Models and Algorithms

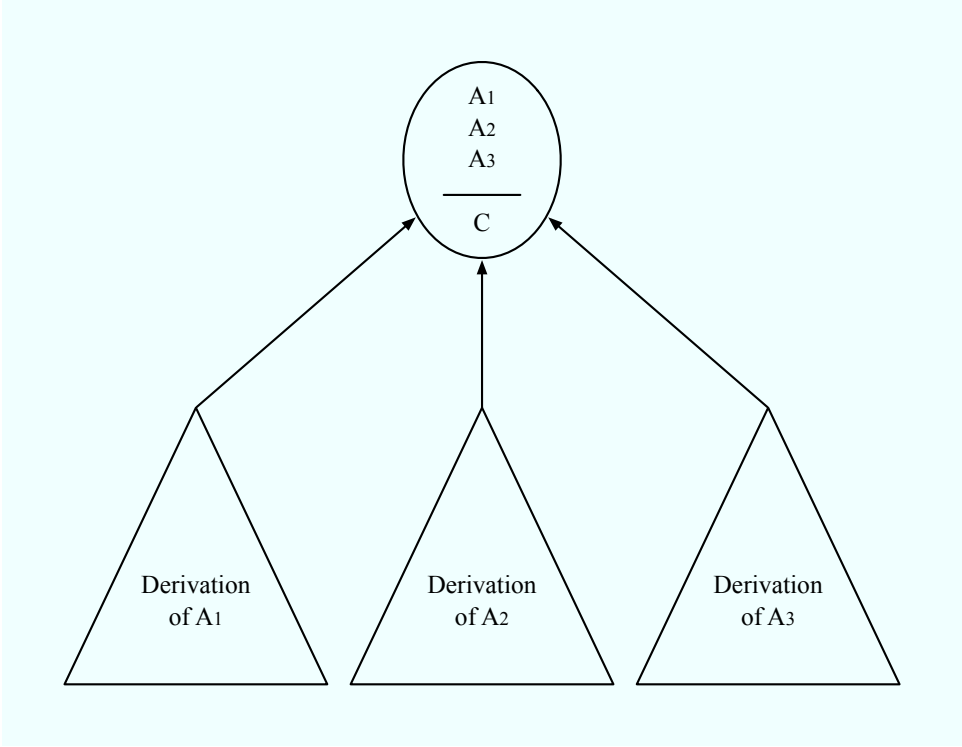
Image Parsing

Pedro Felzenszwalb and David McAllester

Lightest Derivation Problem

Let Σ be a set of statements and R be a set of weighted rules

$$\begin{array}{l} A_1 = w_1 \\ \quad \vdots \\ A_n = w_n \\ \hline C = g(w_1, \ldots, w_n) \end{array}$$



A *derivation* of C is a tree of rules

We want to find a lightest derivation of a *goal* $\in \Sigma$

Shortest Paths

$path(s) = 0$

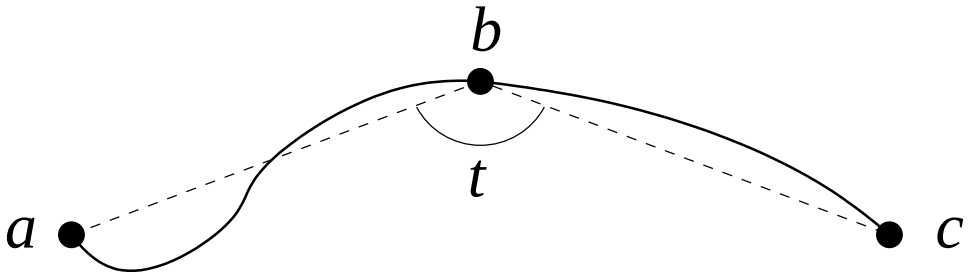
For each edge (u, v):

