Human Pose Estimation for Multiple Frames

Marianna Neubauer
Stanford University
mhneub@stanford.edu

Hanna Winter
Stanford University
hannawii@stanford.edu

Lili Yang
Stanford University
yangl369@stanford.edu

Abstract

Human pose estimation is a well studied topic in vision. However, most modern techniques in human pose estimation on multiple, consecutive frames, or motion capture, require 3D depth data, which is not always readily available. Prior work using single view 2D data, on the other hand, has been limited to pose estimation in single frames. This raises some interesting questions. Can human pose estimation in multiple frames be effected using 2D single frame techniques, thereby discarding the expensive reliance on 3D data? Can these 2D pose estimation models be improved upon by taking advantage of the data similarities across multiple consecutive images? In this paper, we endeavor to answer these questions. We take Yang et al.*'s [1] single frame pose estimation model using flexible mixture of parts and apply it in a multi-frame context. We demonstrate that we can achieve improvements on the original method by taking advantage of the inherent data similarities between consecutive frames. We achieve speed improvements by restricting Yang* et al.*'s to search locally in intermediate frames and, under certain circumstances, accuracy improvements by running a second, corrective, pass using SVMs trained for instance recognition.*

1. Introduction

Human pose estimation has become an extremely important problem in computer vision. Quality solutions to this problem have potential to impact many different aspects of vision such as activity recognition and motion capture. Additionally, success in these aspects can be applied to gaming, human-computer interaction, athletics, communication, and health-care. Despite huge progress in motion capture, as exemplified with the Xbox Kinect, the current solutions used in gaming require extensive hardware making it impossible for such technology to be used in daily human-computer interactions [2]. We hope to improve motion capture to work with simple RGB single view cameras allowing this technology to exist on everyday phones and computers.

State of the art models for human pose estimation that are implemented for single static RGB images, also have some minimal but noticeable accuracy shortcomings [1]. Currently, when used on video frame sequences, these models do not utilize the additional information provided by surrounding frames. Operating under the assumption that human poses change minimally between frames, we improve the accuracy of Yang *et al.*'s [1] efficient and flexible model for human detection and human pose estimation in single static images. We take into account the sift features of other frames in the same video clip by training SVMs on these features. We can improve the output of Yang's model by testing the SVMs on parts of the images and adjusting the original body parts to reflect the scores calculated by the trained SVMs. The result is a notable increase in accuracy of the imperfect Yang pose estimation.

After discussing related work and the implications of our method in Section 2, we further describe our process, resulting algorithm, and our evaluation process in detail in Section 3. Finally, we analyze our testing data and experimental results for our various methods and hyperparameters in Section 4.

2. Background

2.1. Review of Previous Work

Human pose estimation is a well studied subject, both in video (multiple frames) and in images (single frames). Currently, most modern techniques for pose estimation in video rely on 3D depth data. A well known example of this is the xBox Kinect [2], which uses pose estimation to determine the gamer's motion. 3D depth data has many advantages over 2D image data, not the least of which is the additional dimension of information. However, 3D data can only be captured using specialized, and often expensive equipment and is not as nearly ubiquitous as 2D videos.

Recent work in pose estimation on 2D image data

feature a wide range of techniques and approaches, among them Yang's [1], Agarwal's [3], Dantone's [4], and Toshev's [5]. These methodologies are similar in that they focus on pose estimation on single images. We focus primarily on Yang's [1] method of pose estimation using a flexible mixture-of-parts. Yang's method has the advantage of producing relatively good results on full body images across a variety of poses and background contexts, while still retaining a significant speed advantage over certain other approaches, such as Toshev's [5] pose estimation using convolutional neural networks. A relatively fast algorithm is of particular significance when we consider pose estimation in the multi-frame context.

2.2. Our Method

Previous methods for pose estimation in the multi-frame realm rely on 3D depth data. Our method uses only RGB single view image data to accurately locate 26 different body parts. Additionally, our SVMs are trained specifically on information from a given video clip resulting in a more accurate classification of small, specific body parts. Because deep learning would not be feasible in this context, as neural networks take too long to train and require an extremely large amount of training data, we believe our method is the best learning-based technique to improve pose estimation in the multi-frame context.

