Applying Pictorial Structures to Pose Recognition

Vincent Petrella
McGill University

Abstract

We take on the problem of human pose recognition from images using pictorial structures. A probabilistic approach to estimating a pictorial structure model is first developed and evaluated. The model is then used to find the best match of a structure against a test image. Based on Qualitative results, we discussed the validity of the algorithm, as well as extensions to potentially improve the method.

1. Introduction

Recognizing objects from images is a long lasting unsolved problem for computer vision researchers. Its possible applications are numerous: Equipping Artificial Intelligence with the ability to distinguish objects and their states, automating the analysis of medical images and many others potential real-life applications. For many years, researchers have been trying to find solutions to subsets of the problem, rather than to find a general method. *Pictorial Structures for Object Recognition* [1] is one of them. Under the assumptions that objects are composed of various parts spatially arranged in predictable fashions, pictorial structures allow for a representation that leverages such a peculiar property. In this paper, we will study their application to the problem of finding people's pose in static images. The problem of pose estimation is of importance in application such as security, where suspicious behavior may be detected from a person's pose. Other applications of pose estimation can also apply to robotics, and human-computer interaction AIs.

2. Pictorial Structures

As previously mentioned, pictorial structures represent objects composed of multiple connected parts in various configurations. Each part of the object can be captured in a measurement (image) and will generate a particular type of feature. An exact match for a part in an image is therefore an image area exhibiting this exact feature. Likewise, the best match for a pictorial structure, is when each part and their estimated image area best exhibit this "visual" similarity, and in which each connected parts' spatial relations best satisfy some "soft" constraint.

2.1. Energy of a Pictorial Structure

Let a graph $G = \{V, E\}$ represent a Pictorial Structure (PS). V the set of parts, and E the set of connections between parts (not necessarily fully connected). An instance of PS is summarized into $L = \{l_1, l_2, \dots, l_n\}$ where each l_i is the configuration of part v_i.
Following our discussion, an intuitive requirement for a best match of particular structure, is one that minimizes the following expression as defined in [1]:

$$E(L) = \sum_i m_i(l_i) + \sum_{(v_i, v_j) \in E} d_{i,j}(l_i, l_j) \qquad (1)$$

In which $m_i(l_i)$ is a measure of "visual" mismatch if v_i is in configuration l_i, and $d_{i,j}(l_i, l_j)$ is the deformation cost of edge (v_i, v_j). This last term echoes to the "soft" constraint on each connected parts relative configuration. Intuitively, this minimization enforces a globally low visual mismatch from all parts and favors spatial arrangements close to the model specification. A PS instance that best matches an image is thus: $L^* = argmin(E(L))$.

While this definition is simple and intuitive, several questions remain un-answered. What are some good methods to optimally minimize this energy function? What procedure can we employ to learn the Pictorial Structure model (giving m_i and $d_{i,j}$)? To answer these non-trivial questions, we will consider the statistical framework described in [1].

2.2. A Statistical Approach

Let us define the parameter set for a Pictorial Structure model as θ. If I denotes an image, and as previously mentioned, L an instance of a PS, then the distribution $p(I|L, \theta)$ represents the imaging process (probability of generating and image given a configuration and the model), while $p(L|\theta)$ is the prior on possible configurations given a model. Intuitively, $p(I|L, \theta)$ echoes to the "visual" match

of L, and $p(L|\theta)$ to the constraint on configuration given the model. We hence obtain the probability that L generated the image given the model using Bayes rule:

$$p(L|I,\theta) \propto p(I|L,\theta)\, p(L|\theta) \qquad (2)$$

This posterior formulation opens the way for estimation using Maximum a Posteriori as well as sampling techniques. Furthermore, if both terms on the right hand side can be modelled in a meaningful way, then model estimation becomes straightforward. We now turn to the issue of defining those terms.

