# (1) Deformable Templates & POMs

Note Title 11/17/2007

Designing a deformable template representing the object and specifying an inference algorithm.

It is attractive to learn the representation of the object from training examples — this involves both learning the structure of the representation and the parameters of the representation.

This lectures shows how to learn the object representation in an unsupervised manner.

We use a structure pursuit strategy that starts with a simple representations and grows it. We use proposals based on clustered to suggest ways to grow the structure.

(2) As we grow the representation, the model will be able to perform an increasing number of tasks.

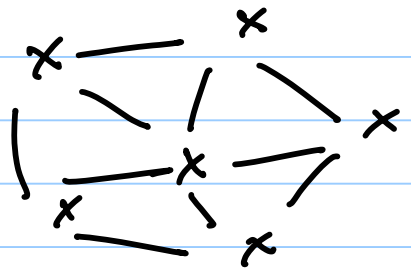
Recall that most deformable templates are designed to perform individual tasks
→ e.g. matching, detection, classification and segmentation.

But a "real" deformable template model should be able to perform all tasks.

(3) The building blocks of the model.

Probabilistic Object Model (POM)

POM-1 sparse model



Attributed feature. AF

$(z_a, A_a, \theta_a)$

$z$ position, $A$ attribute, $\theta$ orientation

But. this model needs to be enhanced by

(i) a model for the background.

(ii) a method for allowing the object to have several different appearances, or aspects

(iii) to build invariance to scale and rotation into the model.

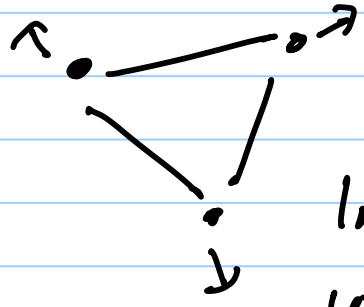
And, we need to ensure rapid inference, and structure & parameter learning.

(4) Image Features.

Detect interest points (IP's) by the Brady-Kadir detector. This detector locates places in the image where the entropy is high. The detector is invariant to scale.

Describe the interest points by the SIFT feature descriptor (LOWE). This descriptor outputs orientation plus a description of the local appearance.

Spatial Combinations of AF - Oriented Triplets.



We can define an Invariant Triplet Vector (ITV) which is invariant to the scale and orientation of the triangle.

$$l(z_1, \theta_1, z_2, \theta_2, z_3, \theta_3)$$

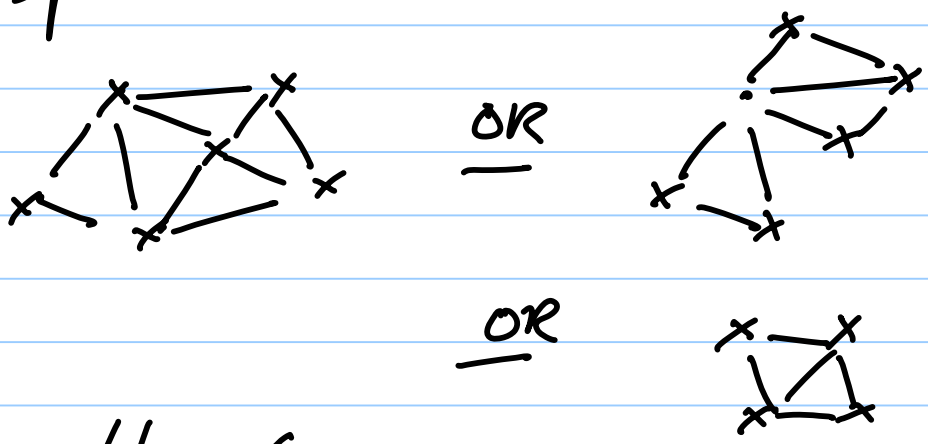
The prior will depend on $l$ only, hence it is invariant to scale & orientation

(5) Background Model.

uniform distribution of AF's in the background.

Different Aspects/Appearance

Allow the object to have several different aspects. This is a mixture model.




Pose Variable G

this specifies the position, orientation, and scale of the object.

Assignment Variables V

a correspondence between AF's in the model and AF's in the image.

(6) Probabilistic Formulation.

$P(z, A, \theta \mid V, y, G, \omega)\, P(V \mid y, \omega)\, P(y \mid \Omega)\, P(G)\, P(\omega)\, P(\Omega)$

Grammar Component $P(y \mid \Omega)$ prior $P(\Omega)$

$y = (s, n)$

$s$ ↑ aspect, $n$ ↑ no of background nodes.

$\sum_{i=1}^{M} \Omega_i = 1$

$P_\Omega(s)$ for the aspect $P_\Omega(s=i) = \Omega_i$

$P_\Omega(n) = \Omega_b e^{-n\Omega_b}$ for no. of background nodes.

(Note: the parameters $\Omega_b, \{\Omega_i\}$ must be learnt and the no. of aspect models $M$ must be learnt.

(Note: the paper uses a simple grammar with a single OR node - e.g. a mixture model - but, in theory, any grammar can be used here.)

$P(G)$ is a uniform distribution -
all poses (position, scale, orientation) are equally likely.