$path(u) = w$

$path(v) = w + w(u, v)$

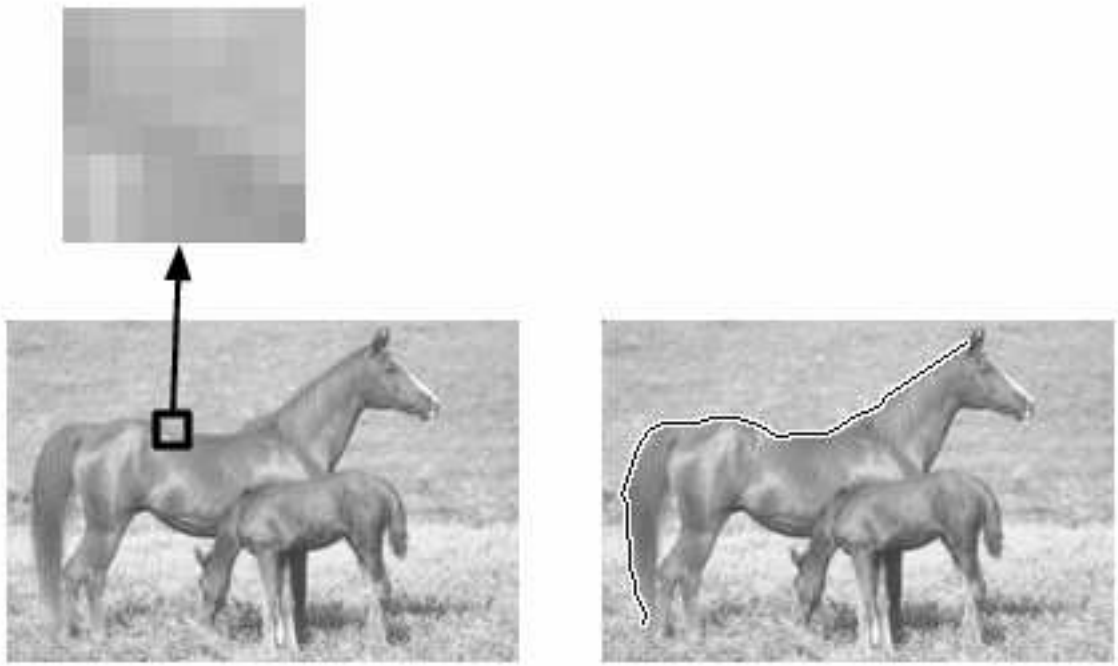
Lightest derivation of $path(v)$ is a shortest path from s to v.

Salient Curves



$$\frac{\mathit{curve}(a,b) = w_1 \quad \mathit{curve}(b,c) = w_2}{\mathit{curve}(a,c) = w_1 + w_2 + \mathit{shape}(a,b,c)}$$

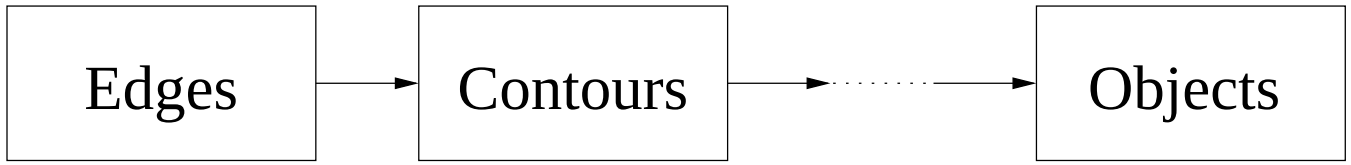
$$\frac{||a-b|| \leq k}{\mathit{curve}(a,b) = \mathit{grad}(a,b)} \qquad \frac{\mathit{curve}(a,b) = w}{\mathit{goal} = w - \lambda||a-b||}$$



Lightest derivation of *goal* is the "most salient curve".

Curve may go over pixels with low gradient.
This is a form of top-down influence.

Vision Pipeline



Vision system are often defined in terms of processing stages.

Can define each stage using rules for building structures by composing structures from previous stage.

Solving a single lightest derivation problem allows high-level knowledge to influence low-level interpretations.

But leads to a difficult computational problem.