A Pictorial Structure model θ is defined as $\theta = \{u, E, c\}$. $u = \{u_i, \dots, u_n\}$ are the "visual", or "appearance" parameters for part v_i. E defines which part is connected with which, and $c = \{c_{i,j}\}_{(i,j)\in E}$ are the parameters for each of these connections. Observe that $p(I|L,\theta)$, the image process, will solely depend on the appearance parameter of each part in L (since they describe how a part generates an visual image). We can thus write $p(I|L,\theta) = p(I|L,u)$. Provided that parts do not overlap, a decent approximation of this likelihood, as defined in [1], would be the product of individual probabilities of generating image I given a part in configuration l_i:

$$p(I|L,\theta) \propto \prod_i p(I|l_i,u_i)$$

Next, we need an expression of $p(L|\theta)$. We previously mentioned, we can think of this prior as carrying knowledge about the constraint our Pictorial Structure enforces on part configuration. More precisely, given our model parameters, these constraints intuitively only apply to E and c, which are the only parameters defining configurations. A good approximation for this prior is then:

$$p(L|\theta) = \prod_{(v_i,v_j)\in E} p(l_i,l_j|c_{i,j})$$

Where $p(l_i,l_j|c_{i,j})$ encodes the likelihood of having two parts in each configuration given the model. Inserting the above in (2) gives the following formulation (as in [1]) for the posterior distribution:

$$p(L|I,\theta) = \prod_i p(I|l_i,u_i) \prod_{(v_i,v_j)\in E} p(l_i,l_j|c_{i,j}) \qquad (3)$$

Finally, a good intuitive understanding of the above quantities should convince the reader that minimizing the negative Logarithm of $p(L|I,\theta)$ yields to the same formulation as (1). We are hence solving a similar problem.

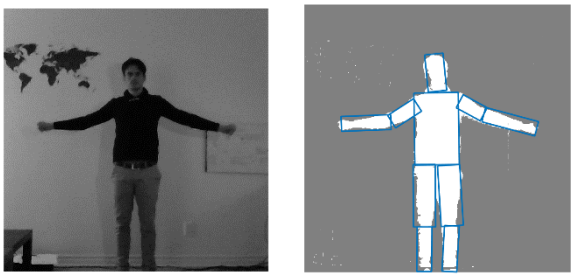

Figure 1: A human body image (left) and its background-subtracted annotated counterpart (right).

3. Pictorial Structures for Pose Recognition

The human body exhibits an arrangement very similar to those of Pictorial Structures. It is divided in limbs, which, for example in Computer Animation, can be thought of as rigid bodies connected with rotating joints [2]. Each joint is differently constrained in its rotation abilities as well as its position on the body. This is a perfect occasion to apply Pictorial Structures. In this section, we proceed in defining the model parameters θ for a human body, while slightly departing from the model described in [1]. The Pictorial Structure used in this work contains 10 parts represented by rectangular shapes, and the set E of connections is restricted to the 9 joints connecting each the 10 limbs. An instance of a pose estimate is hence a collection $L = \{l_1, l_2, \dots, l_{10}\}$. The configuration for each part is as follows: $l = \{X, s, \alpha\}$ with X the position of the center of the rectangle, s the length of the rectangle (varies with foreshortening), and α its orientation. The width of the rectangle is fixed in our implementation and is related to the diameter of the body part.

3.1. Appearance Parameters

The algorithm described in this paper takes its input data from binary images. Given an image of a person in any setting, the background is removed (using simple background subtraction technics) and is then thresholded to obtain a binary foreground/background map (see Figure 1). The appearance model parameters is thus related to the probability of a part to contain foreground pixels over background pixel. As an intuitive convention, pixels belonging to a body are foreground pixels.

As in [1], we divide the area enclosing a part into an inside area (of the size of the limb) and an outside area enclosing the surrounding of the limb. For a well-placed part, the inside area should be filled with mainly foreground pixels (as it encloses the limb) with probability q_1, while its direct outside should be mainly background pixels with probability q_2. Therefore the probability that l_i generates

