Pose Estimation From Occluded Images

Kanako Hayashi
kanako.hayashi@stanford.edu

Lionel Heng
lionel.heng@cs.stanford.edu

Vikram Srivastava
vicky528@stanford.edu

Abstract

We propose a learning-based framework for inferring the 3D pose of a person from monocular image sequences. We generate a silhouette from each input image via a robust background subtraction algorithm, and compute the corresponding shape context descriptor using the shape context algorithm. We compute the weighted average of neighbor poses in a database to estimate the positions of different body parts in the input image. We discuss several ways to make the framework more robust.

1. Introduction

The estimation and tracking of 3D human body pose is a challenging problem in computer vision. This problem has important applications in a wide variety of areas such as visual surveillance and human-computer interaction. With tools to recover 3D human body pose from images, computers can model and recognize human behaviors and analyze human body dynamics.

To make pose estimation more robust to ambiguities resulting from occlusion of body parts, cluttered backgrounds, varied clothing and other nuisance parameters, we can use a range camera [1] or multiple cameras [2]. However, it is desirable to be able to estimate 3D poses using monocular vision as the use of a multiple-camera setup or a range camera is infeasible for many situations such as surveillance and analysis of archived videos.

There are two types of approaches to the pose estimation problem: example-based and model-based. Generally, example-based approaches utilize supervised learning in which we store a set of training examples with known 3D poses, search for training examples similar to the given input image, and interpolate from the set of poses corresponding to the similar training images. In contrast, model-based approaches assume an explicitly known parametric body model, and estimate the pose either by directly inverting the kinematics or by numerically optimizing some form of model-image correspondence metric over the pose variables, using a forward rendering model to predict the images.

We adopt an example-based approach in order to avoid the use of complicated models and overcome the occlusion problem. We use silhouettes with interior edge information as they are invariant to many nuisance parameters such as illumination, clothing, color and texture. These silhouettes are represented by shape contexts [6], which are robust feature descriptors.

To estimate the pose in an input image given a database of images with known 3D poses, we find shape context descriptors whose distances from the context descriptor representing the pose in the input image are less than a specified threshold. To estimate the parameters for the input pose, we compute the weighted average of the body part positions corresponding to these shape context descriptors.

2. Previous Work

There has been much recent work on the monocular pose estimation problem. Sigal et al. [9] adopts a model-based approach, eliminating the use of a database. This approach uses a learned Mixtures of Experts (MoE) model to infer a distribution of 3D poses conditioned on 2D poses. However, this approach is constrained to finding 3D pose in a monocular image sequence, and does not work for a single image. In addition, this approach makes assumptions about a reasonable image likelihood model and the availability of detectors for specific body parts. In [6], shape contexts are used to represent the contour shape of the human body, resulting in generally accurate pose estimation, but we can do better by utilizing useful information relating to appearance within a silhouette, such as the presence of body parts, especially the arms, within the silhouette. Also, this approach has a high time complexity and can take several minutes for the analysis of a single image. Hence it is infeasible for a real-time system. In [7], parameter-sensitive hashing permits possibly real-time pose estimation. The estimation of 3D pose from single input images gives rise to ambiguous poses in some cases, but we can use a tracking framework such as in [8] to mitigate this problem. Our approach to the monocular pose estimation problem is similar to [8], but we introduce interior edge information into silhouettes, and use parameter-sensitive hashing to quickly infer the body joint angles from a single image.

None of the approaches discussed so far take into account the problem of occlusion by foreign objects. The features used to describe the pose can change drastically under occlusion. The bottom-up approaches, as described in [9], which depend on finding the different body parts may fail to operate under such cases. Hence, we model the effects of occlusion explicitly in order to make the system more robust. We specifically investigate occlusions in five different ways which can cover most of the scenarios we can come across practically.

3. Pose Estimation

We propose a shape-context-based approach for the problem of pose estimation. Belongie et al. have used this concept for matching shapes in [6]. For our purpose, we apply this algorithm for finding the best matching image from our training database. The entire procedure is divided into two phases: the training and testing phases which are described in sections 3.1 and 3.2 respectively.

3.1. Training Phase

In this phase, we use Vikram as the model for creating the training database. 114 images were collected with him standing in different poses. Artifacts such as shadow attachment and background noise can significantly impact the performance of our pose estimation system, as these artifacts distort the silhouettes. Hence, we require a robust background subtraction algorithm. We use Tola's implementation [3] of the kernel density estimation based background subtraction algorithm explained in [4], [5] to obtain silhouettes. This algorithm is robust to background clutter, changes in illumination, and shadows. Then we run the shape context algorithm on these silhouettes.

