Pictorial Structures for Object Recognition

Mohamed Vadhel EBNOU OUMAR

Juan Esteban RIOS GALLEGO

Giovanni BENEDETTI DA ROSA

January 31, 2024

Outline

Introduction

- **Problem** : recognizing objects using generic part-based models.
- **Motivation** : pictorial structure representation (Fischler and Elschlager) (1973).

Pictorial Structures

Definition

- Collection of parts with connections between certain pairs of parts
- **Fischler and Elschlager (1973)** problem defined in terms of an energy to be minimized.
- Used to represent generic objects.

$$L^* = arg\ min_L(\sum_{i=1}^{n} m_i(l_i) + \sum_{(vi,vj)\in E} d_{ij}(l_i, l_j)) \quad (1)$$

Efficient Algorithm

Definition

- T_{ij} and T_{ji} are one-to-one.
- Let M_{ij} be a diagonal matrix

Mahalanobis distance:

$$d_{ij}(l_i, l_j) = T_{ij}(l_i) - T_{ji}(l_j)^\top M_{ij}^{-1}(T_{ij}(l_i) - T_{ji}(l_j)) \tag{2}$$

Statistical Framework

Definition

- Let θ be a set of parameters that define an object model;
- **I** denote an image;
- **L** denote a configuration of the object (a location for each part).

By Baye's Rule:

$$p(L \mid I, \theta) \propto \underbrace{p(I \mid L, \theta)}_{\text{posterior}} \underbrace{p(L \mid \theta)}_{\textit{prior}} \tag{3}$$

Statistical Framework

Definition

- The pictorial structure is parametrized $\theta = (u, E, c)$
- $u = \{u_1, ..., u_n\}$ are appearance parameters;
- E is set of edges E indicates which parts are connected
- $c = \{c_{ij} | (v_i, v_j) \in E\}$ are connection parameters.

We can write the prior distribution over object configurations by a tree-structured Markov random field with edge set E:

$$p(L \mid \theta) = p(L \mid E, c) = \frac{\prod_{(v_i, v_j) \in E} p\left(l_i, l_j \mid \theta\right)}{\prod_{v_i \in V} p\left(l_i \mid \theta\right)^{\deg v_i - 1}} = \prod_{(v_i, v_j) \in E} p(l_i, l_j \mid \theta)$$

(4)

Statistical Framework

So, we can rewrite the Baye's Rule (6), and that the likelihood of an image can be seen as the product of the individual likelihoods:

$$P(L \mid I, \theta) \propto \left(\prod_{i=1}^{n} p\left(I \mid l_i, u_i\right) \prod_{\left(v_i, v_j\right) \in E} p\left(l_i, l_j \mid c_{ij}\right) \right) \tag{5}$$

We can easily that is an equivalent to the energy function that is being minimized in equation (1), where $m_i(l_i) = log\ p(I|l_i, u_i)$ is a match cost measuring how well part v_i matches the image data at location l_i , and $d_{ij}(li, lj) = log\ p(li, lj|c_{ij})$ is a deformation cost measuring how well the relative locations for v_i and v_j agree with the prior model.

Learning Model Parameters

Given a set of example images $\{I_1, ..., I_m\}$ and the corresponding configurations $\{L_1, ..., L_m\}$ of the object and the model parameters $\theta = (u, E, c)$, assuming that each example was generated independently

$$p\left(I^1, \ldots, I^m, L^1, \ldots, L^m \mid \theta\right) = \prod_{k=1}^{m} p\left(I^k, L^k \mid \theta\right),$$

Using the Baye's Rule $p(I, L \mid \theta) = p(I \mid L, \theta)p(L \mid \theta)$, we can estimate the Maximum Likelihood:

$$\theta^* = \arg\max_{\theta} \prod_{k=1}^{m} p\left(I^k \mid L^k, \theta\right) \prod_{k=1}^{m} p\left(L^k \mid \theta\right) \tag{6}$$

Estimating the Dependencies

Goal: pick a set of edges that form a tree and the connection parameters for each edge (algorithm of Chow and Liu) described in, which estimates a tree distribution for discrete random variables. From equation (6) we get

$$E^*, c^* = \arg\max_{E,c} \prod_{k=1}^{m} p\left(L^k \mid E, c\right).$$

Rewriting using the prior:

$$E^*, c^* = \arg\max_{E,c} \prod_{(v_i,v_j)\in E} \prod_{k=1}^{m} p\left(l_i^k, l_j^k \mid c_{ij}\right).$$