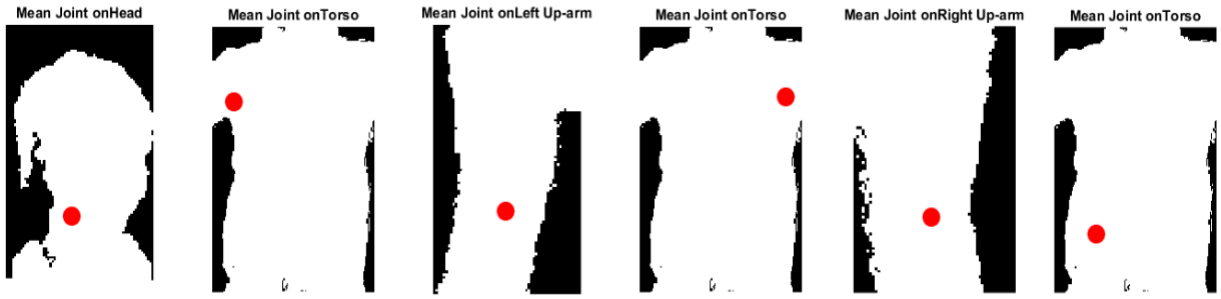

Figure 2: Mean joint position on parent part for the 6 first connections.

the measured pixels is:

$$p(I|L_i, u_i) = q_1^{count_1}(1-q_1)^{|area_1|-count_1} \times q_2^{count_2}(1-q_2)^{|area_2|-count_2}$$

where $count_1$ and $count_2$ counts the number of foreground pixels in $area_1$ and background pixels in $area_2$ respectively.

The Maximum Likelihood estimate of q_1 and q_2 for each part is then straightforward.

3.2. Joints Parameters

For simplicity, we know slightly depart from the [1] when representing parameters. As described above, our model contains 10 different parts ('Head', 'Torso', 'Left Up-arm', 'Left Low-arm', 'Right Up-arm', 'Right Low-arm', 'Left Up-leg', 'Left Low-leg', 'Right Up-leg' and 'Right Low-leg'), and 9 joints between these parts expressed in the same order. To model all the constraints on the joints, we only need 3 parameters in $c_{i,j}$: The position of the joint X_{ij}, the relative length Δs and relative angle $\Delta\alpha$ between the two parts.

Assuming that these parameters are all independent, finding an expression for $p(l_i, l_j|c_{i,j})$ reduces to finding a distribution for each of them. We model each of them using Normal distributions:

$$N\left(X, \mu X_{ij}, \sigma_{X_{ij}}^2\right)$$
$$N(s, \mu s, \sigma_s^2)$$
$$N(\alpha, \mu\alpha, \sigma_\alpha^2)$$

3.3. Implementation of the learning algorithm

Each of these distributions parameters are learned from the labelled training images by working directly with parts information (Saved from user input). For each part type, the parameters are computed for each training image then aggregated into a Mean and Covariance using MATLAB's built-in functions.

As we will see in the next section, our matching algorithm works by generating multiple instances in a recursive manner. Therefore, the information about the joint position X_{ij} cannot be expressed in image coordinates, as it is relative to the parent part's configuration. To obtain μX_{ij} and $\sigma_{X_{ij}}^2$ from examples, each joint location is expressed in its parent normalized (as width and length may vary) coordinate system (see Figure 2 and 3). We used an external MATLAB function to convert the covariance to the nearest positive semi-definite matrix in case it was not [3].

Please refer to the MATLAB code provided with this paper to obtain more details on the implementation.

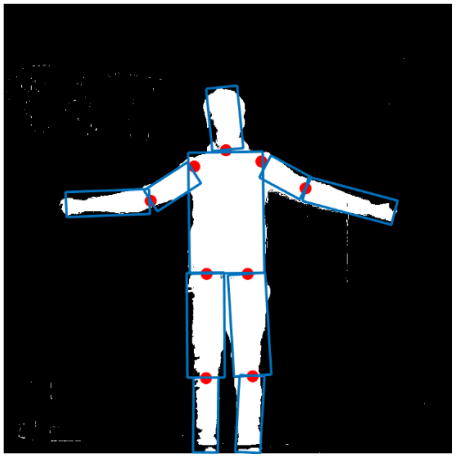

Figure 3: Mean Joint locations in image coordinates transformed using the labelled parts information.

4. The Matching Algorithm

Now that we have found a model to express the Pictorial Structure that fits best our training data, we can put it to use on novel, unlabeled images. To find a pose estimate using

our model, we will derive a procedure to maximize $p(L|I,\theta)$ in a different fashion as in [1]. For simplicity, we will only work on estimating the pose knowing the position of the person's head in the image. Furthermore, we use a conceptually simpler method to find an appropriate match given the image and the prior on configurations.