(7) $P(V|S,\{\lambda_S\})$ specifies the prior probability that a point generated by the model will be visible in the image.

Each aspect $S$ is represented by a set of AF's $a=1,2,\ldots,n_S$ which are organized in terms of cliques

$(1,2,3), (2,3,4), (3,4,5), (4,5,6) \ldots (n_S-2, n_S-1, n_S)$

Each clique is an ordered triplets of AF's

$$P(V|S,\omega) = \frac{1}{Z} \prod_{a=1}^{N_S} e^{-\lambda_S \delta_{i(a),0}}$$

where 0 is the clutter index.

Note: the set of parameters $\{\lambda_S\}$ must be learned. $\{\lambda_S\} \in \omega$

(8) $P(z, A, \theta \mid V, y, G, \omega)$

$= P(A \mid V, y, \omega)\, P(z, \theta \mid G, V, \omega)$

We specify the appearances by distribution

$$P(\{A_i\} \mid y, \omega^A) = \prod_{a=1}^{N_s} P(A_{i(a)} \mid \omega^A_{s,a})$$

$\propto$ Gaussian distributions.

The $\{\omega^A_{s,a}\}$ are the mean and covariance of the Gaussian distribution.

This assumes a uniform distribution on the appearances of the background pixels.

Note: the parameters $\{\omega^A_{s,a}\}$ must be learnt.

We can express this distribution in exponential form $\frac{1}{Z} \exp \sum_{a=1}^{N_s} \lambda_{i(a)} \cdot \phi(A_{i(a)})$

(9) $P(z, \theta | G, V, \omega)$

We make an approximation here to simply the inference.

Define a distribution on the ITV's $\underline{l}_a$ the cliques $a$ of the aspect.

$$P(\underline{l} | y, V) = \frac{1}{Z} \prod_{a=1}^{N_a - 2} e^{\underline{\lambda}^s_a \cdot \underline{\psi}_a(\underline{l}_{i(\omega)})}$$

This will be product of Gaussian distributions on the ITV's of the aspect

The parameters $\{\underline{\lambda}^s_a\}$ must be learned.

This distribution will put prior constraints on the spatial poses and orientations of the AF's, $\{z_{i(a)}, \theta_{i(a)}\}$

We represent the points by $l, G$ or by $z, \theta$,

$$l = l(z, \theta; V), \quad G = G(z, \theta; V)$$
$$z = z(l, G : V), \quad \theta = \theta(l, G : V)$$

Specify $P(z, \theta | G, V, \omega) = \frac{1}{Z} P(l(z, \theta; V) | G, V, \omega) \, \delta(G - G(z, \theta; V))$

(10) We can combine all the terms in the distribution to get

$$P(z, A, \theta | V, y, G, \omega) P(V | y, \theta) P(y | \Omega) P(G) P(\omega) P(\Omega)$$

We now have three tasks:

(1) Inference: Detect the object in the image. Estimate V & G.

(2) Parameter learning: Estimate the parameters of the model assuming that the structure (i.e. no of aspects, no. of nodes in each aspect) is known.

(3) Estimate the structure of the model.

The structure estimation is the hardest task. There are many possible structures → how to select the structures.

Now address these three tasks in turn.

(11) Inference: For each aspect, the model can be expressed in terms of products over ordered cliques.

$$\prod_{a=1}^{N_a-2} \pi_a [i(a), i(a+1), i(a+2)].$$

This can be optimized using dynamic programming. (Junction Trees Algorithm).

Recursively define.

$$h_{a+1}((z,A,\theta)_{i(a+1)}, (z,A,\theta)_{i(a+2)})$$

$$= \max_{i(a)} \{ \pi_a [(z,A,\theta)_{i(a)}, (z,A,\theta)_{i(a+1)}, (z,A,\theta)_{i(a+2)}]$$
$$h_a((z,A,\theta)_{i(a)}, (z,A,\theta)_{i(a+1)}).$$

This is the forward pass.
Backward pass determining the optimal correspondence $\{\hat{i}(1), \hat{i}(2), \ldots, \hat{i}(N_s)\}$.

Compute the optimal match for each aspect. Select the aspect and match with the best score.

# (12) Parameter Learning.

The no. of aspect is known and the number of nodes for each aspect is known.

The input is a set of images which contain the object in different poses (unknown). (Generalization allow the object to only present in some images, or to be one of a set of objects).

The parameters to be learnt include the appearance models, the geometry probabilities (distributions on the ITU's), the probability for AF's to be undetected, the probability of each aspect, the probability of the number of background points.

There are unknown variables — the aspects for each image, the assignments for each image, the pose.

(13) # Parameter learning.

Set of images $\Lambda$

The EM algorithm,

$$(\omega^*, \Omega^*) = \arg\max_{\omega, \Omega} P(\omega, \Omega) \prod_{\tau \in \Lambda} \sum_{y_\tau, V_\tau} P(y_\tau, V_\tau, X_\tau | \omega, \Omega)$$

where $X_\tau = \{(z, A, \theta)_\tau\}$ for the image $\tau$

EM introduces a distribution.