We can characterize the "quality" $q\left(v_i, v_j\right)$of a connection between two parts as the probability of the examples under the ML estimate for their joint distribution.

$$E^* = \arg\max_{E} \prod_{(v_i,v_j)\in E} q\left(v_i, v_j\right) = \arg\min_{E} \sum_{(v_i,v_j)\in E} -\log q\left(v_i, v_j\right)$$

Solving for E is equivalent to compute the minimum spanning tree (MST) of a graph, We build a complete graph on the vertices V , weight $logq(v_i, v_j)$, with each edge (v_i, v_j).

- Kruskal's $\mathcal{O}(n^2 logn)$

Estimating the Appearance Parameters

The u^* can independently solve for the u_i^*,

$$u_i^* = \arg\max_{u_i} \prod_{k=1}^{m} p\left(I^k \mid l_i^k, u_i\right).$$

This is exactly the ML estimate of the appearance parameters for part v_i, given independent examples $\left\{\left(I^1, l_i^1\right), \ldots, \left(I^m, l_i^m\right)\right\}$.

Energy Minimization or MAP Estimate

- Problem Equation:

$$L^* = \arg\min_L \left(\sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i, v_j) \in E} d_{ij}(l_i, l_j) \right)$$

- Equation Components:
  - $m_i(l_i)$: Match cost for part i at l_i.
  - $d_{ij}(l_i, l_j)$: Deformation cost between parts i and j.
  - E: Edges connecting parts.
  - Objective: Minimize total cost for configuration L.
- Dynamic Programming Approach for Efficiency.

Dynamic Programming in Energy Minimization

- Leaf Node:

$$B_j(l_i) = \min_{l_j}(m_j(l_j) + d_{ij}(l_i, l_j))$$

- Non-Leaf Node:

$$B_j(l_i) = \min_{l_j}\left(m_j(l_j) + d_{ij}(l_i, l_j) + \sum_{v_c \in C_j} B_c(l_j)\right)$$

- Root Node:

$$L^* = \arg\min_{l_r}\left(m_r(l_r) + \sum_{v_c \in C_r} B_c(l_r)\right)$$

- Complexity:
  - Time: $O(nh^2)$.
  - Due to h^2 combinations per node for n nodes.

Posterior Sampling in Pictorial Structures

- Problem Formulation:

$$p(L|I,\theta) \propto \prod_{i=1}^{n} p(I|l_i,u_i) \prod_{(v_i,v_j)\in E} p(l_i,l_j|c_{ij})$$

- Key Elements:
  - $p(I|l_i,u_i)$: Likelihood of observing I given part i at l_i.
  - $p(l_i,l_j|c_{ij})$: Probability of relative locations of connected parts.
  - Objective: Sample configurations L from this posterior distribution.
- Approach:
  - Algorithm similar to energy minimization, but uses probabilities and summations.
  - Based on a tree structure of dependencies between parts.

Computing and Sampling in Posterior Distribution

- Root Sampling:

$$p(l_r|I,\theta) \propto p(I|l_r,u_r) \prod_{v_c \in C_r} S_c(l_r)$$

- S Function for Nodes:

$$S_j(l_i) \propto \sum_{l_j} p(I|l_j,u_j) p(l_i,l_j|c_{ij}) \prod_{v_c \in C_j} S_c(l_j)$$

- Child Node Sampling:

$$p(l_j|l_i,I,\theta) \propto p(I|l_j,u_j) p(l_i,l_j|c_{ij}) \prod_{v_c \in C_j} S_c(l_j)$$

- Complexity:
  - Time: O(h' * n) using Gaussian convolution in transformed space.
  - Efficient computation via separable Gaussian filter and discrete grid.

Iconic Models

- Parts Location: Defined by (x, y) in a two-dimensional pose space.
- Iconic Representation:
  - Based on Gaussian derivative filters responses.
  - Image patch represented as a 27-dimensional vector (using scales $\sigma_1 = 1, \sigma_2 = 2, \sigma_3 = 4$).
  - Iconic index insensitive to lighting, scale changes, and deformations.
- Appearance Model:

$$p(I|l_i, u_i) \propto N(\alpha(l_i), \mu_i, \Sigma_i)$$

  - $\alpha(l_i)$: Iconic index at location l_i.
  - $u_i = (\mu_i, \Sigma_i)$: Gaussian model parameters for each part.