3. Technical Details

3.1. Overview of Methodology

Utilizing the available source code, we are improving Yang *et al.*'s Image Parse model algorithm [1] on a variety of image sequences of human motion gathered from Youtube. As an initial attempt, we implemented a HOG features search algorithm where we compute HOG features for each frame and find the location of body parts by searching for similar features to those calculated for the body part in the frame prior. We found that although this method dramatically speeds up the process, the results are worsened. Then, we implemented an SVM correction method where we train an SVM for each body part for each video clip. We improve the original Yang by testing the SVMs on parts of the image and adjusting the Yang output based on the scoring results. Expanding upon this method, we integrated hard negative mining [6] for computing logical negatives for each SVM. Additionally, we added a double-pass with another SVM trained to classify a sub-image as a body part or background. Finally, in order to measure the accuracy of our computed bounding boxes we manually annotate ground truth bounding boxes on the same image sequences.

3.2. Yang Algorithm Speedup

The original implementation of Yang's mixture of parts algorithm runs in 30 seconds on a typical clip from our test set (see section 4.1). Since we are testing on upwards of 2000 images, this is unacceptably slow. Also, in a multi-frame video with multiple people the highest scoring bounding boxes often migrate from person to person. To remedy these issues we reduced the space in which the mixture of parts algorithm searched for the bounding boxes.

For the first frame of the video clip we run the full Yang algorithm. For the second frame, we crop the image to the box bounding containing the entire person plus a little extra, the size of a body part bounding box, on the top, bottom, and sides. We then run the full Yang algorithm on the cropped image. We store the pyramid level that is used for the bounding boxes on the second image. For the third frame and all subsequent frames, we crop the image using the same method to crop the second image and we search only within the pyramid levels above, at, and below the previously stored pyramid level.

Cropping the image ensures the bounding boxes do not migrate to another person and speeds up the search for the bounding boxes. Reducing the pyramid levels also results in significant speedup. Instead of 30 seconds, the algorithms runs in about 0.1-0.4 seconds per frame. This speedup made our SVM correction method, described in section 3.4, feasible because it allowed us to run Yang on all the frames of a given video clip in a reasonable amount of time. This was necessary to obtain enough training data for the SVMs.

3.3. Interpolation with HOGs

The HOG interpolation method relies on the assumption that a person's pose can change only so much between consecutive frames. Therefore, given the bounding boxes for body parts in one frame, we are assured that the associated bounding boxes in subsequent frame may be found in the same vicinity and would retain similar features.

Our implementation uses Yang's model to select bounding boxes for the first frame of the target sequence. In each subsequent frame, for each bounding box, we run a sliding window search in the local vicinity of its location in the prior frame to select candidate bounding boxes. We then select the candidate with the closest match in HOG features to the associated bounding box in the prior frame.

By running Yang's relatively expensive procedure only on the first frame, we are able to achieve significant speed improvements over a full run of Yang's across all frames. However, this methodology has two disadvantages. Firstly,

any pose estimation error made by Yang in first frame are propagated into the subsequent frames. Secondly, the quality of the interpolation disintegrates the farther removed we are from the initial frame. The key weakness of interpolation with HOGs is that it takes into account the output of Yang's model for only a single frame. In subsequent investigations, we focus instead on producing accuracy improvements using SVMs trained on the output across all frames.

3.4. SVM Correction

Considering only a single human in each of the image sequences, we notice that various features, such as the color of their skin or clothing, do not change over frames. Using this observation, we train video clip specific SVMs to improve the output from Yang's model [1]. From Yang, there are 26 bounding boxes indicating locations of 26 body parts for each frame. We split up the frames into sub-images defined by each bounding box as seen in Figure 1 and treat each of these sub-images as training data for the SVMs. Additionally, for each frame, we compute negative examples by randomly selecting bounding boxes within a certain area of the human and then discarding those that overlap with any of the calculated body part bounding boxes. We then repeat this process until enough negative examples are found (Figure 2).

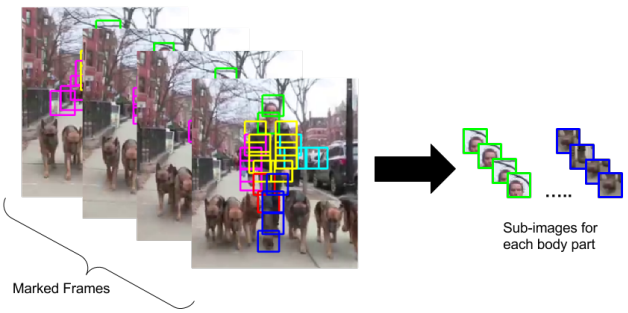


Figure 1: A visualization of segmenting the frames from the bounding boxes calculated by Yang [1] to create the training data used to train the 26 different SVMs.