$q(\{y_\tau, V_\tau\})$ for the aspects & correspondences for each image.

This factorizes into a product of distributions for each image. $\prod_\tau q_\tau(y_\tau, V_\tau)$

The EM algorithm has two steps:

M-step

$$\omega^{t+1}, \Omega^{t+1} = \arg\min_{\omega, \Omega} \left\{ - \sum_{\{y_\tau\}\{V_\tau\}} q^t(\{y_\tau, V_\tau\}) \log P(\omega, \Omega, \{y_\tau\}, \{V_\tau\} | \{X_\tau\}) \right)$$

E-step

$$q^{t+1}(\{y_\tau, V_\tau\}) = P(\{y_\tau\}, \{V_\tau\} | \omega^{t+1}, \Omega^{t+1}, \{X_\tau\})$$

(14) The form of the distribution makes the E-step practical. (explicit formula)

The M-step is also practical, because:

(i) the distribution is of exponential form:

$$\log P(\omega, \mathcal{R}, \{y_\tau\}, \{v_\tau\} \mid \{x_\tau\}) = \underline{\lambda}^{\omega,\mathcal{R}} \cdot \phi(\{y_\tau, v_\tau, x_\tau\})$$

(ii) the summation over $\{y_\tau\}, \{v_\tau\}$ is possible because we can sum over the aspects exhaustively and we can use dynamic programming (replacing max with sum) to sum over the correspondences.

But the EM algorithm requires good initial estimates of the parameters and we also need to know the structure of the distribution.

(15) # Structure Learning.

We define a score function for any structure:

$$P(\{x_t\} \mid (\hat{\omega}, \hat{\pi})) = \prod_t \sum_{\{v_t, y_t\}} P(\{x_t\}, \{v_t, y_t\} \mid (\hat{\omega}, \hat{\pi}))$$

This computation is possible because we can exploit the structure of the model to do the summations over $\{v_t, y_t\}$ — sum over the aspects exhaustively, and use dynamic programming to sum over the correspondences.

The score enables us to rank different structures.

But there are many possible structures so we need a procedure to decide which structures to score.

(16) # Structure Learning.

Use proposals for structures obtained by clustering on the image.

**Procedure:**

List all the I.P.'s found in the training images. $\{(z_i^\tau, A_i^\tau, \theta_i^\tau)\}$ $\tau \in \Lambda$, $i = 1 \text{ to } N_\tau$.

Perform clustering on the appearances $\{A_i^\tau\}_{\tau,i}$ $\Rightarrow$ K-means, or an alternative.

This gives an appearance vocabulary $\{\underline{a}_\alpha\}$, where each $\underline{a}_\alpha$ has a mean appearance and a variance.

Then we consider oriented triplets of $(z_i^\tau, \theta_i^\tau, z_j^\tau, \theta_j^\tau, z_k^\tau, \theta_k^\tau)$ where the appearances are fixed.

(17) We perform clustering on these triplets using the ITV $\underline{l}$

(we only cluster ITV with similar assignments of vertices to the appearance vocabulary).

This gives a triplet vocabulary $\{\underline{b}_\beta\}$, where $\underline{b}$ contains the mean and covariance of the ITV's.

We can use this triplet vocabulary (with the appearances) as proposals for triplets $\{(z_a, \theta_a, A_a), (z_{a+1}, \theta_{a+1}, A_{a+1}), (z_{a+2}, \theta_{a+2}, A_{a+2})\}$ in the model.

The elements in the vocabulary that occur most frequently are the best proposals.

Intuition → triplets that occur frequently are probably due to the object — because it is roughly invariant from image to image. Background triplets will occur only a few times.

(12) Structure Persist. proceeds as follows.

Initialize the model to be empty.
(i.e. all points are background).

Select the triplet from the triplet vocabulary with best value.

Propose an object model consisting of this single triplet
Use parameter learning to estimate the parameters $(\omega, \lambda)$
Evaluate the score of the model and compare it to the score of the default (all background) model.
If the new score is better, then make this the new default model.

Proceed in the same way.
Either add a new triplet to an existing aspect or make a new aspect, by a new triplets

Stop when the score stops increasing.

(19) Properties of the system.

(1) Good Performance on standard datasets (Caltech) evaluated for detection & classification.

But, this method is more robust to "noise" in the data. It will learn the object model even if a large fraction (e.g. 1/2) of the images do not contain the object.

(2). Invariance to scale & rotation, for learning and inference.

Manipulate the Caltech Dataset so that the pose of the objects varies considerably. The method learns and performs inference correctly.

(3.) Learning hybrid models for classes of objects. E.G Images contain airplanes, faces, or motorbikes. Different objects as different aspects

(4.) Inference speed - fast (seconds) per image.

(20) Limitations of the Model.

The reliance on Interest Points (IP) makes the learning easier — IP's are very sparse in the image and can have easily distinguishable appearances.

But IP's only give a limited representation for an object and only exploit a limited set of image cues (i.e. IP's).

Problems:

(i) It is impossible to perform tasks such as segmentation or parsing using IP's.

(ii) IP's only contain a limited set of image cues. Better performance on detection and classification if we use additional cues.