2D Pose Estimation Using Active Shape Models and Learned Entropy Field Approximations

Ian Dewancker
Department of Computer Science
University of British Columbia
Image Understanding II Project, April 23 2012
idewanck@cs.ubc.ca

Abstract

This report describes the implementation of a framework for exploring full-body, generative 2D pose estimation approach in single, broadcast quality frames. A dataset for training is manually constructed from image frames taken from NBA footage. Poses can be generated using an active shape model consisting of a simple linear combination requiring only a small number of parameters. To guide the search for an optimal pose, an energy function is proposed that compares image entropy with a learned entropy field estimator, computed using the current pose estimate. Current results are presented and future work or exploration is discussed.

1 Introduction

Pose estimation and understanding human body configurations in images is an important problem in computer vision. A large number of applications ranging from surveillance, human-computer iteration and motion analysis. This problem is challenging because of the high dimensional search space of body poses, the presence of ambiguous poses, and the need to identify the human body in each image. Determining pose in video sequences is likely a key stepping stone towards solving the more difficult problem of general action recognition.

This report outlines a generative framework for 2D pose search in NBA video frames. By 2D pose we mean that the joints are only estimated in image space, and not estimated in 3D. To create a representative model of the pose space, an application was developed to annotate images quickly to create a low resolution training set of about 150 poses. Following the core ideas behind active shape models, principle component analysis is applied to a mean adjusted set of training poses to find the modes of greatest variation. These modes form the basis of a simple linear model that can be used to generate poses. A bounding box, centered on the each players chest is assumed to be given, that is, the joint positions are represented as scaled offsets from the chest point.

Next, a method for computing the likelihood of the image data is presented. Image entropy is explored as a texture similarity measure. Two regression methods, neural networks and radial basis functions are explored to learn a function that takes the current 2D pose and generates a 2D entropy field (an estimate of the expected entropy signature for a given pose) that can then be directly compared with the entropy of the image. Further steps are discussed to improve this entropy estimation, speed up the computation and finally perform the search using a prior on the pose space with a suitable stochastic method.
2 Related Work

There has been a fair amount of research effort spent investigating this problem and as a result, a
diverse array of approaches exist. One approach formulates the problem as a regression between 2D
silhouette data and 3D poses of joints using radial basis functions [2]. The drawback with this ap-
proach is that it requires very clean segmentation between the human (foreground) and background,
which is very challenging when dealing with real scenes. Another recent approach uses a histogram
of oriented gradients (HOG) approach to detect limbs and body parts in images and then estimate a
likely pose [5]. One drawback with this approach is that it requires fairly high resolution images of
the persons body (500X500 pixels) to successfully detect the limbs and return an estimated pose.

Other ideas have come from looking at sequences of frames and computing a motion signature from
the optical flow of the image and then matching this to a database of templates [11]. This template
database contains known poses for each motion signature and so the estimated pose can be matched
from these examples. Finally, detection and tracking sport players and pedestrians has been well
studied with fairly robust solutions in existence [6]

3 Problem Definition

At a high level, we can view the task of pose estimation as an inference problem, where we are
trying to estimate the most likely pose $X$ (which here denotes a vector representing the 2D pose of a
player) given the image data $I$ (which here denotes the pixel data within a provided bounding box in
a single video frame). We can view the pose estimation problem as finding a suitable (most likely)
pose over the posterior as shown in equation 2

For this framework, we present the following definition for the pose space. Poses exist as elements
of a 32 dimensional space, where each are represented as 16 pairs of normalized 2D co-ordinates
for 16 joints as shown in the figure below. The center chest point (highlighted in red in the figure)
represents the point $(0,0)$ and every other point is defined as an offset from this point, for example
the head could be located at $(0,-0.2)$. A scaling relative to the height of the provided bounding box
is performed to normalize the image co-ordinates of the joints, that is, every joint location offset in
pixel co-ordinates is divided by the height of the bounding box.

$$X = \begin{bmatrix}
  \text{head}_u \\
  \text{head}_v \\
  \vdots \\
  \text{rfoot}_u \\
  \text{rfoot}_v
\end{bmatrix}, \quad I = \text{Image Data} \quad (1)$$

$$\arg\max_X P(X \mid I) = \arg\max_X \alpha P(I \mid X) \cdot P(X) \quad (2)$$

Figure 1: 2D Pose Vector and Problem Definition

We approach the problem with a generative model and look to design a solution that samples hy-
potheses from the pose space and evaluates their likelihood given the data present $I$. An alternative
approach in machine learning is to use discriminative models which are designed to directly map
image information and features computed from the data $I$ into likely pose estimates [1]. Here we
pursue a generative approach and outline steps towards a full solution. There are two important steps
for a generative approach, creating a model to generate poses from the pose space and the function
method of evaluating how well that pose fits the data (likelihood).
4 Training Data Generation

To better understand and represent the pose space, an application was developed to generate example poses from frames of data. The application was written in python using a game library and allowed the user to quickly click and generate skeletons to be fit to the image. The user could scale skeletons and manipulate joint positions with the mouse.

![Figure 2: Screenshot of Pose Training Application](image)

The training examples were saved in plaintext format to a file. Players were chosen that were relatively isolated in the frame, that is, not occluded by other players, however self-occlusion was present. Approximately 150 poses were created, but more could easily be generated.

5 Active Shape Models

A well studied approach for modeling shape variation comes from active shape models [10]. A parameterized linear model can be estimated by using principle component analysis (PCA) to find the main axes or principle components of the data cloud. This can be performed fairly simply, by taking a matrix $M$ which consists of $N$ poses collected from training, and offsetting them by the mean training vector as shown in equation 3 and 4.