Utilizing the VLFeat library [7], we compute cluster centers from the combined training data by calculating the sift features for each training example and using k-means clustering to find centers for all the sift features. Note that sift features were computed using the RGB information as the colors are important features for training. We found that a larger number of centers, such as 100, produced better results. For all of the training data, we create Bag of Words feature vectors. For each pyramid depth, p, we break the training example into a $p \times p$ grid of sub-images and take the sift feature vector for each section. After finding the

Figure 2: The process of finding negative examples in each frame. The leftmost image shows the boundary around the person in which random bounding boxes are found. The center image shows these boxes. Then, all the boxes that overlap with any of the body parts are filtered out and the resulting bounding boxes that will become negative examples are displayed in the rightmost image. This process is repeated until a sufficient amount of negative examples are found.

closest cluster center to each sift feature vector, we create a histogram of this distribution and concatenate all sub-image histograms together to form our Bag of Words. The Bag of Words features are then used to train the 26 SVMs. For a given SVM for body part a, all features for the 25 other body parts and for the negative sub-images are treated as negative examples.

In order to improve the original output from Yang's model [1], we test the SVM on every 10 frames using a sliding window. As shown in Figure 3, for a given frame and a given body part, a, we initialize a score for the SVMs associated with a on the original calculation from Yang. Then, we start sliding a window of the same size as the original computing a score at every position with the SVM for a. The window position with the maximum score becomes the corrected bounding box.

Figure 3: The sliding window method. The image to the left displays the original Yang output for the left hand. The middle image shows the sliding window starting from the to left and moving across and down. A score is calculated for each position. The image on the right indicates the corrected body part which is the position of the sliding window that resulted in the best score.

3.4.1 Double-Pass SVM

After our initial results, we noticed that if Yang's model mistakenly placed enough bounding boxes on parts of the background that our SVM would do the same. We improve our method by using an additional, background distinguishing SVM. We train this SVM on the same feature set as the 26 body parts SVMs but using as positives all body parts bounding boxes and as negatives all background bounding boxes. During the sliding windows stage, this SVM is used to filter candidate bounding boxes. Only bounding boxes that are classified as non-background are kept and subsequently scored by the corresponding body part SVM.

3.4.2 Hard Negative Mining

To further improve our method, we take advantage of the hard negative mining method [6]. In this addition to our SVM Correction technique, we train our original 26 SVMs without any negative examples aside from other body parts. Then, using these SVMs, we test the on the randomly selected negatives collected by our previous method. We do this over a series of iterations where in each iteration we collect new negative examples, test these negative examples on all 26 SVMs, take the maximum score, and then keep a maximum of 30 negative examples for each video frame that have a positive score. Our iterations stop once we have kept a sufficient amount of negative examples. Using this technique, we are able to collect the most "confusing" negatives to train our SVMs on. We then recompute the cluster centers and Bag of Words features including the negative examples and re-train all 26 SVMs. The correction step using the sliding window technique remains the same.

3.5. Evaluation

To evaluate the performance of our algorithm, we measure how many body parts are correctly localized by comparing the pixel positions of the computed bounding boxes and the manually annotated ground truth bounding boxes. The Image Parse model outputs the four corners of a square bounding box while the manual annotation only stores the centroid of a bounding box. We measure the intersection over union of the computed bounding box and the ground truth. We assume the size of the bounding box for the ground truth is the same as the size of the computed bounding boxes. A bounding box is labeled "correct" if its IOU is above a certain threshold.

To aggregate this data for a single video clip, we count the number of frames a body part is correctly localized and divide that by the total number of frames. This number is the average precision (AP) of the algorithm for that body part in that video clip.

To evaluate the performance of our algorithms, we compute an AP vs. overlap threshold curve (AOC), similar to the AP curve described in [8]. A robust algorithm should generate a curve that maintains high AP for all overlap thresholds, however some drop off is expected. If there is a drop off it should occur at high overlap thresholds.