Prioritized Rules

Rules $A_1, \ldots, A_n \rightarrow_g C$ with priority function $p(w_1, \ldots, w_n)$.
Execution generates weight assignments $(B = w)$.

$\mathcal{S}$: set of assignments
$\mathcal{Q}$: priority queue of assignments

Procedure *Run*(P)
1. $\mathcal{S} \leftarrow \emptyset$
2. Initialize $\mathcal{Q}$ using rules with no antecedents
3. **while** $\mathcal{Q}$ is not empty
4. Pop assignment $(B = w)$ from $\mathcal{Q}$
5. **if** B has no assigned weight in $\mathcal{S}$
6. $\mathcal{S} \leftarrow \mathcal{S} \cup \{(B = w)\}$
7. Push assignments derivable in one step using $\mathcal{S}$ into $\mathcal{Q}$
8. **return** $\mathcal{S}$

Knuth's Lightest Derivation

Define prioritized rules $\mathcal{K}(R)$ by setting the priority of each rule in R to be $g(w_1, \ldots, w_n)$.

- Generalizes Dijkstra's shortest paths algorithm.
- Computes lightest derivations in increasing weight order.
- We can stop without deriving every statement.
- Requires *non-decreasing* and *superior* weight functions. Example: $g(w_1, \ldots, w_n) = w_1 + \cdots + w_n + v$

A* Lightest Derivation

Let $h(C)$ be a heuristic function:
estimate of the additional cost for deriving *goal* using C.

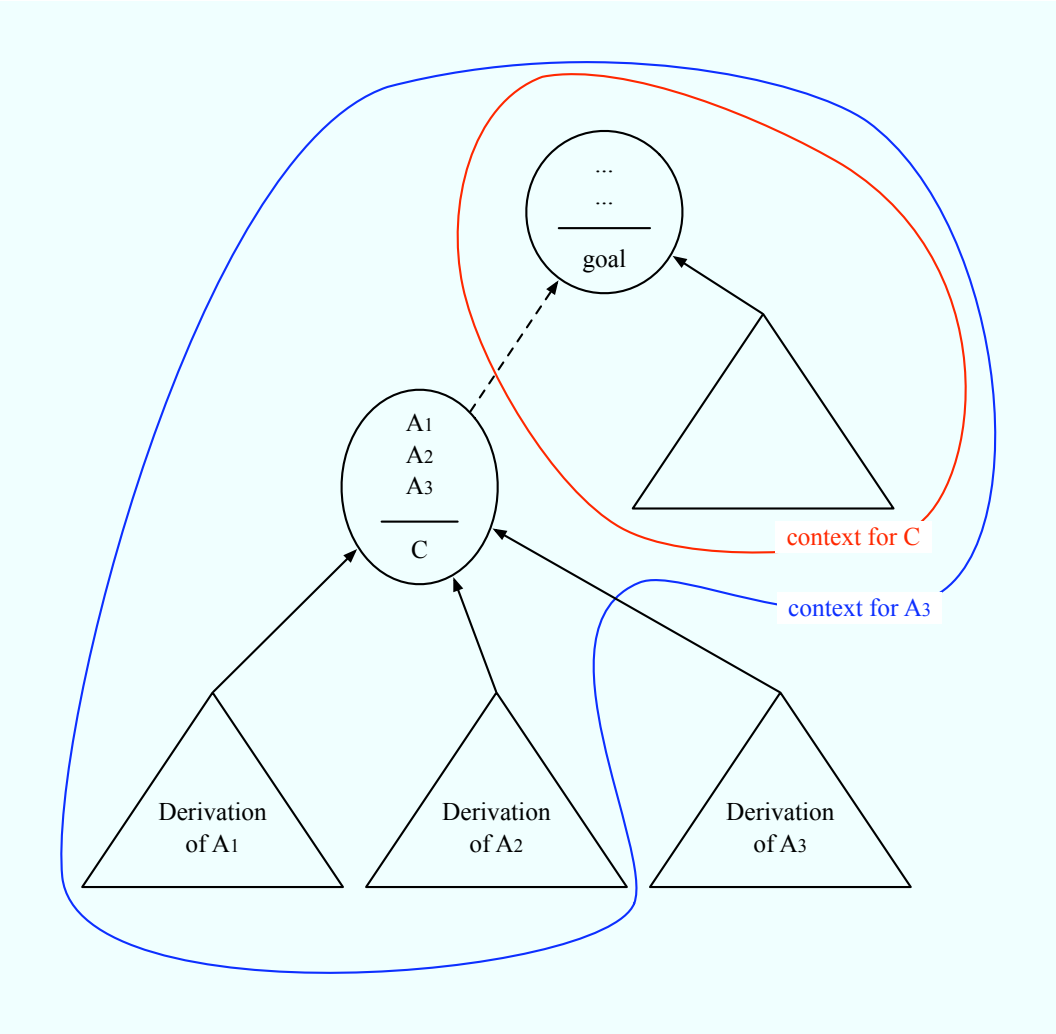
A*LD: Define prioritized rules $\mathcal{A}(R)$ by setting the priority of each rule in R to be $g(w_1, \ldots, w_n) + h(C)$.

