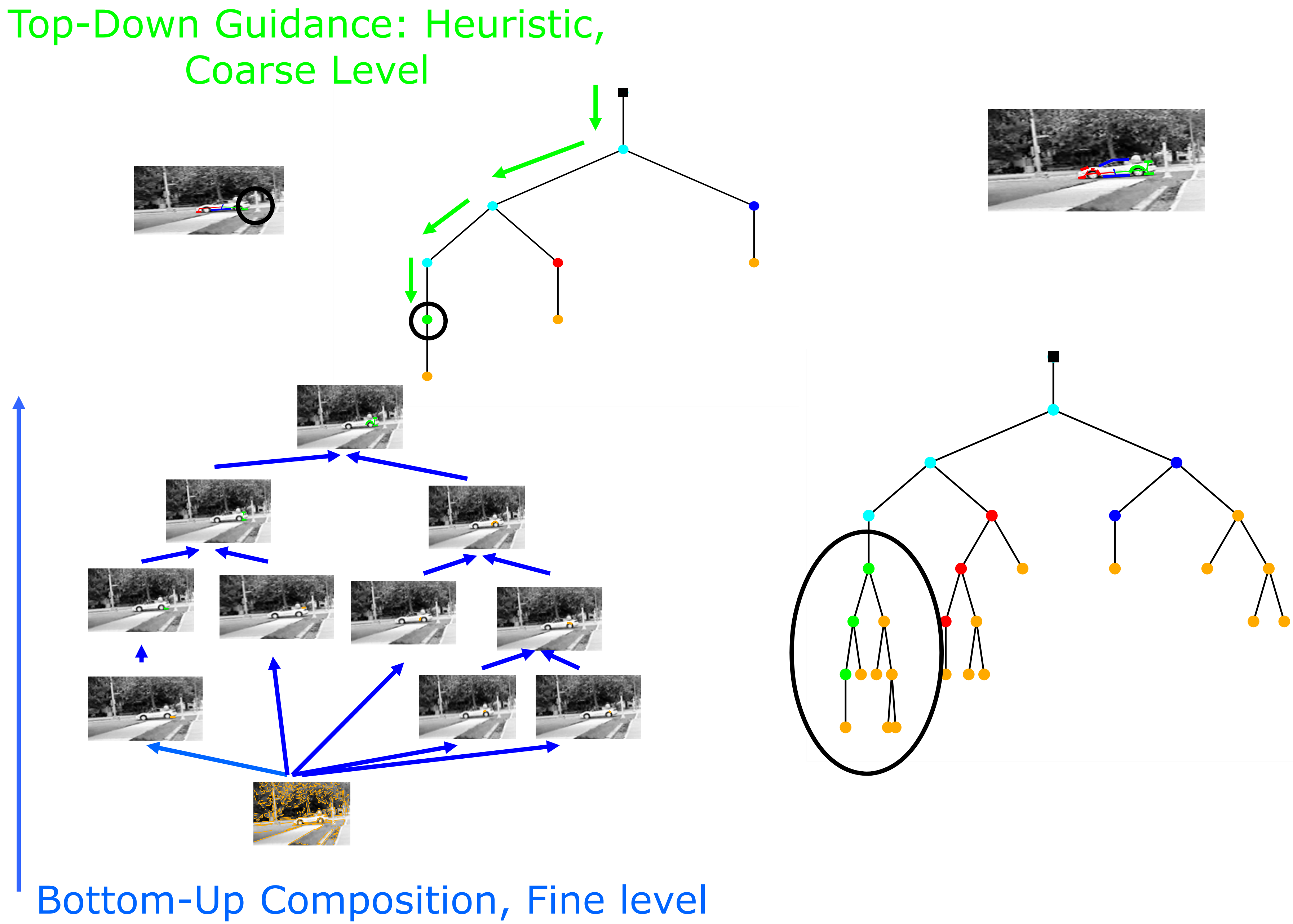
$$\sum_{p \in P} \log P_{p|O}(\mathbf{p}_{p|O}) = \sum_{p \in P} \frac{1}{2} \left[\log((2\pi)^n |\Sigma_{p,O}|) + \mathbf{p}_{p|O}^T \Sigma_{p,O}^{-1} \mathbf{p}_{p|O} \right] \geq$$