Different regions of the body have drastically different performances. In general arms and legs perform more poorly than head and torso in Yang's algorithm. Therefore, we also look at the average raw IOU for each region of the body for each clip to see if the relative performance between different algorithms depends on the body region. We defined seven regions: head, left torso, left arm, left leg, right torso, right arm, and right leg.

4. Experiments

4.1. Dataset

Yang's model [1] is pre-trained on the Image Parse dataset [9]. For testing, we require a dataset containing human full-body footage because the model is trained on images containing full-body poses.

To capture a variety of poses, we pulled video footage from Youtube containing varied subject matter [10], [11], [12], [13] such as people walking, dancing, and playing sports. We cut these videos such that each clip contains a single camera view and the full-body of the subject. We preprocess the clips to obtain image sequences of the frames. Each frame is downsized using bicubic interpolation to be about 256x256 pixels while maintaining the original aspect ratio. The downsizing is done to match the approximate size of the testing images used in [1].

The ground truths associated with our dataset were made by manually clicking the points of all 26 different body parts for every 10 frames. Each click is the centroid of a bounding box for a given body part. For evaluation, we believe comparing every 10 frames with the ground truth values is sufficient to determine accuracy.

4.2. Results

4.2.1 HOG interpolation

The HOG interpolation failed to provide accurate bounding boxes for subsequent frames because of drift. Any pose estimation error made by Yang in first frame are propagated into the subsequent frames and the quality of the interpolation disintegrates the farther removed we are from the initial frame. Figure 4 shows the decrease in average IOU with increasing frame number. In general, the average IOU

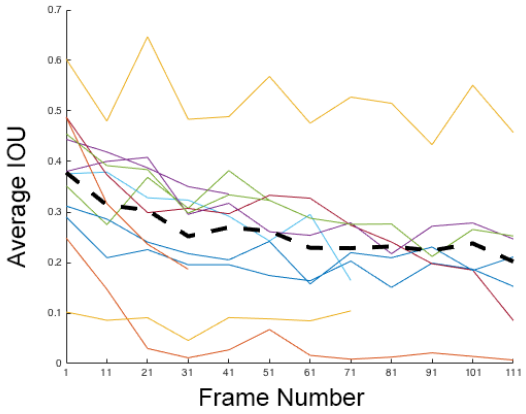


Figure 4: Average IOU over for all video clips. Each thin solid line represents a clip. There are 12 clips ranging from 51 to 121 frames. The black dotted line is an average IOU over all clips.

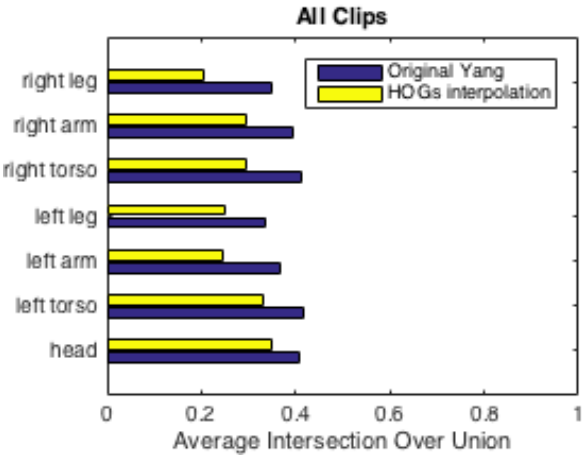


Figure 5: Average IOU over all clips for each body region of Yang output (blue) and HOGs interpolation (yellow).

overlap with the ground truth over all frames in all clips is significantly lower than in the original Yang output (Figure 5).

All clips performed worse under HOG interpolation except for Walking Clip 3 (the diamond in Figure 6). A histogram of the average IOU for each body region reinforces that finding (Figure 7). This is likely not because the HOGs performed well but instead because the Yang output performed poorly for that particular clip. Note that the left arm in Figure 8 is not properly localized by the Yang output, but the HOGs have some overlap with the ground truth. Also note that the right arm has better localization in the HOGs interpolation than the Yang output.