A*LD computes derivations in increasing $\ell(C) + h(C)$ order.

Correctness requires monotonicity: for rule $A_1, \ldots, A_n \rightarrow_g C$ and derivable assignments $(A_i = w_i)$,

$$w_i + h(A_i) \leq g(w_1, \ldots, w_n) + h(C)$$

A *context* for B is a derivation of *goal* with a "hole" that can be filled in by a derivation of B.



Can define contexts in terms of a lightest derivation problem.

For each rule $A_1, \ldots, A_n \rightarrow_v C$ define

$$\mathit{context}(C), A_1, \ldots, A_{i-1}, A_{i+1}, \ldots, A_n \rightarrow_v \mathit{context}(A_i)$$

Lightest context weights are *perfect* heuristics:

If $h(C) = \ell(\mathit{context}(C))$, A*LD only derives statements that are in a lightest derivation of *goal*.

Abstractions

An abstraction of a problem (Σ, R) is defined by a smaller problem (Σ', R') and a map $\mathit{abs}: \Sigma \to \Sigma'$ such that

$$A_1, \ldots, A_n \to_v C \in R \;\Rightarrow\; \mathit{abs}(A_1), \ldots, \mathit{abs}(A_n) \to_{v'} \mathit{abs}(C) \in R'$$

with $v' \leq v$.

The abstract problem defines lower bounds:

$$\ell(C) \geq \ell(\mathit{abs}(C))$$

$$\ell(\mathit{context}(C)) \geq \ell(\mathit{context}(\mathit{abs}(C)))$$

$h(C) = \ell(\mathit{context}(\mathit{abs}(C)))$ is monotone.
We can pre-compute h and use it to solve the concrete problem.

Abstraction for Curves

Define a partition of the image into boxes.

$\mathit{abs}(\mathit{curve}(a, b)) = \mathit{curve}(A, B)$
A and B are the boxes containing a and b.

$\mathit{curve}(A, B) = w_1$
$\mathit{curve}(B, C) = w_2$

$$\mathit{curve}(A, C) = w_1 + w_2 + \mathit{shape}(A, B, C)$$

$$\mathit{shape}(A, B, C) \leq \min_{a \in A, b \in B, c \in C} \mathit{shape}(a, b, c)$$

Hierarchical A*

Sequence of problems (Σ_k, R_k) with *abs* mapping Σ_k onto Σ_{k+1}.

(Σ_{k+1}, R_{k+1}) is abstraction of (Σ_k, R_k).

Simultaneously compute contexts and derivations at every level.

Each level guides the one below.