$$\underbrace{\frac{1}{2} \left[\log((2\pi)^n |\Sigma_{a,O}|) + \mathbf{p}_{a|O}^T \Sigma^{-1} \mathbf{p}_{a|O} \right]}_{C_a} + \sum_{p \in P \setminus a} \max\left(\frac{1}{2} \log((2\pi)^n |\Sigma_{p,O}|), C_a \right)$$

Coarse Level Parsing



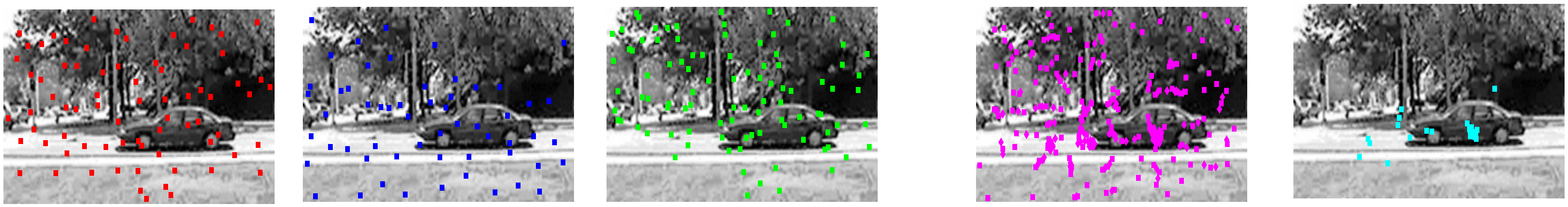
Fine Level Parsing



A* versus Best First Parsing

- A* Parsing

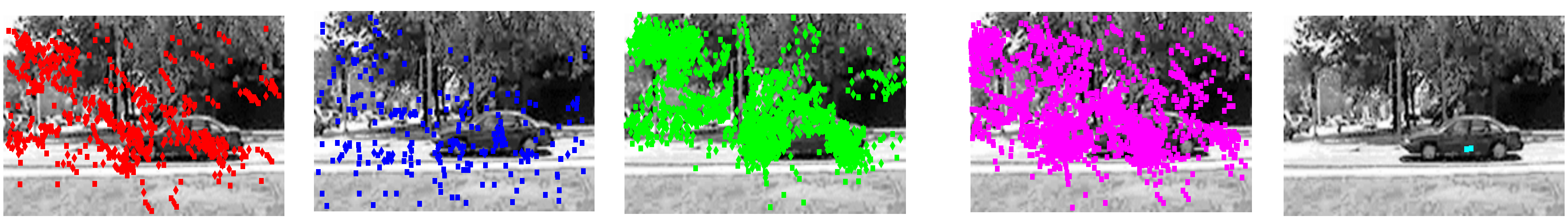
Front Part | Middle Part | Back Part | Object | Goal