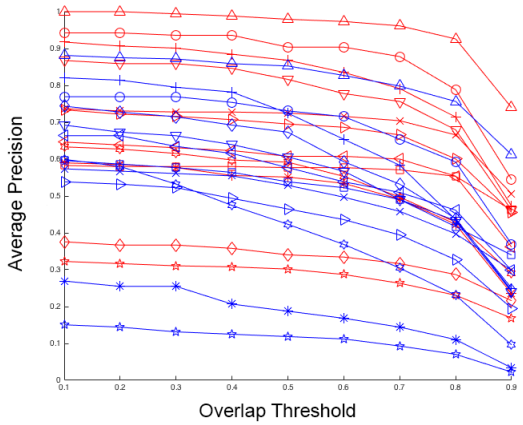


Figure 6: AP vs. Overlap Threshold Curve of the original Yang output (red) and the HOGs interpolation output (blue). Lines with corresponding symbols indicate corresponding clips. For example, the triangle symbol is the Yang and HOG evaluation for Walking Clip 1.

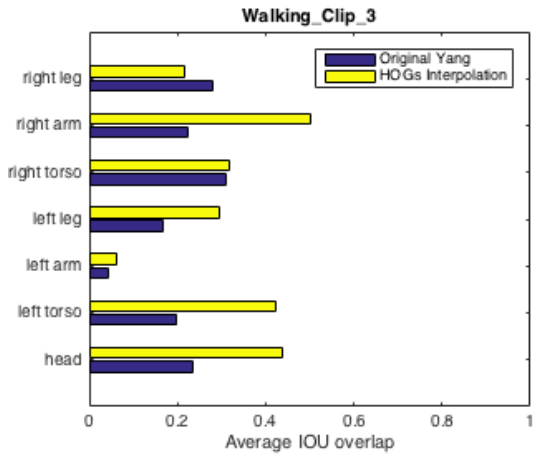
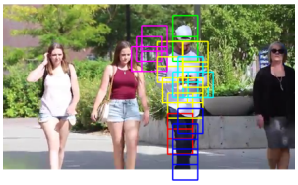


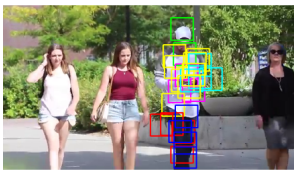
Figure 7: Average IOU in Walking Clip 3 for each body region of Yang output (blue) and HOGs interpolation (yellow).

4.2.2 One Pass SVM with Randomly Selected Negatives

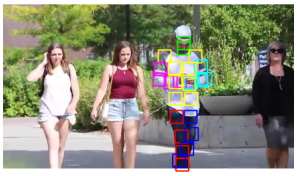
Our single pass SVM has a pyramid depth of 5 and 100 cluster centers because those parameters produced consistently good results. We trained and tested the SVM on 5 clips and found that it improved the performance of two of the clips, and decreased performance in two of the clips, and did not change the performance in one of the clips (see Figure 9). Specifically, the SVM improved Beyonce Clip 1 and MLB Clip 1, it made worse Dog Walking Clip 2 and Walking Clip 1, while Dog Walking Clip 1 remained the same. The improvement in Beyonce Clip 1 was very



(a) Yang Output

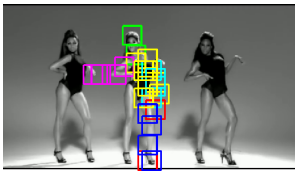


(b) HOGs

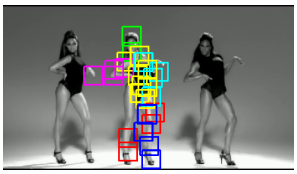


(c) Ground Truth

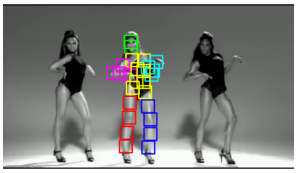
Figure 8: Frame 41 of Walking Clip 3.



(a) Yang Output



(b) Single Pass SVM



(c) Ground Truth

Figure 10: Frame 91 of Beyonce Clip 1

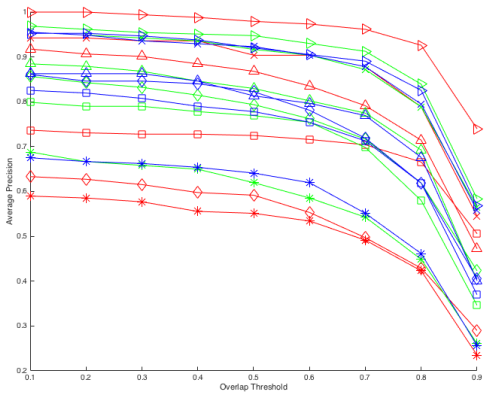


Figure 9: AP vs. Overlap Threshold Curve of the original Yang output (red), the single pass SVM correction (blue), and the double pass SVM correction (green). Lines with corresponding symbols indicate corresponding clips. diamond: Beyonce CLip 1, asterisk: Beyonce CLip 6, x: Dog Walking Clip 1, triangle: Dog Walking Clip 2, square: MLB Clip 1, carrot: Walking Clip 1.