3.1.1 Shape Contexts

The shape context algorithm treats a shape as a set of n points. These points can be taken by doing sampling on the silhouettes or edge images. From each of these n points, we consider the vectors to the remaining $n - 1$ points. These vectors are quantized into 60 bins on the basis of their lengths and the angle they make with the horizontal. Thus, for each of the n points, we obtain a histogram with 60 bins and each bin containing the number of vectors that are closest to it in terms of (r, θ) co-ordinates. This process is described for a single point in figure 1 with $n = 350$. In figure 1(e), we divide the lengths of the vectors into 5 levels by dividing each of them by the length of the longest vector. The x-axis contains the 12 polar bins corresponding to each of the 5 levels in ascending order.

The shape context of a single point gives information about how the rest of the image looks with respect to it. Hence, it is a rich descriptor of the overall shape of the image. Now, the image is divided into 9 zones of equal size as illustrated in figure 2(a). The mean of histograms of all

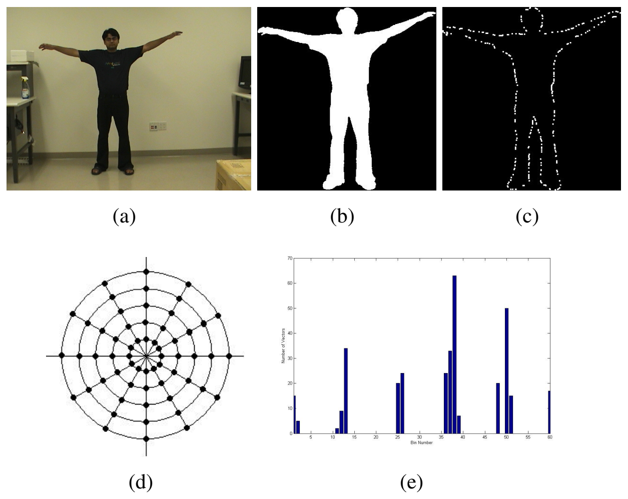


Figure 1: (a) Original image of Vikram. (b) The corresponding silhouette. (c) 350 points on the silhouette. (d) The 60 bins used to quantize the vectors. (e) Shape context histogram for the uppermost point in (c).

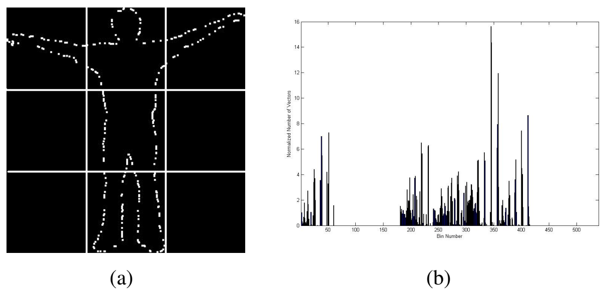
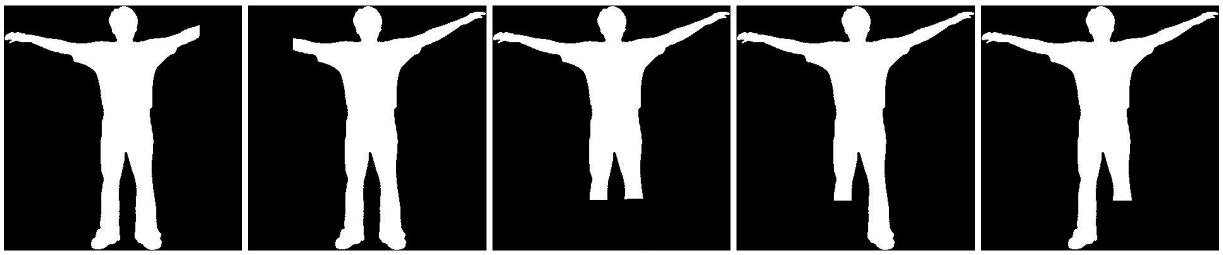


Figure 2: (a) The division of the points into 9 zones. (b) The final histogram for the image in figure 1(a).


