Real-time Inference of 3D Human Poses by Assembling Local Patches

Kota Hara
Intelligent Systems Lab, SECOM CO., LTD.
Tokyo, 181-8528 Japan
kot-hara@secom.co.jp

Abstract

This paper presents a novel patch-based approach for 3D human body pose estimation from a static silhouette image. Our approach uses a database which contains example patches extracted from different parts of images rendered using a 3D human body model in various poses. Each example patch is represented as a shape context histogram and contains pose parameters of the model that is used to render the image. At the estimation step, example patches in the database that have a similar shape context to patches extracted from the input silhouette image are rapidly retrieved using a modified Locality-Sensitive Hashing algorithm. The pose parameters are then estimated by a probabilistic Hough voting scheme in a hierarchical manner. Our approach is flexible and needs a small number of examples since combinations of local patches can be used to identify previously unseen entire poses. Thus, it is not necessary for every possible pose to be stored in the database. This property significantly reduces computation time. Experiments have shown that our proposed method can handle a variety of unseen articulations and output accurate pose estimations in real time.

1. Introduction

Estimating human body poses from still images is one of the most challenging tasks in computer vision. In particular, being able to cover a wide variety of poses is extremely difficult due to highly articulated human bodies.

Example-based pose estimation algorithms exploit the example data during recognition to find the example that is most similar to the input image. The algorithm outputs pose parameters that are attached to this example. Although the example-based approach is attractive due to its simplicity and performance, the high dimensionality of the human pose articulation makes the number of examples needed to cover a variety of poses infeasible.

In this paper, we address the aforementioned issue of example based algorithms. Instead of using one example for each possible pose as most existing example-based pose estimation algorithms do, we use an image patch locally extracted from a pose image as one example. Since all poses can be seen as a combination of the poses of individual parts of the human body, we reconstruct poses by assembling local patches. This approach dramatically reduces the number of necessary examples allowing for accurate and fast estimation.

In order to create patch examples, patches are extracted from different parts of images rendered using a 3D human model in a variety of poses and stored in a database. Each example patch is represented as the shape context [1] and contains the pose parameters used to generate it. During recognition, several patches are extracted from the input silhouette and similar example patches are retrieved for each extracted patch from the database. The pose parameters are then estimated by a probabilistic Hough voting scheme [2, 6].

In a general probabilistic Hough voting scheme, each example votes in the parameter space and the point in the space which gets the most votes around it is found using Mean-Shift Mode Estimation [3]. However, the amount of computation for Mean-Shift procedure becomes prohibitively large as the dimensions of the space increases. Thus, we estimate the parameters in a hierarchical manner where parameters that affect the entire body are first estimated (global stage) and the patches which vote around this estimated value are selected. Subsequently, each limb's parameters are estimated independently using only the patches that have the appearance of each limb (local stage).

In order to further reduce the estimation time, we apply a modified version of the Locality-Sensitive Hashing algorithm [7] to patch retrieval. We modify the original hash function by considering the distribution of data so that each data point is uniformly hashed, leading to more efficient and stable retrieval. Using this algorithm, we drastically reduce computational time and achieve real-time estimation.

We use a monocular static silhouette image as our input as the silhouette of a human can be readily obtained through background subtraction. Using a silhouette greatly reduces the difficulty of pose estimation, which is important for practical use. We assume that the humans in the images are standing and facing the camera although we don’t restrict the position and scale of the human in the
image.

We have implemented our approach and conducted experiments using both synthetic silhouette images and real (non-synthetic) silhouette images obtained using standard background subtraction. Both the 2D and 3D positions of several key points of the body are used to measure the accuracy of our approach when using synthetic images. Experiments have shown that our proposed method successfully estimates unknown poses by assembling local patches from different poses.

This paper is structured as follows: In Section 2, we discuss related work. We describe the image patch retrieval in Section 3. In Section 4, we present our novel voting scheme. Section 5 contains our experiments and results. Finally, we present a discussion on future work and conclude our paper.

2. Related Work

Model-based approaches use a 3D human body model and optimize the model parameters based on model-image consistency. Much of the work using this approach tries to consolidate parts detection with global constraints [14, 15, 16]. Ren et al. [14] first detect candidate body parts from an image and find the most probable configuration using pairwise constraints such as the relative position and symmetry of the parts. Lee and Cohen [15] find the most consistent pose using data-drive MCMC which uses the result of parts detection as state proposals. Sigal and Black [16] also estimate human pose using a bottom-up body-part proposal with non-parametric belief propagation to find the best configuration. Although model-based approaches reduce the parameter space by exploiting local information, i.e., the result of the parts detector, the difficulty in using these approaches is that their performance relies on the success of parts detection which is generally difficult due to occlusions and ambiguities.