4.1. Sampling from the configuration distribution

Knowing the probability distribution for each connection parameters, it is straightforward to build a Pictorial Structure in a generative manner. Given a head part configuration, we can generate each part by sampling from the joint parameters distribution of its parent. This can be thought of as a grid search over the space of possible configurations. This space of possible configurations is notably high, as a configuration $l_i = \{X, s, \alpha\}$ takes values in large discrete set. However, the information gained via model estimation (as well as encoded by design, like the connections between parts) allows to drastically reduce the size of this search space.

A likely configuration is however not necessarily a good configuration. Once an instance is sampled, it must be ranked against all others according to the first term of the posterior. Recall, the "matchness" term in our maximization problem, associated to the test image: $\prod_i p(I|l_i, u_i)$. To pick a best match for our Pictorial Structure, we require this quantity to be maximized. In our implementation, we actually compare the logarithm of this likelihood, to avoid floating point vanishing to 0 issues.

The pseudo-code for this algorithm is as follows:

```
BestScore ← -Inf
For N_SAMPLES do
   Sample ← GenerateSample (Head, ModelParams)
   LogLikelihood ←ComputeAppearanceLikelihood(sample)
   If LogLikelihood > BestScore
      BestScore ← LogLikelihood
      BestSample ← sample
   Endif
Endfor
Return BestSample
```

4.2. Drawbacks of the simple sampling method

Indeed, this simple method, while giving encouraging results (see Figure 4), has many drawbacks and shortcomings. The computational cost of generating many different configurations might be intractable for very exhaustive search.

Furthermore, maximizing the likelihood of each part does not necessarily yield to the best fit. Indeed, if two parts end up overlapping on foreground pixels (for example, arm and torso) the likelihood of the configuration might be higher than that of the actual best match (see figure 5). To tackle this issue, a modification should be made to pick the best matching sample. While still sampling preferred part location by maximizing the appearance likelihood, we need another metric to validate several "good matches". The authors of [1] suggest using the Chamfer distance [4]. However, by lack of time, such an implementation was not possible.

Another important shortcoming of this technique, is that the evidence used by the appearance parameter for parts carries relatively low evidence of actual appearance. In this technique, color and other essential features of various body parts (such as the face, hands, particular clothing) is

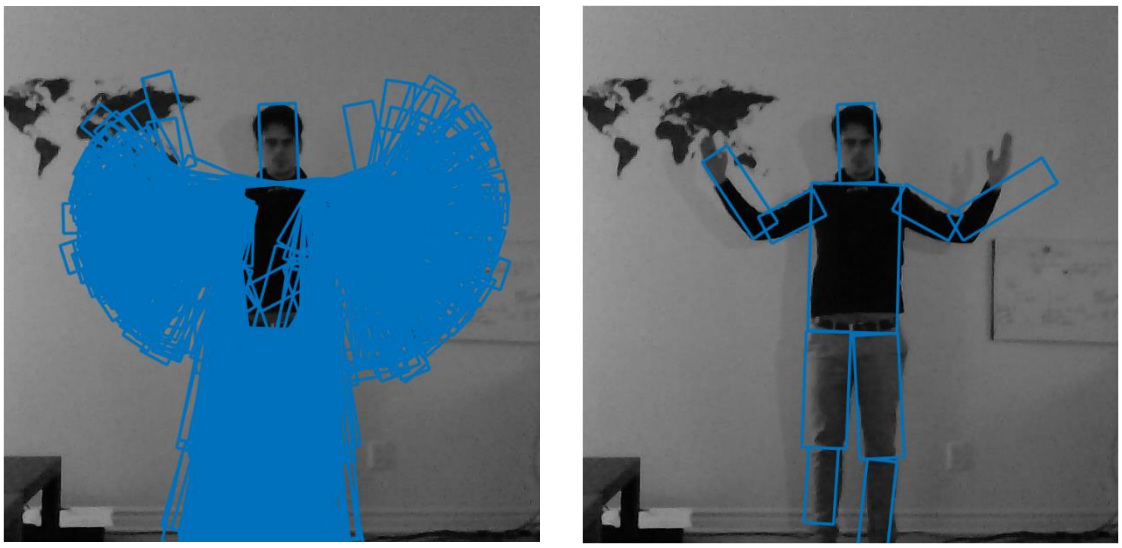

Figure 4: All sampled configurations for a matching request in one image (Left) and the final decision based on ML for appearance the model (Right). As we can see, since limbs at the extremities (legs and arms) exhibit more variations in the training set, the sampling strategy explores many more rotations for these parts, giving rise to an angel-like shape