(a)

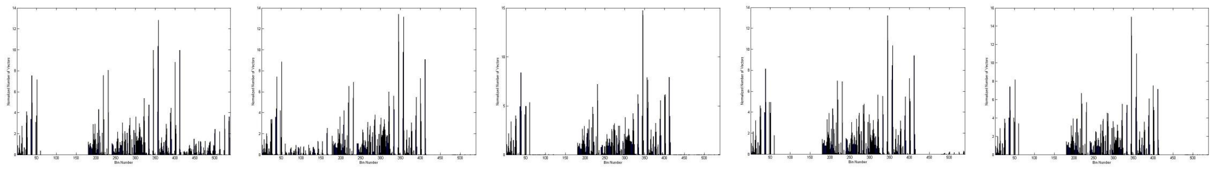

(b)

Figure 3: (a) The five images give an example of the five categories of occlusion. (b) Shape context histograms for each image.

points falling in the same zone is taken, and then all the 9 histograms are appended to create a final histogram of 540 bins representing the entire silhouette. This histogram is taken as the feature vector for comparing the silhouettes. The histogram corresponding to figure 1(c) is shown in figure 2(b).

3.1.2 Histograms for the Occluded Images

In addition to the histogram for the complete image, we also compute 5 other histograms for occluded versions of each image. The occluded images are created by artificially blocking the images in the following 5 ways: (i) Right, (ii) Left, (iii) Below, (iv) Left Bottom, and (v) Right Bottom. For each of the five categories, we computed histograms by occluding the original image to different extents and then finding the mean of all the histograms. Figure 3(a) shows one of the occluded images belonging to each of the five categories and the corresponding shape context histograms are shown in figure 3(b).

3.1.3 Labeling of Training Data

We hand-label the positions of the following body parts for each silhouette in the training database: the head, hands, shoulders, center of the body, and knees. We normalize the coordinates of these positions so that we can use them with test silhouettes of varying sizes. This normalization allows us to use the training database for people standing at different distances from the camera, and with different physiques.

3.2. Testing Phase

We divide the testing phase into two parts: testing on stored image sequences, and testing using a live demo. For both parts, we use the following distance metric to compute the similarity between two shape context descriptors, each representing a silhouette:

$$D(i,T) = \sum_{k=1}^{540} \frac{(h_i(k) - h_T(k))^2}{h_i(k) + h_T(k)}$$

where i corresponds to the ith image from the training database, T corresponds to the test image, and $h(k)$ gives the normalized histogram value for the kth bin.

3.2.1 Testing on Stored Images

We collected images of Lionel and Kanako, which then served as the test images for the training images of Vikram. A shape context descriptor was computed for each of these images, and compared with the descriptors corresponding to all the training images to obtain the best matching silhouette. We first tested for cases where there was no occlusion. The algorithm performed reasonably well in the cases where the silhouette can express the overall shape of the pose. The physiques of Lionel and Kanako are significantly different from that of Vikram, and the algorithm still manages to find the best match for both of them. Figure 4 shows the best match for the images of Lionel and Kanako.

We further tested the algorithm by artificially occluding the images of Lionel and Kanako and finding the best matching image from the database. The occlusion for the test cases was random and did not belong specifically to one of the five categories described in section 3.1.2. Figure 5 shows the best matching image from the database for each of two test cases.

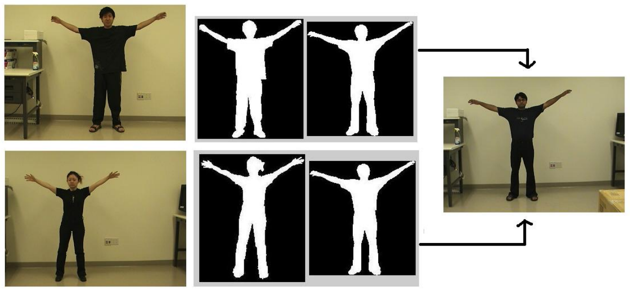


Figure 4: The best match for images of Lionel (top) and Kanako (bottom) from the database of images of Vikram.

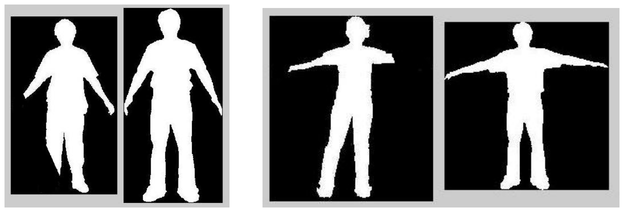


Figure 5: Best matching silhouettes under random occlusion.