Iconic Models - Spatial Relations and Experiments

- Spatial Relations:

$$p(l_i, l_j | c_{ij}) = N(l_i - l_j, s_{ij}, \Sigma_{ij})$$

    - Connections modeled by springs between parts.
    - $c_{ij} = (s_{ij}, \Sigma_{ij})$: Parameters for each connection.
    - Deformations modeled by Gaussian distribution with full covariance matrix.
- Transformation for Algorithm Compatibility:

$$T_{ij}(l_i) = U_{ij}^T (l_i - s_{ij}), \quad T_{ji}(l_j) = U_{ij}^T (l_j)$$

    - U_{ij}: From SVD of Σ_{ij}.
- Experiments:
    - ML estimation for model training.
    - Models tested on face detection with varying conditions.
    - Demonstrated robustness in occlusion and multiple face detection.

Results

Figure: Three examples from the first training set showing the locations of the labeled features and the structure of the learned model.

Results

Figure: Matching results demonstrating the model's effectiveness in detecting facial features.

Results

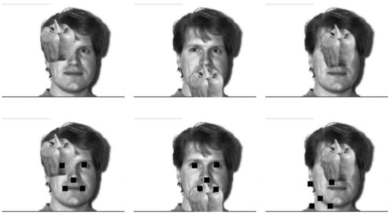

Figure: Matching results on occluded faces, illustrating the model's robustness in handling partial occlusions.

Results

Figure: Matching results on an image with multiple faces, showcasing the model's capability in complex scenarios.

Articulated Models

Definition

- (x_i', y_i') and (x_j', y_j') are the positions of the joints.

$$\begin{aligned} p(l_i, l_j \mid c_{ij}) = \mathcal{N}(x_i' - x_j', 0, \sigma_x^2) \\ \mathcal{N}(y_i' - y_j', 0, \sigma_y^2) \\ \mathcal{N}(s_i' - s_j', 0, \sigma_s^2) \\ \mathcal{M}(\theta_i' - \theta_j', 0, \sigma_\theta^2) \end{aligned} \tag{7}$$

In the right form :

$$p(l_i, l_j \mid c_{ij}) \propto \mathcal{N}(T_{ji}(l_j) - T_{ij}(l_i), 0, D_{ij}) \tag{8}$$

Results

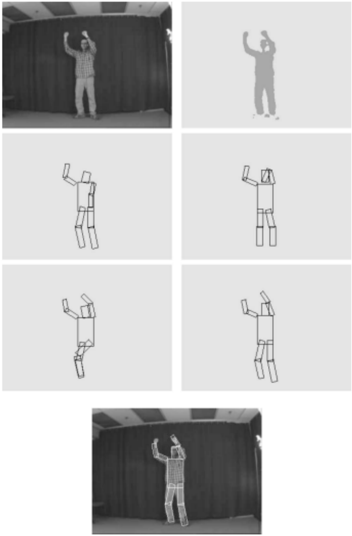

Figure: Input image, binary image, random samples from the posterior distribution of configurations, and best result selected using the Chamfer distance

Results

Variety of Poses

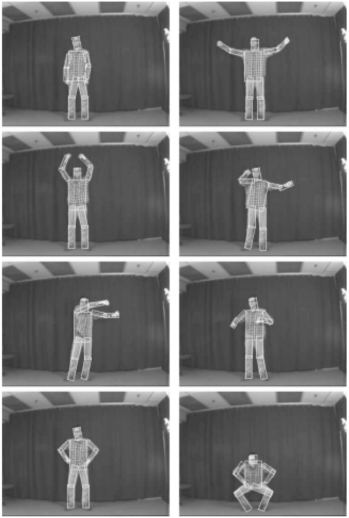

Figure: Matching results

Results

Drawbacks

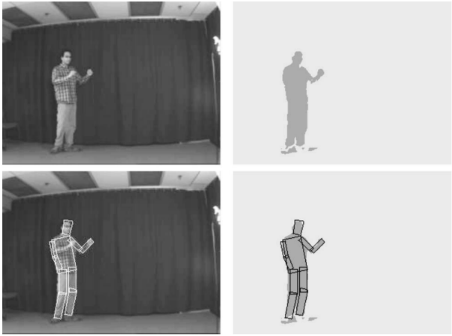

Figure: In this case, the binary image doesn't provide enough information to estimate the position of one arm.