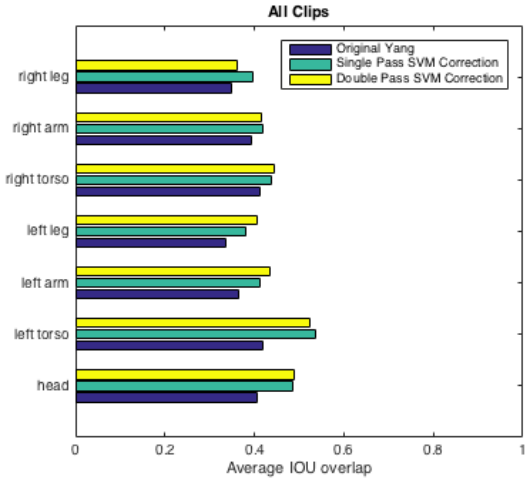


Figure 11: Average IOU over all clips for each body region of Yang output (blue), single pass SVM correction (green) and double pass SVM correction (yellow).

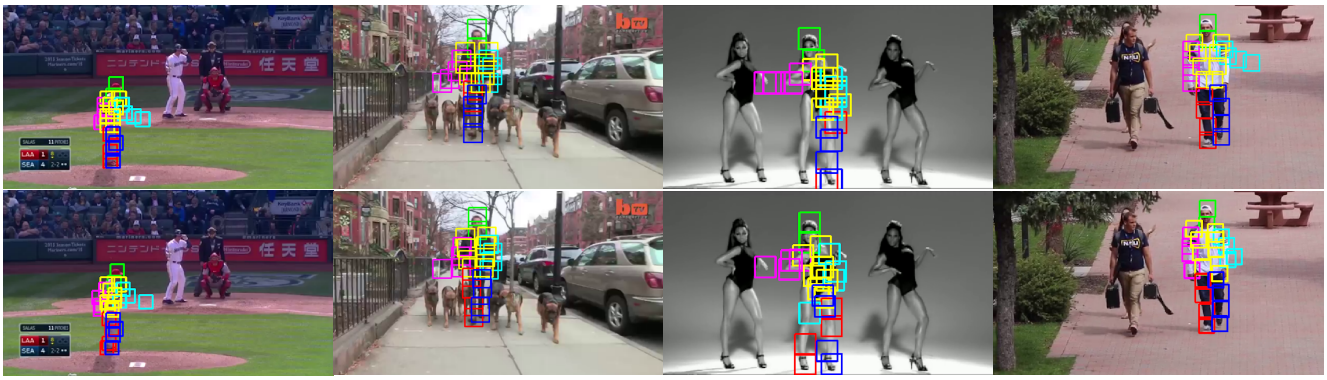
significant (the asterisk in Figure 9). Figure 10 shows that the original Yang output placed the bounding boxes too far right and the SVM correction shifted them back to the center. The SVM also fixed one of the bounding boxes in the left (pink) arm.

Averaging the IOU over all of the clips (Figure 11) reveals that the SVM did slightly worse for all body regions except for the head, left torso and left arm.

4.2.3 Double Pass SVM with Hard Negatives

The double pass with hard negative mining improves the performance over the single pass SVM in Beyonce Clip 6, Dog Walking Clip 2, and Walking Clip 1 (Figure 9). However, in Dog Walking Clip 2 and Walking Clip 1, it still performed worse than the original Yang output. The double pass SVM performed significantly better than the original Yang output in Beyonce Clip 1 and Beyonce Clip 6. For MLB Clip 6, the SVM corrections have higher average precision at lower and middle thresholds while the original Yang output has a higher average precision at the highest thresholds. In Dog Walking Clip 1 the performance of all three methods are similar.