Discriminative approaches [8, 9, 10, 11, 12] use a number of labeled examples (image-pose pairs) to obtain a mapping from visual observations to pose parameters. Mori and Malik [8] use example images with labeled 2D joint positions and estimate those positions in the input image using a shape context matching. They then reconstruct the 3D pose using the estimated positions. Shakharovich et al. [10] propose Parameter-Sensitive Hashing, an extension of Locality-Sensitive Hashing, to rapidly retrieve similar examples from a database containing labeled examples. Athitsos et al. [12] propose an embedding method for efficient nearest neighbor retrieval and apply it to a synthetically generated hand pose database. Agarwal and Triggs [9] use relevance vector machine (RVM) to learn the mapping function to achieve rapid inference. Although these methods do not require computationally expensive stochastic optimization and human models during estimation, it is difficult to have a sufficient number of examples to cover all possible poses since they consider the entire body when matching the input image to the examples.

Meanwhile, in the field of object detection, Leibe et al. [2] propose a novel approach which can detect and localize unfamiliar objects in different articulations. Their method requires a smaller number of training examples since it can combine the information locally observed on different training examples to create a previously unseen appearance of the object. They use the probabilistic voting scheme to estimate the position of the object. Their method is flexible since the patches don’t necessarily need to correspond to the explicit body-parts unlike model-based approaches using the parts detector. Demirdjian and Urtasun [5] extend Leibe’s approach to pose estimation and show the potential of Leibe’s approach in the field of pose estimation. They use local patches and labeled pose parameters as training examples and estimate pose parameters using the probabilistic voting scheme. Our approach is also inspired by Leibe’s work but differs from Demirdjian’s work in some respects such as the use of the modified Locality-Sensitive Hashing and the hierarchical estimation.

3. Image Patch Retrieval

Our approach estimates a pose parameter by assembling local example patches from a number of different poses. As a preparing step, we construct a database containing those example patches using a 3D human model. In the estimation stage, patches are extracted from the contour of the input silhouette and visually similar example patches are retrieved from the database for each extracted patch using modified Locality-Sensitive Hashing.

3.1. Constructing a Patch Database

We use a 3D human model consisting of truncated circular cones and a spheroid, as shown in Figure 1(a), as the basis for the patch database. The model parameters are three joint angles for each shoulder and thigh, one joint angle for each elbow and knee and one fatness parameter that controls the radius of all truncated circular cones and spheroid. We denote a set of joint angles as \( \mathbf{X} = (x_1, \ldots, x_{16}) \) and the fatness parameter as \( f \). The fatness parameter is important in accommodating different body types. Our human model has 17 DOF in total.

We render \( N_{\text{pose}} \) different silhouette images using the human model according to model parameters generated from anatomically possible ranges. For each rendered image, \( N_{\text{db}} \) points on the contour of the silhouette are sampled sparsely to define the locations of the patches. For each patch, the shape context [1] with constant radius \( r_{\text{db}} \), vector \( \mathbf{v} \) from the patch's position to the reference
point of the model and the area of each limb within a patch \( a = (a_1, a_2, a_3, a_4) \) within the patch are calculated and stored in the database. We define \( a_1 \) to be the area of the left arm, \( a_2 \) the right arm, \( a_3 \) the left leg and \( a_4 \) the right leg as shown in Figure 1(b). The joint angle parameters \( \mathbf{X} \) and the fatness \( f \) of the pose which generates the patch are also stored. As a result, we get the database containing \( N_{\text{pose}} \times N_{\text{db}} \) patches with each patch represented by \( \{\mathbf{s}, \mathbf{v}, \mathbf{a}, \mathbf{X}, f\} \).

In order to make the algorithm scale invariant, we calculate the mean distance between all possible pairs of points for each pose and calculate \( R_{\text{db}} \), the mean value of all these mean distances. This is used to determine the radius of the shape contexts computed from the input image in the estimation stage.

### 3.2. Patch Retrieval

Given an input image, \( N_{\text{input}} \) points on the contour of the silhouette are sampled sparsely and the shape contexts are calculated on each of them. In order to give moderate scale invariance to the algorithm, \( R_{\text{input}} \), the mean distance between all possible pairs of points, is used to compute the shape context radius \( r_{\text{input}} \) as follows.

\[
r_{\text{input}} = \frac{R_{\text{input}}}{R_{\text{db}}} R_{\text{db}}
\]

For each such patch, the \( K \) most similar patches in the database are retrieved based on the similarity of the shape contexts. We employ \( L_1 \) distance, due to its simplicity.