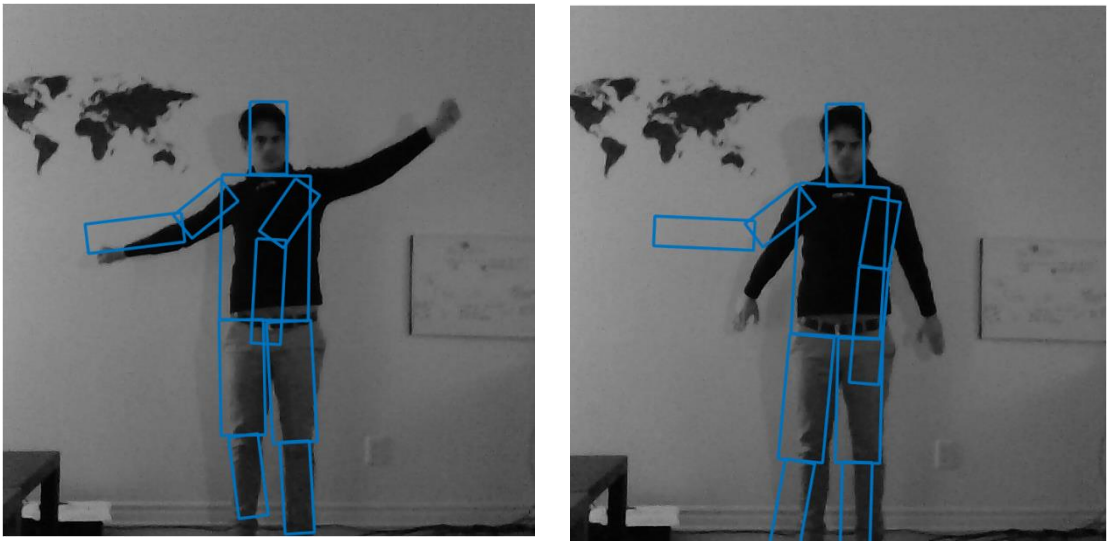

Figure 5: Very bad match configuration end up being chosen as best match because of the issue of over-counting the evidence. If two parts overlap in the sampled configuration, the foreground pixel counts will be registered to both. Here, in those two examples, the left arm along the body line is considered a much better configuration because all of its pixels end up being foreground, whereas samples matching the pose are discarded because the parts are not tight enough and will include a lot of background pixels.

completely left off of the available source of data. Using a stronger and more sophisticated appearance model would help prevent artifacts of appearance likelihood estimation.

5. Conclusion

We have seen that the Pictorial Structure model is a good model for specific objects recognition task. The articulated nature of the human body makes it particularly good fit. We have explored the statistical framework derived in [1] to approach the problem of finding a good match to the Pictorial Structures model, as well as to find a good parametrization of the model. The simple sampling method that we experimented with gave promising results, but suffers from an incomplete and ambiguous notion of “best match”. Further improvements could be made to avoid over-counting evidence, or to try leveraging extra information retained in color and other feature details that are discarded by the binary thresholding utilized in the appearance model.

References

[1] Felzenszwalb, Pedro F., and Daniel P. Huttenlocher. "Pictorial structures for object recognition." International Journal of Computer Vision 61.1 (2005): 55-79.

[2] Gomes J., Velho L., Costa Sousa M., Computer Graphics: Theory and Practice, 2012.

[3] John D'Errico, Mathworks FileExchange website: http://www.mathworks.com/matlabcentral/fileexchange/42885-nearestspd

[4] G. Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence,10(6):849{865, November 1988.