Results

Drawbacks

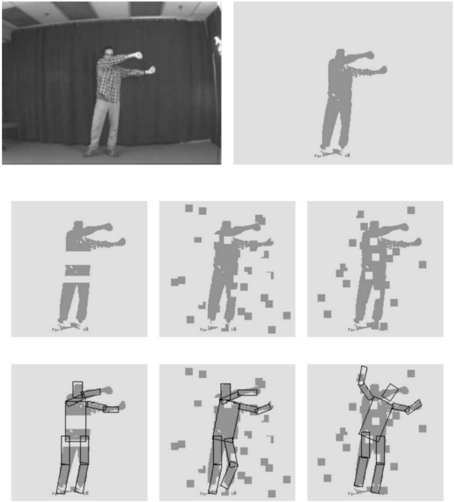

Figure: Matching results on corrupted images.

Conclusions

- Main Contributions
  - Introduce Efficient Algorithms
  - Use of statistical sampling
  - Use of stastistical formulation from labeled example images
- Models based on the pictorial structure representation Fischler and Elschlager (1973)
- Statistical framework for representing visual appearance

References

- Felzenszwalb, P.F., Huttenlocher, D.P. (2005). Pictorial Structures for Object Recognition. International Journal of Computer Vision, 61, 55–79. https://doi.org/10.1023/B:VISI.0000042934.15159.49

Appendix

Learning Model Parameters - Estimating the Appearance Parameters
From equation (6) we get

$$u^* = \arg\max_u \prod_{k=1}^{m} p\left(I^k \mid L^k, u\right),$$

The likelihood of seeing image I^k, given the configuration L^k for the object is given by by the product of the likelihoods

$$u^* = \arg\max_u \prod_{k=1}^{m} \prod_{i=1}^{n} p\left(I^k \mid l_i^k, u_i\right) = \arg\max_u \prod_{i=1}^{n} \prod_{k=1}^{m} p\left(I^k \mid l_i^k, u_i\right).$$

Looking at the right hand side we see that to find u^* we can independently solve for the u_i^*,

$$u_i^* = \arg\max_{u_i} \prod_{k=1}^{m} p\left(I^k \mid l_i^k, u_i\right).$$

This is exactly the ML estimate of the appearance parameters for part v_i, given independent examples $\left\{\left(I^1, l_i^1\right), \ldots, (I^m, l_i^m)\right\}$.

Generalized Distance Transforms

- Traditional Distance Transforms:

$$D_B(x) = \min_{y \in B} \rho(x, y)$$

  - $\rho(x, y)$: Distance measure on grid G.
  - $B \subseteq G$: Set of points.
  - $D_B(x)$: Distance to nearest point in B.
- Generalized Form:

$$D_f(x) = \min_{y \in G}(\rho(x, y) + f(y))$$

  - $f(y)$: Arbitrary function over grid G.
  - Optimizes distance and function value.
- Application to Dynamic Programming:

$$B_j(l_i) = D_f(T_{ij}(l_i))$$

  - $B_j(l_i)$: Computed using generalized distance transform.
  - $T_{ij}(l_i)$: Transformation in the grid.
  - Efficient computation under Mahalanobis distance.

Computing the S Functions

- S Function Formulation:

$$S_j(l_i) \propto \sum_{l_j} N(T_{ij}(l_i) - T_{ji}(l_j), 0, D_{ij}) p(I|l_j, u_j) \prod_{v_c \in C_j} S_c(l_j)$$

  - Gaussian convolution in transformed space.
  - T_{ij} and T_{ji}: Transformations.
  - D_{ij}: Covariance in Gaussian filter.

- Convolution and Efficiency:

$$S_j(l_i) \propto (F \otimes f)(T_{ij}(l_i))$$

  - F: Gaussian filter.
  - $\otimes$: Convolution operator.
  - Efficient computation on a discrete grid.

- Algorithm Complexity:
  - Time: O(h' * n).
  - Efficient for large models and configurations.

Questions