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i$$  \hspace{1cm} (3)

$$M = \left[ (x_1 - \bar{x}) \ (x_2 - \bar{x}) \ \cdots \ (x_N - \bar{x}) \right], \quad \Phi = ([U], \text{SVD}(M))$$  \hspace{1cm} (4)

$$X = \bar{x} + \Phi b$$  \hspace{1cm} (5)

The figure below shows the mean pose shape offset by the first 3 modes. The top row of the figure shows the mean pose modified by the 1st mode only, the middle by the second and the bottom by the third. Intuitively one can understand that these modes adjust the limbs and posture of the skeletons along dimensions of large variance as learned from the training set.
With a shape model complete and capable of generating poses the next step was to investigate possible methods for determining the likelihood of an image given a pose, that is a method of computing $P(I \mid X)$. One idea that has been explored is that of edge matching using an algorithm known as chamfer matching, however edge data is often noisy, and this leads to poses being drawn to areas dense with edges. Another idea is creating a colour model for players and poses from the training data so that a pose could be matched to the image, however this is also fragile, since players have very different colours for their skins and jerseys and colour is very sensitive to illumination changes. What was sought was something that captured slightly more information than edges, but slightly less sensitive to player and luminance changes than colour.

### 6.1 Image Entropy

Entropy is a statistical measure of randomness that can be used to characterize the texture of an image. It is a measure of the variability within a set of measurements and if we take the measurements to be grayscale intensities in a 9×9 window around each pixel, we can compute the entropy of the gray scale histogram after binning the values into 12 different bins.

Intuitively, players should exhibit high entropy patterns around the edges of their limbs and the text on their jerseys and lower entropy patterns on the interior areas of their skin and non-text jersey areas as shown below in the figure.

$$H(Y) = - \sum_{i=1}^{N} p(y_i) \log_2 p(y_i)$$ (6)
6.2 Learning the Entropy Field Approximation

What was required now was a method to generate an entropy field, which we will define to be a 2D scalar field given a pose vector, that could be compared to against the entropy of the actual image. In other words, what should the entropy image look like for a given skeleton pose? Two common regression methods were experimented with having varying degrees of success to estimate the entropy field $H_{u,v}(X)$, note we have augmented the 32 dimensional vector $X$ with 2 more dimensions for to capture the dimensions of the 2D entropy field.

The first regression model was a single layered neural network consisting of 10000 internal nodes. A neural network is a parametric regression approach that learns a set of weights to combine inputs into internal nodes and internal nodes into outputs using activation functions as shown in equation 7 [9]. Several parameters were modified and experimented with including the number of internal nodes and type of activation function. The best results for the example are shown below in the figure:

$$H_{u,v}(X) = \sigma \left( \sum_{j=0}^{10000} w_{kj}^{(2)} h \left( \sum_{i=0}^{34} w_{ji}^{(1)} x_i \right) \right)$$ (7)

Radial basis functions, or RBFs are also universal function approximators, however they use a non-parametric approach, storing centers taken from the training data to and use them to interpolate new values [8] [7]. We used Gaussian basis functions and took 5000 centers to produce the results shown in the figure below. As you can see, it does a better job than the neural net, but it is still not great.

$$H_{u,v}(X) = \sum_{i=1}^{5000} w_i \phi(\|X - c_i\|)$$ (8)

Figure 5: From left to right: Skeleton Pose, Ground Truth Entropy, ANN Entropy, RBF Entropy

While some good initial experimentation took place, the problem with the framework was that the function was complicated and computation time to train these systems grew increasingly large. With the RBF estimate, structure is clearly present, however for the method to work well, it should begin to very closely estimate the ground truth so that a simple comparison method could be run on the estimate and the actual image.

Alternatively, what could have been more computationally effective would have been to use an array of neural nets, at representative co-ordinates in the entropy field and train them to produce expected values and then interpolate other values in the field given these grid nodes as reference values, instead of trying to regress for the entire field. Unfortunately time ran out to implement and test this idea.
7 Conclusions and Future Work

While work on the image likelihood function proved promising, a prior on the poses would still be required to help restrain the search to poses that are close to those in the training data. A typical approach to this problem is to use a Gaussian mixture model. Similar to RBFs, a Gaussian Mixture Model (GMM) is a weighted sum of M Gaussian densities as given by the equation parameters are estimated from training data using the iterative Expectation-Maximization (EM) algorithm [9]

\[
P(X) = \sum_{i=1}^{M} w_i g(X|\mu_i, \Sigma_i) \quad (9)
\]

\[
g(X|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}} \exp \left\{-\frac{1}{2}(X - \mu_i)'\Sigma_i^{-1}(X - \mu_i)\right\} \quad (10)
\]

This would provide the full definition of the probability distribution function and from there an optimization search could be performed. Due to the fact that the gradient would be difficult or impossible to compute for this function, it seems likely that a stochastic approach like simulated annealing would work well for this problem. Neighbours could be generated by concatenating random offsets from Gaussian distributions and the temperature scheduling could be adjusted with experimentation.

To really evaluate any approach, much more training data would have to be produced and so the efforts of this project focused more on experimenting with potentially useful approaches. Ideally a full evaluation of a solution could take place with reports on the accuracy of estimated poses compared with the ground truth. It is unfortunate that I ran out of time to fully evaluate a method, but the preliminary results were promising and the framework is written now for further experimentation.

A more intelligent approach could also be taken with the active shape models as well. Instead of simply finding the mean of the entire training set, some unsupervised clustering could be performed beforehand and then k shape models could be constructed. This could potentially retain more important information about the variance in the pose clusters associated with various actions (jumping, running, shooting etc.). Included after the report is all the source code I wrote for the project.

References