A SIFT-based Pictorial Image Model

James Lenfestey

The Problem

These buggers are relentless...

...if only I had my own cognitive robot!

I will discuss

- target models for object recognition
- SIFT features
- undirected models and the pictorial framework
- message-passing algorithm for target identification

SIFT Image Features (Lowe 99)

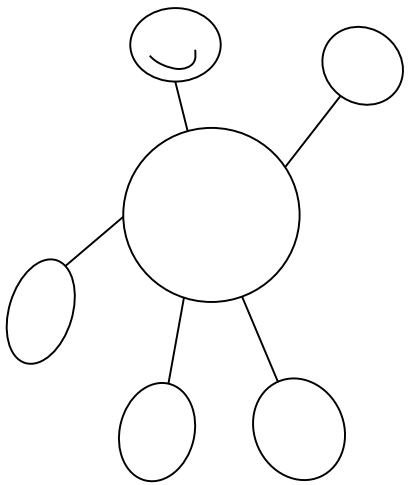
- associates high-dimensional descriptor to keypoints
- engineered to be scale and rotation invariant
- robust to noise and small affine motions

SIFT Image Features

- Lowe defines image queries against a set of predefined images
- neighboring keypoints confer on presence and orientation of a query image
- Not good enough for Quake:
  - enemy images deformable
  - many target different poses

Pictorial Image Models

- model image as loosely conglomerated ensemble of objects spatially related
- each object induces the presence of distinctive keypoints in some area of the image
- represent spatial relationships with undirected graphical model



Pictorial Image Models

- undirected model encodes conditional independence: $m.b.(node) = neighbors$
- $p(x) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(x_C)$
- if our graph is a tree, becomes a product of pairwise interactions: $p(x_i, x_j), (i, j) \in E$.
- in my model, $p(x_i, x_j) = \mathcal{N}(x_i - x_j; \mu_{ij}, \sigma_{ij} I)$
- this encodes the prior distribution of model configurations
- after observing an image I, we want to estimate the posterior: $p(x|I) \propto p(I|x)p(x)$

Observation Model

- $p(I|x)$ encodes the observation model
- I is a set of SIFT keypoints sampled i.i.d. from a gaussian mixture model

$$p((l_i, d_i)|x) = \sum_j \alpha_j \mathcal{N}(l_i; x_j, \sigma_j)\mathcal{N}(d_i; \mu_j, \Sigma_j)$$

- means/variances for each component in both image and descriptor space

MAP estimation

- if we assume $p((l_i, d_i)|x) = \prod_j p((l_i, d_i)|x_j)$, we can use the tree structure of the graph to perform efficient inference
- generalization of Viterbi
- use message passing:
  - pick a leaf j, child of i
  - tabulate
    $$M_j(x_i) = max_{x_j}\ p(x_i, x_j) p(I|x_j) \prod_c M_c(x_j),$$
    where c indexes over the (original) children of j.
  - remove j, pass the message to i and iterate

EM Learning

- The MAP procedure assumed we know the parameters observation model parameters
- we don’t; learn them using EM
  - given pose and observation model, infer a distribution on mixture components for each keypoint
  - use these to get ML estimates for the observation and pose model parameters
  - get MAP estimates of component locations
  - iterate

Results inconclusive!

- still in the throes of EM
- the MAP estimate can be very expensive if performed naively (i.e. $O(h^2)$, where h is the number of possible pixels).
- I downsampled, but better algorithms are available
- because this estimation is too slow, I have
  - too *much* data to learn the models in a short time,
  - and too *little* data to learn reasonably sophisticated models (e.g. isotropic covariance assumption throughout).