In general, both the single and double pass SVM, when averaged over all the clips, resulted in more accurate bounding boxes than the original Yang output (Figure **??**). The Single pass SVM performs the best for the left torso, and right leg, while the double pass SVM performs the



(a) MLB Clip 1 (b) Dog Walking Clip 1 (c) Beyonce Clip 1 (d) Walking Clip 1

Figure 12: Example frames from various different clips displaying the SVM correction using hard negative mining and a double pass. The top row is the original Yang result and the bottom row is the result after our SVM correction.

best with the head, left arm, left leg, and right torso. This indicates that the extra background SVM pass and the hard negatives mining did improve the performance of the SVM correction overall especially since the arms from incorrect Yang outputs tend to include background sub-images.

Figure 12 shows example frames of where the double pass SVM corrects errors in the original Yang output. For example, the right arm for Walking Clip 1 in the Yang output bounds the background while in the SVM correction it bounds the right arm. For MLB Clip 1 and Beyonce Clip 1 the arms move closer to the body in the SVM correction except for one bounding box. The solitary bounding box remains far away because the true arm is outside of the search space defined by our correction algorithm. In Beyonce Clip 1 the left and right legs alternate probably because the sift features are very similar between left and right legs. There is also a right arm bounding box on the left leg because, please, Beyonce's legs basically look like arms anyway.

5. Future Work

If given the time, we could make several modifications to our SVM. Firstly, we did not tune all of the parameters of the SVM across all of the clips to find the best overall set of parameters. We also noticed that while some values worked well for some clips, they worked less well for others. More investigation in this area could produce interesting insights.

Secondly, in the implementation of the hard negatives for the SVM, we arbitrarily set a threshold to decide whether to include the negative example in our set of negatives for the final SVM. For some clips this threshold was too high and it was difficult to collect enough negative examples in a reasonable time. In the future we could vary the threshold and create another AP curve or ROC curve based on that threshold to determine its effect on the performance of the SVM.

The current implementation of the SVM is impractically slow. The most time is spent computing the Bag of Words feature vectors in various parts of our algorithm including the hard negative mining loop and the sliding window correction section. Therefore, we believe this to be the bottleneck of our method. Thus, parallelizing this computation such that all frames or even all body parts in each frame are computed in tandem could have a significant speed up.

6. Conclusion

It is certainly true that human pose estimation is a challenging subject with many avenues of research yet to be explored. We have made a small effort by introducing a method that utilizes the similarities among video frames to improve a single image pose estimation model when used in a multi-frame context. The improvement was particularly marked on the clips where the original Yang's algorithm performed the most poorly - and arguably where improvement was most necessary.

More importantly, we have highlighted areas where more research is possible and laid the groundwork for future avenues of investigation.

References

[1] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *IEEE Conf. on Computer Vision and Pattern Recognition*

(CVPR), pages 1385–1392, Washington, DC, USA, 2011. IEEE.

[2] B. Bonnechre, Jansen B., P. Salvia, H. Bouzahouene, Omelina L., J. Cornelis, M. Rooze, and S. Van Sint Jan. What are the current limits of the kinect sensor? In *9th International Conf. on Disability, Virtual Reality and Associated Technologies*, pages 287–294, Laval, France, 2012.

[3] A. Agarwal and B Triggs. 3d human pose from silhouettes by relevance vector regression. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages II–882–II–888 Vol.2, June 2004.

[4] M. Dantone, J. Gall, C. Leistner, and L. van Gool. Human pose estimation using body parts dependent joint regressors. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3041–3048, Portland, OR, USA, June 2013. IEEE.

[5] A Toshev and C Szegedy. Deeppose: Human pose estimation via deep neural networks. *CoRR*, abs/1312.4659, 2013.

[6] Andrea Vedaldi. Object category detection practical. http://www.robots.ox.ac.uk/ vgg/practicals/category-detection.

[7] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008.

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[9] D. Ramanan. Learning to parse images of articulated bodies. In *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, pages 1129–1136, 2006.

[10] beyonceVEVO. Beyonce - single ladies (put a ring on it). https://www.youtube.com/watch?v=4m1EFMoRFvY.

[11] Barcroft TV. Dog whisperer: Trainer walks pack of dogs without a leash. https://www.youtube.com/watch?v=Cbtkoo3zAyI.

[12] Cesar Bess. Mlb top plays april 2015. https://www.youtube.com/watch?v=mpe9w-CHsoE.

[13] BigDawsVlogs. Walking next to people extras. https://www.youtube.com/watch?v=776niN4-A58.