3.2.2 Testing Using a Live Demo

We decided to convert all the code written in Matlab to C++ in order to increase the processing speed for the live demo. We used a popular open source computer vision library, OpenCV, to perform all the image processing. For the live demo, we estimated the locations of the different body parts mentioned in section 3.1.3. We first obtained the shape context feature vectors for the silhouettes, and then obtained all feature vectors from our training database whose distance from the test feature vector was below a certain threshold. Since the number of such vectors is usually small, and each vector is extremely high-dimensional (540 dimensions), we cannot use linear regression to estimate the positions of the different body parts as the matrix $X'X$ will often be singular where X is the matrix containing the training examples' body part positions in its rows. Hence, we obtained the positions of the body parts using a weighted mean of the positions of the body parts in the set of close training examples. The formula for the position is:

$$P_{est} = \frac{\sum_{i=1}^{N}(\lambda - D(i,T))P_i}{\sum_{i=1}^{N}(\lambda - D(i,T))}$$

where P_{est} is the estimated position in the test image T, N is the number of close training examples, λ is the upper threshold of the distance between two shape context descriptors, $D(i,T)$ is the distance between the shape context descriptor of the ith close example and that of test image T, and P_i is the position of the body part in image i.

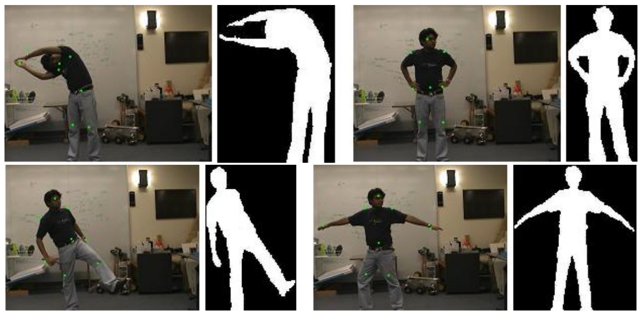


Figure 6: Positions of body parts for different poses along with the closest silhouette from the training database.

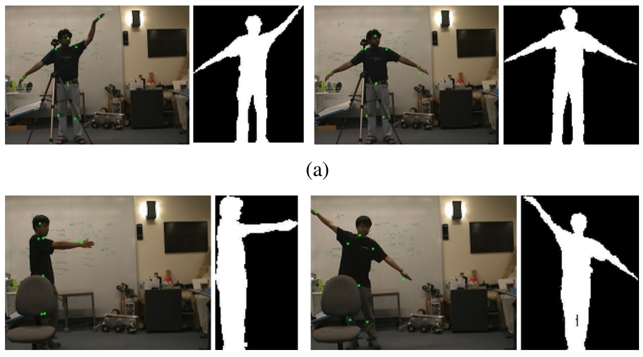


Figure 7: Positions of body parts for different poses along with the closest silhouette from the training database for (a) occlusion by a tripod and (b) occlusion by a chair.

Figure 6 shows the positions of the different body parts marked by green dots for an unoccluded test subject.

We further tested the system by placing a tripod and a chair in front of the test subject in order to create natural occlusion. Figure 7 shows the estimates of the positions of body parts for a few cases.

The hands move most rapidly compared to all the other body parts. Hence, we plot the results for our estimate of the hand positions for all the three cases described above. The graphs are shown in Figure 8.

4 Discussions and Future Work

Our pose estimation system runs in real-time and works well for poses which have close neighbors in the training database. As we see in figures 6 and 7, for each test image, the closest silhouette in the training database closely matches the image, and the green dots correctly mark the positions of the body parts. Occlusion of a significant part of the body does not compromise the accuracy of pose es-