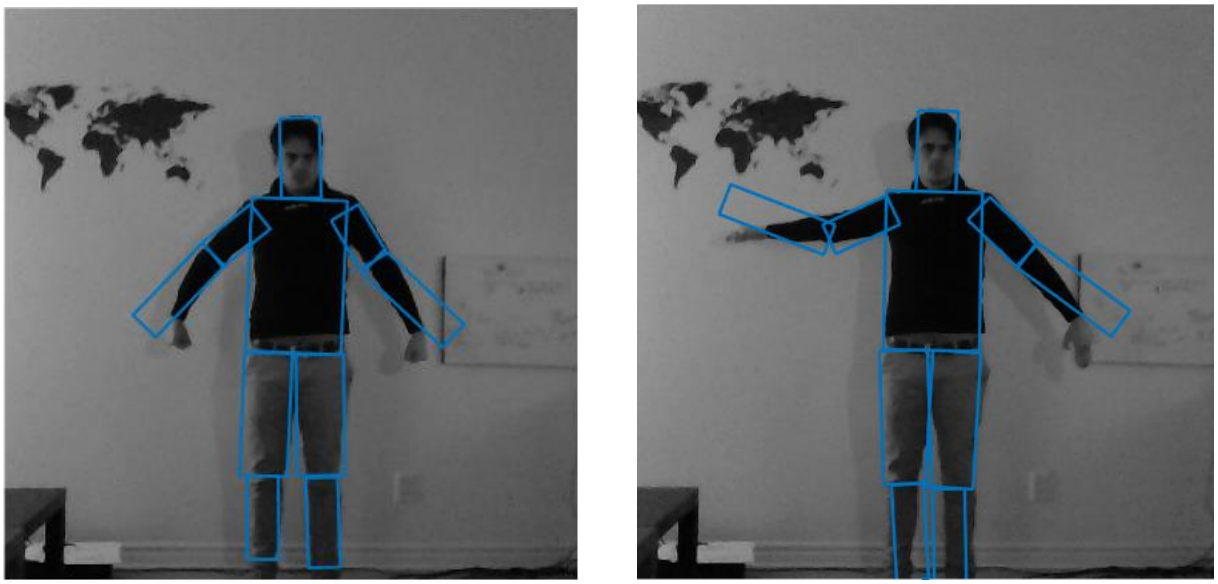

Figure 6: Various results obtained via the sampling method.

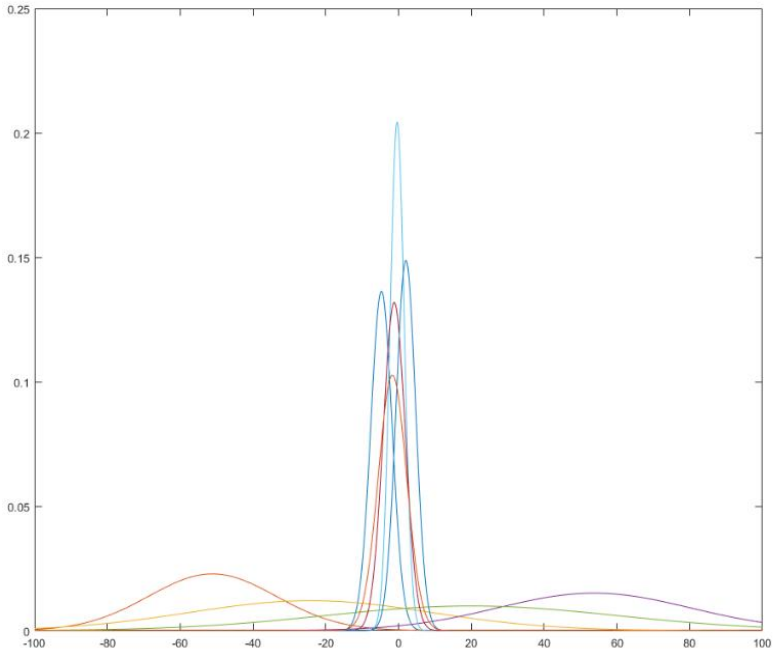

Figure 6: Probability distributions over the Angle differences
for each connection type.