Note that the \( L_1 \) distance is equivalent to histogram intersection in our case since the areas of the two histograms (shape contexts) are equal.

For each of the retrieved patches, the calculated shape context distance \( D \) is converted to the patch’s weight \( w \in [0,1] \) using \( w = \exp(-D^2) \). Additionally, the vector \( \mathbf{v} \) from the patch position to the reference point is converted to the position of the reference point in the given image using:

\[
\mathbf{m} = \mathbf{u} + \frac{R_{\text{input}}}{R_{\text{db}}} \mathbf{v},
\]

where \( \mathbf{u} \) is the position of each extracted patch.

As a result, we retrieve \( N_{\text{input}} \times K \) patches in conjunction with \( \{\mathbf{w}, \mathbf{m}, \mathbf{a}, \mathbf{X}, f\} \). These patches cast votes in the parameter space during recognition.

### 3.3. Locality-Sensitive Hashing for Patch Retrieval

Since the number of example patches in the database is large, retrieving a set of nearest neighbors is time consuming. In order to speed up retrieval, we use a modified version of the Locality-Sensitive Hashing (LSH) [7] algorithm instead of a brute-force search.

LSH is an approximate nearest neighbor method which hashes each data point using several hash functions so that similar data points get hashed to the same bucket while dissimilar ones do not. The proper hash function is chosen according to the distance metric used for retrieval. For example, if using Hamming distance, where each data is represented by \( d \)-dimensional binary vectors \( \mathbf{p} \in [0,1]^d \), the hash function would be \( h_i(\mathbf{p}) = p_i \), where \( i \in [1,\ldots,d] \) is a randomly chosen index. Thus, each hash function produces one bit.

For predefined parameters \( l \) and \( k \), \( l \) hash tables are prepared and for each table, \( k \) hash functions are randomly generated. For each table, each data point in the database is hashed using \( k \) hash functions and put into the bucket with the hash value. Since the total number of buckets can become intractably high as \( k \) increases, those indices are...
hashed again using standard hashing.

Given an input, while the total number of retrieved data points is smaller than a predefined threshold \( m \), hash value of the input is calculated for each table and the data points in the bucket with the same hash value are collected. This threshold is important for real-time systems to keep computation time per query within a certain time frame. A brute-force \( K \) nearest neighbor search is conducted only on these collected data points.

If using \( L_1 \) distance as a distance metric, each data point can be represented as vectors \( p \in \{0, \ldots, M\}^d \), where \( M \) is a maximum in any dimension over all vectors. We can convert these to binary vectors, where distance is calculated by Hamming distance, by replacing each coordinate \( p_i \) by a sequence of \( p_i \) ones followed by \( M - p_i \) zeros without changing the resultant distance.

However, converting data this way is time consuming and if \( M \) is big, the length of each vector will be too large. Thus, we adopt a more efficient hash function which produces the same hash values without converting the data. Instead of converting the data to binary vectors and randomly choosing an index, we randomly choose one dimension and put the value into a threshold function, i.e. \( T(x) = 1 \) for \( x \geq t \) and \( T(x) = 0 \) otherwise, where \( t \) is a threshold randomly sampled between \([0, M]\). As a result, similar to the original function, each hash function still produces one bit.

One problem with the original LSH is that it doesn’t consider the distribution of the data. Therefore, if the distribution of data is non-uniform, hash values are also non-uniform, which leads to inefficient retrieval. We calculate the distribution of the shape context values in the database. The shape context values are normalized between \([0, 1000]\). As can be seen in Figure 3, the distribution of the shape context values concentrate on lower values. Therefore, selecting threshold uniformly leads to significant inefficiency. We approximate this distribution as a Gaussian distribution and generate the threshold \( t \) according to this Gaussian. This way, the probability of data point being hashed as one and being hashed as zero is almost the same, which leads to an unbiased distribution of hash values.

![Figure 3: Distribution of shape context and approximate Gaussian distribution.](image)

4. Probabilistic Voting and Mode Search

The generalized Hough transform [6] is a well-known algorithm to detect template shapes in a given image. The template shape is represented by a set of points on the shape and listed in a table with vectors to a reference point on the shape and the tangent orientation of the point. Given the input image, points are extracted, and for each, points in the table which have similar tangent orientation values are activated. Each activated record votes for the possible position of reference point. The position which gets the most votes is the estimated reference position of the template shape.