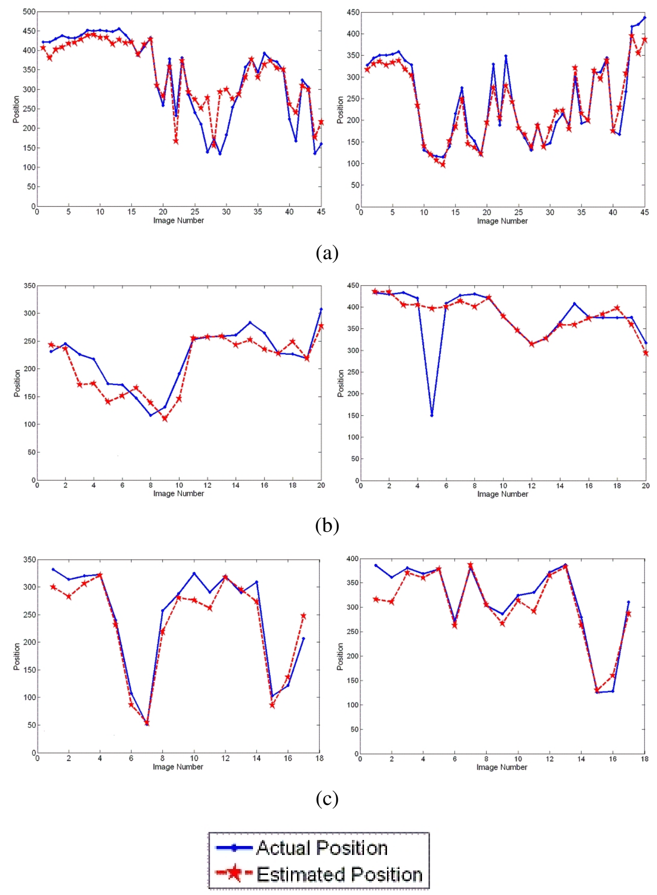


Figure 8: The graphs on the left column show the plot of the estimated position of the left hand against the actual postion, and the graphs on the right column show the plot of the estimated position of the right hand against the actual position, for a) an unoccluded test subject, b) a test subject occluded by a tripod, and c) a test subject occluded by a chair.

timation. In figure 8, the estimated position of each hand does not deviate significantly from the actual position.

We note that the performance of the system is constrained by the size of the training database, which in our case contains roughly 1300 images. This is clearly insufficient for robust pose estimation, as these images cover a small subset of all possible human poses. To obtain reliable results, the database should contain at least 100000 images with parameter values sampled independently and uniformly within anatomically feasible ranges [7]. In future work, we can use POSER [10] to render synthetic training images from a humanoid model.

In such a large database, it is inefficient to search through all the images to find the best match. We can use parameter-sensitive hashing [7] to preserve the real-time performance of the system. Parameter-sensitive hashing uses parameter-sensitive hash functions, in other words, hash functions that are sensitive to the similarity in the parameter space, and retrieves in sublinear time approximate nearest neighbors of the test image with respect to parameter values as well as the shape context feature vectors. The sublinear running time achieved by examining only a fraction of the dataset, yet not compromising the accuracy of the pose estimation makes real-time performance possible. We can then use robust locally-weighted linear regression [11] to find the positions of the body parts in the image. Even if we come across a test image which has no exact match in the training database, we can still obtain an estimate of their coordinates by computing a weighted average of the values for the k training images most similar to the test image. Furthermore, robust LWR minimizes the influence of outliers.

Due to the loss of depth and limb labeling information for single silhouettes, the resulting 3D pose can be ambiguous. To make the system robust to such ambiguity, we can implement a regressive tracking framework described in [8] as long as our pose estimation system can accommodate a high frame rate. This framework recovers the most likely pose at each time step by using a dynamical model of the human body learned from training data to predict the 3D pose distribution and a learned regression value. Results in [8] show that the regressive tracking framework tracks long sequences stably.

5 Acknowledgements

We would like to thank Rahul Biswas for his help and guidance throughout the course of the project and for providing the necessary equipment.

References

[1] David A. Simon, Martial Hebert and Takeo Kanade, Real-time 3-D Pose Estimation Using a High-Speed Range Sensor, *Proceedings of the IEEE International Conference on Robotics and Automation*, 1994.

[2] Tomas Izo, W. Eric L. Grimson, Simultaneous Pose Estimation and Camera Calibration from Multiple Views, *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops*, 2004.

[3] Engin Tola, <http://cvlab.epfl.ch/∼tola>

[4] B. Birant Orten, Medeni Soysal and A. Adym Alatan, Person identification in surveillance video by combining MPEG-7 experts, *6th International Workshop on Image Analysis for Multimedia Interactive Services*, Montreux, Switzerland, 2005.

[5] Ahmed Elgammal, David Harwood, and Larry Davis, Non-parametric Model for Background Subtraction, *6th European Conference on Computer Vision*, Dublin, Ireland, 2000.

[6] Serge Belongie, Jitendra Malik and Jan Puzicha, Shape Matching and Object Recognition Using Shape Contexts, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002.

[7] Gregory Shakhnarovich, Paul Viola and Trevor Darrell, Fast Pose Estimation with Parameter-Sensitive Hashing, *Proceedings of the International Conference on Computer Vision*, Nice, France, 2003.

[8] Ankur Agarwal and Bill Triggs, Recovering 3D Human Pose from Monocular Images, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):44-58, 2006.

[9] Leonid Sigal and Michael J. Black, Predicting 3D people from 2D pictures, *AMDO 2006 - IV Conference on Articulated Motion and Deformable Objects*, Mallorca, Spain, 2006.

[10] e frontier, Scotts Valley, CA, *Poser 6 - Reference Manual*, 2005.

[11] W. S. Cleveland, Robust locally weighted regression and smoothing scatter plots, *Journal of American Statistical Association*, 74(368):829-836, 1979.