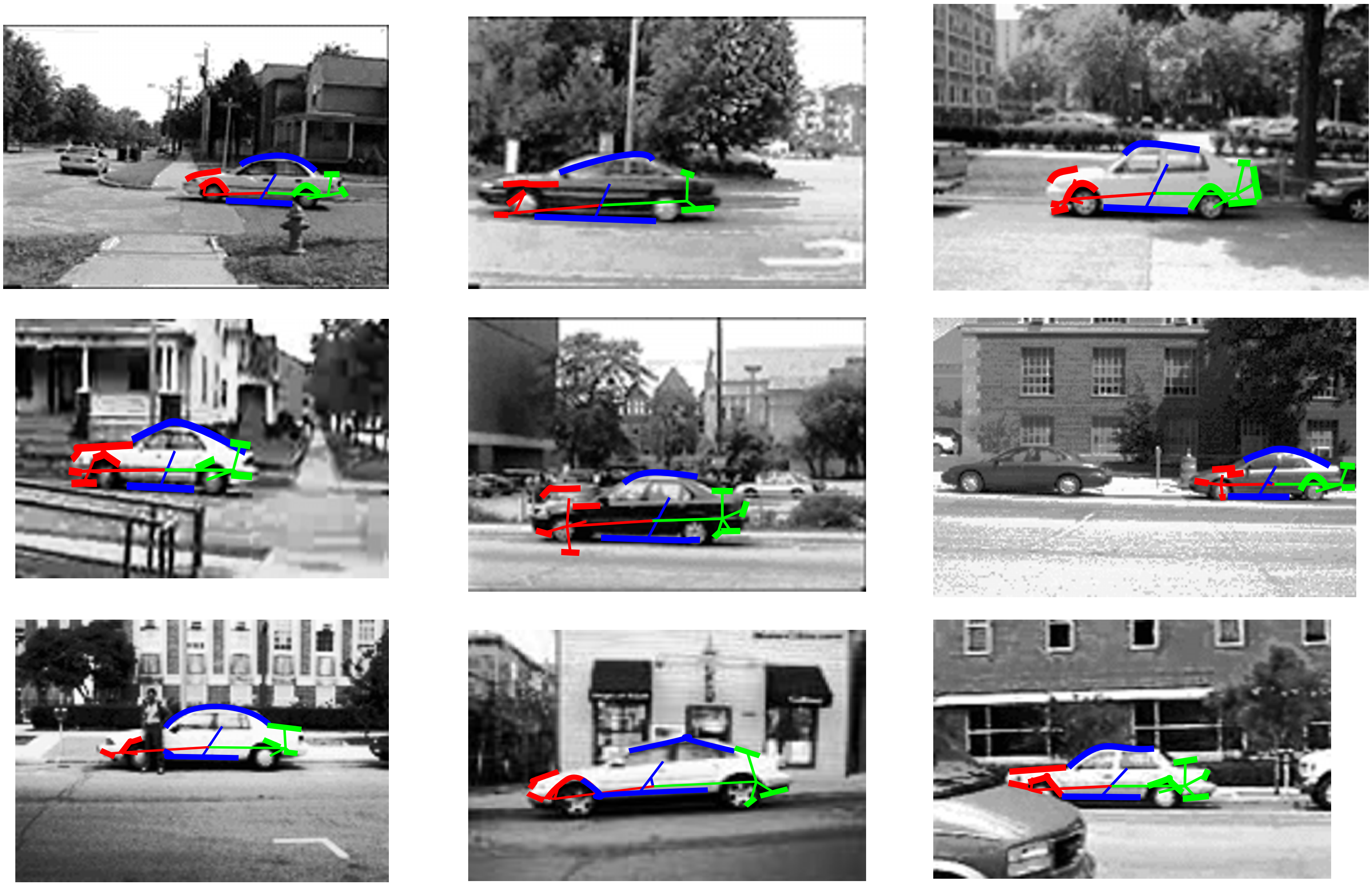
Coarse Level

Fine Level

- Knuth's Lightest Derivation Parsing



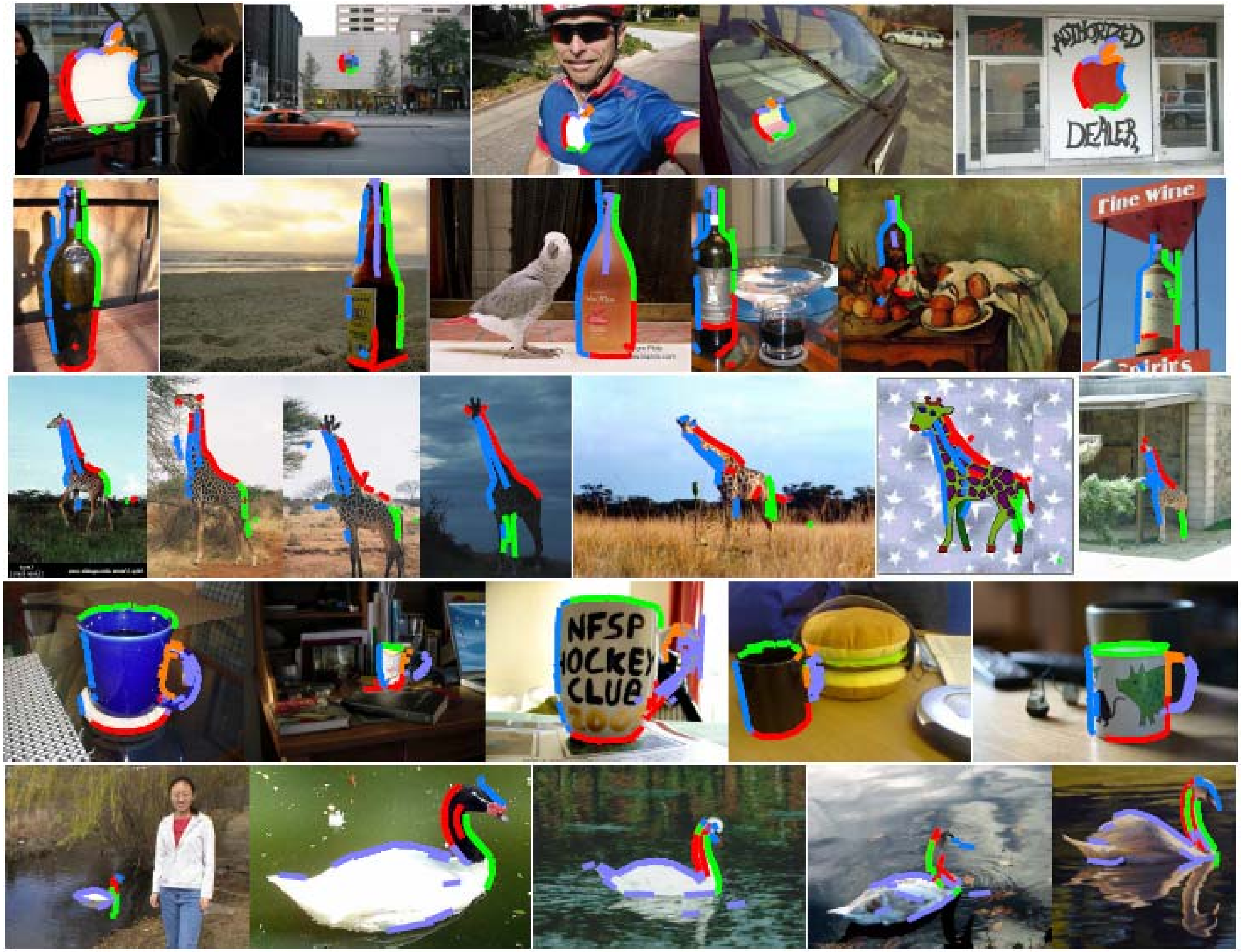
Parsing & Localization Results - I



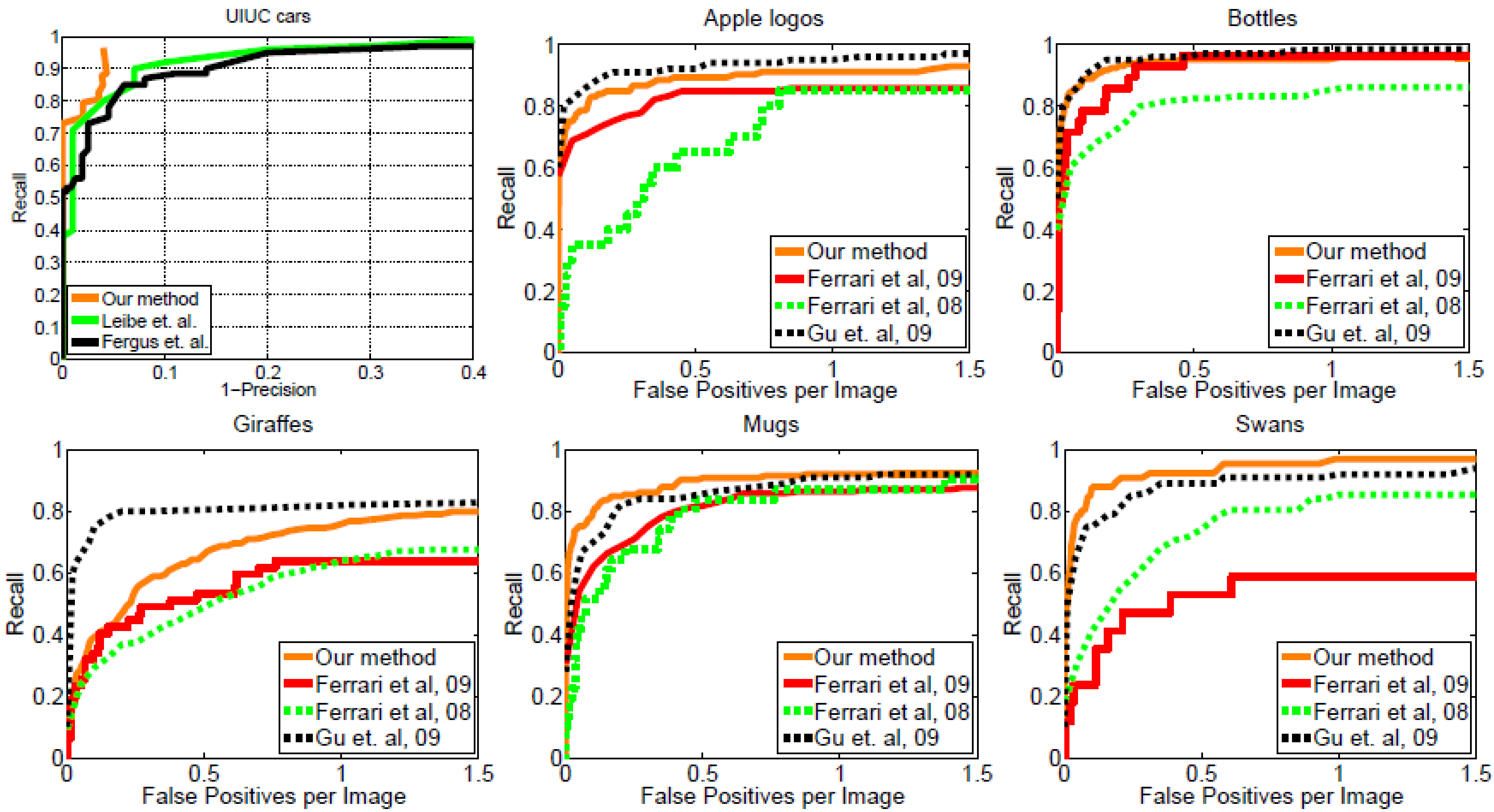
Parsing & Localization Results - II

Object – ETHZ Shape dataset

Parsing and localization results



ETHZ Benchmark results



Forward pointers

- Learning the model parts:
  - Statistical Shape Models, 3rd week
- Learning the model parameters:
  - Latent SVM training, 3rd week
- Branch & Bound for star-shaped models:
  - Rapid Object Detection with Branch & Bound, 3rd week
    - spatial coarsening
    - score bounding