Leibe et al. [2] propose a novel method to detect and localize uncertain objects using the generalized Hough transform. They use image patches instead of points on the shape to allow for a more reliable representation of the image. A codebook of image patches is learned and patches extracted from the input image are matched to the codebook entries. Each matched entry casts votes in a continuous parameter space with a Gaussian kernel to estimate the probability distribution. The local maxima are found by the Mean-Shift Mode Estimation [3]. The codebook representation is not adequate for pose estimation since only appearances that are prototypical for the object are kept. In order to discriminate between poses with subtle differences, appearances which are discriminative for the poses should be used for extension of this approach to pose estimation.

Demirdjian and Urtasun [5] extend Leibe’s idea to human pose estimation. They represent human poses as 3D positions of joints. In order to handle large training sets, they present a boosting-like feature selection approach which selects the most discriminative subsets of patches for pose estimation. However, as the number of dimensions of the model is still larger than that of just the location of the object, Mean-Shift search suffers from the curse of dimensionality. Estimating each joint’s position independently is likely to output false estimations due to the lack of constraints between each joint’s position.

We also extend Leibe’s idea, which allows us to combine patches from different parts of various poses to construct a novel pose as shown in Figure 4. This reduces the number of necessary training images since a novel pose can be created by patches from different poses. Unlike Demirdjian’s approach, however, where the 3D position of each joint is estimated, we estimate parameters of the 3D human body model which allows us to divide parameter space while retaining strong constraints.

4.1. Hierarchical Estimation

We divide the parameter space into several subspaces and estimate the highest mode (global maximum) in a hierarchical global-to-local manner.

To divide the parameter space, we use the following observations: The appearance of a human is globally
influenced by changes in parameters such as the location of the reference point and fatness. In contrast, influence of changes in a limb’s joint angles are locally limited and don’t necessarily affect the appearance of the other parts of the body. For example, a change in the angle of the right elbow only affects the appearance around the right elbow while a change in location and fatness of the body affect the entire appearance. Additionally, since each limb is only connected to the torso, we can consider each limb’s parameters independently from each other.

Therefore, we divide the parameter space into one global parameter space that consists of the location \( m \) and fatness \( f \), and four local parameter spaces, one for each limb’s joint angles. The  joint angles space \( X = (x_1, \ldots, x_{16}) \) is thus divided into four subspaces, \( x_1 = (x_1, \ldots, x_4) \) for right arm, \( x_2 = (x_5, \ldots, x_9) \) for left arm, \( x_3 = (x_{10}, \ldots, x_{12}) \) for right leg and \( x_4 = (x_{13}, \ldots, x_{16}) \) for left leg.

In the estimation stage, the global parameters are estimated first and the only patches which vote around the estimated value are selected and forwarded to the next step. Note that the forwarded patches are consistent in respect that they are patches from the body with the same location and fatness of the body affect the entire appearance. Additionally, since each limb is only connected to the torso, we can consider each limb’s parameters independently from each other.

Therefore, we divide the parameter space into one global parameter space that consists of the location \( m \) and fatness \( f \), and four local parameter spaces, one for each limb’s joint angles. The  joint angles space \( X = (x_1, \ldots, x_{16}) \) is thus divided into four subspaces, \( x_1 = (x_1, \ldots, x_4) \) for right arm, \( x_2 = (x_5, \ldots, x_9) \) for left arm, \( x_3 = (x_{10}, \ldots, x_{12}) \) for right leg and \( x_4 = (x_{13}, \ldots, x_{16}) \) for left leg.

In the estimation stage, the global parameters are estimated first and the only patches which vote around the estimated value are selected and forwarded to the next step. Note that the forwarded patches are consistent in respect that they are patches from the body with the same location and fatness of the body affect the entire appearance. Additionally, since each limb is only connected to the torso, we can consider each limb’s parameters independently from each other.

Therefore, we divide the parameter space into one global parameter space that consists of the location \( m \) and fatness \( f \), and four local parameter spaces, one for each limb’s joint angles. The  joint angles space \( X = (x_1, \ldots, x_{16}) \) is thus divided into four subspaces, \( x_1 = (x_1, \ldots, x_4) \) for right arm, \( x_2 = (x_5, \ldots, x_9) \) for left arm, \( x_3 = (x_{10}, \ldots, x_{12}) \) for right leg and \( x_4 = (x_{13}, \ldots, x_{16}) \) for left leg.