HA*LD

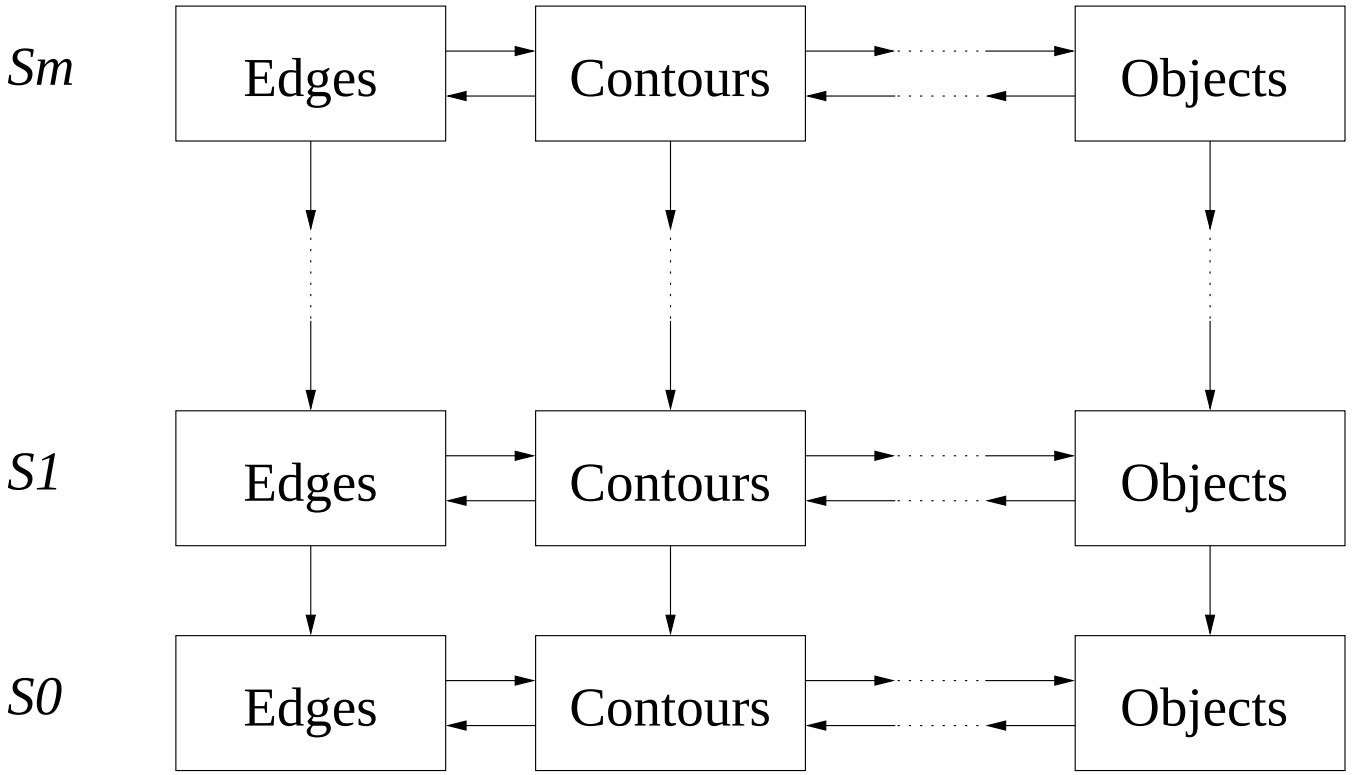
For each rule $A_1, \ldots, A_n \rightarrow_v C$ in R_k,

$$\frac{\begin{array}{l} \mathit{context}(\mathit{abs}(C)) = w_c \\ A_1 = w_1 \\ \vdots \\ A_n = w_n \end{array}}{C = \sum w_i + v + w_c}$$

$$\frac{\begin{array}{l} \mathit{context}(C) = w_c \\ A_1 = w_1 \\ \vdots \\ A_n = w_n \end{array}}{\mathit{context}(A_j) = \sum w_i - w_j + w_c + v}$$

$$\frac{\mathit{goal}_k = w}{\mathit{context}(\mathit{goal}_k) = 0}$$

Hierarchical Pipeline



Contexts from one level of abstraction guide computation at more concrete level.