In the estimation stage, the global parameters are estimated first and the only patches which vote around the estimated value are selected and forwarded to the next step. Note that the forwarded patches are consistent in respect that they are patches from the body with the same location and fatness of the body affect the entire appearance. Additionally, since each limb is only connected to the torso, we can consider each limb’s parameters independently from each other.

Therefore, we divide the parameter space into one global parameter space that consists of the location \( m \) and fatness \( f \), and four local parameter spaces, one for each limb’s joint angles. The  joint angles space \( X = (x_1, \ldots, x_{16}) \) is thus divided into four subspaces, \( x_1 = (x_1, \ldots, x_4) \) for right arm, \( x_2 = (x_5, \ldots, x_9) \) for left arm, \( x_3 = (x_{10}, \ldots, x_{12}) \) for right leg and \( x_4 = (x_{13}, \ldots, x_{16}) \) for left leg.

In the estimation stage, the global parameters are estimated first and the only patches which vote around the estimated value are selected and forwarded to the next step. Note that the forwarded patches are consistent in respect that they are patches from the body with the same location and fatness of the body affect the entire appearance. Additionally, since each limb is only connected to the torso, we can consider each limb’s parameters independently from each other.
\[ z(a_i) = \begin{cases} \frac{2}{1 + e^{-c a_i}} - 1, & a_i > 0 \\ 0, & a_i \leq 0 \end{cases} \] (6)

where \( c \) is a predefined parameter. Voting and mode search are conducted in each of the four subspaces independently. Probabilistic interpretation of this procedure is similar to the global parameter estimation except that each voting is weighted by the confidence.

5. Experiments

5.1. Dataset

In order to create a test dataset for quantitative evaluation, we used a 3D human model rendering software, Poser\(^2\), to render synthetic silhouette images of both a male and a female figure. We set the human model to face front and stand upright, but the position and scale of the model vary. We created 1000 test images with joint angles and body type parameters randomly generated from anatomically possible ranges determined by analyzing CMU motion capture database [4].

We used both 2D and 3D positions of the center of the head, the right shoulder, the left shoulder, the right elbow, the left elbow, the belly, the right knee, the left knee, the right ankle and the left ankle to measure the accuracy of the estimation. In the test images, both the 2D and 3D positions of these key points were represented by pixel coordinates. Estimated positions of these key points were easily computed from the estimated pose of our cylindrical human model. We computed the average RMS error of the estimated key points’ 2D and 3D positions over all test images. For reference, the average height of the individuals in the test images was approximately 210 pixels. All experiments were conducted on an Intel Xeon 3.3GHz PC with unoptimized C++ code.

maximum in the entire parameter space at one time. The obtained average RMS error was 33.9 pixels for 3D positions and 12.4 for 2D, which are, as expected, much worse than the result of hierarchical method. This result indicates that the proper space division allows Mean-shift search to successfully find the modes in each divided space.

Finally, we measured the effect of the number of extracted points from the input \( N_{\text{input}} \). We changed \( N_{\text{input}} \) from 10 to 200 using intervals of 10. As shown in Figure 9, the accuracy showed reasonable improvement as \( N_{\text{input}} \) increases until it reaches 50. However, there is no notable improvement for higher although the computation time increased.

Several sample estimation results using synthetic images with \( N_{\text{input}} = 100 \) are shown in Figure 10 and those using real human images extracted via standard background subtraction in Figure 11 for qualitative evaluation. Our method performed well on not only the synthetic images but also the real human images with a few background subtraction errors.

### 6. Conclusion and future work

In this paper, we propose a novel patch-based 3D human body pose estimation algorithm. We focus on the fact that different parts of various poses can create previously unseen poses. To this end, we construct a database containing many patches extracted from different parts of various poses. Patches are extracted from the input image and visually similar patches are retrieved from the database using modified Locality-Sensitive Hashing allowing for real-time estimation. Estimation is successfully done in a hierarchical voting scheme where globally influential parameters are first estimated and then local pose parameters are estimated.

In the future, we plan to extend our approach to cover a wider range of poses such as leaning and crouching. The body’s orientation and occlusion would also have to be handled for practical use.

**Acknowledgement**

The author would like to thank Jacky Chang from MIT for her work on the developing PoserPython scripts.
References


Figure 10: Our method is able to accurately identify poses from synthetically generated images with known foreground.

Figure 11: Our method is able to accurately infer poses from real human silhouette extracted via standard background subtraction.