# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 Pose estimates by HPE algorithms. The six upper body parts, estimated by the algorithms, are color coded as pink for head, red for torso, green for upper arms and yellow for the lower arms.</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Pose retrieval example: The query pose (leftmost image) is represented by the upper-body stickman figure. It is followed by the top five retrieved images where the people present have the most similar pose as the query.</td>
<td>3</td>
</tr>
<tr>
<td>2.1 In the left image, the pictorial structure model for the face illustrated by a spring model in Fischler and Elschlager [43]. When viewed as a graph, the left edge, right edge and other parts on the face form the nodes. The edges between them are illustrated as a spring. The deformation function $d_{ij}$, like a spring, should be flexible in allowing the parts to take a wide range of relative positions while penalizing them as they move away from the expected relative positions. In the right image, the various parts of the face, which form the nodes in the graph, are depicted by the rectangles. The connections between them, which form the edges, are depicted by the red line segments.</td>
<td>8</td>
</tr>
<tr>
<td>2.2 The example graph displayed above has six nodes with $v_1$ as the root, ${v_5, v_6}$ as the leaves and the rest as intermediate nodes. The belief propagation algorithm proceeds from the left to right as follows. First (left image) the messages $\psi_{5-&gt;3}$ and $\psi_{6-&gt;4}$ are computed at the leaf nodes ${v_5, v_6}$ using the equation 2.7. Next (center image) the messages at the intermediate nodes ${v_3, v_4}$ are computed using the equation 2.8. Note that the messages passed by the leaf nodes are used in computation at the intermediate nodes. Finally (right image) the message at the root node is computed using the equation 2.9. Note that each message computation has a complexity of $O(d^2)$ where $d$ is the number of states of each random variable.</td>
<td>10</td>
</tr>
<tr>
<td>2.3 Various layers in CNN: The convolutional layer is taking an input of dimensions $30 \times 30 \times 3$, applying 50 filters each of dimension $5 \times 5 \times 3$ and to output an array of dimension $26 \times 26 \times 50$. Here, the number of parameters relative to the inputs and outputs are significantly small at 3750. On the convolutional layer’s output, max pooling with filter size $2 \times 2$ is applied to obtain $13 \times 13 \times 50$ array. Finally in a fully connected layer, all the $M$ input neurons and $N$ output neurons are connected to each other requiring $MN$ parameters.</td>
<td>12</td>
</tr>
</tbody>
</table>
2.4 The images above illustrate the two ways in which parts and their relations have been defined. In (a), the whole anatomical body parts form parts in the pictorial structures. The body parts modelled are head, the two arms and the torso. The state space for each part would be the location and orientation. In (b), the body joints and few other points are modelled as parts in the pictorial structures. The state space for each part would be the location and orientation. In both figures, the rectangles represent the parts and the two neighboring rectangles connected by line segments have an edge in the pictorial structures.

2.5 Pose estimations by (a) Andriluka et al. [5], (b) Eichner and Ferrari [26], (c) Sapp et al. [80] and (d) Yang and Ramanan [103].

3.1 Annotated ground-truth samples (stickman overlay) from the Movie Stickmen dataset (all except last row, last column) and Buffy-2 dataset (last row and last column).

3.2 Pose similarity measures: The PCP and angle difference measures are computed on pair of example poses here. Given a pair of poses (1), they are transformed into a normalized pose space (2). The origin for this normalized pose space is the mean point of the head and torso line segments. The scaling factor is the length of the torso. In some implementations, the mean lengths of all the body parts are also taken. Finally the PCP and angle difference measures (3) are calculated. The exact formulas are given in the above image. In this image only the calculation for the lower left arm is depicted. (Best viewed in color)

3.3 Ground-truth samples (stickman overlay) from all the movies present in Movie Stickmen dataset (first row) and Buffy-2 dataset (second row).

4.1 Pose evaluator illustration. The HPE algorithm is run on images in the top row to output the pose estimates and a confidence score (row two). It incorrectly assigns low confidence score to a correct pose estimate (Ed Harris’s image). Pose evaluator accurately distinguishes (row three) the correct pose estimates from the incorrect ones.

4.2 Features based on the output of HPE. Examples of unimodal, multi-modal and large spread pose estimates. Each image (first row) is overlaid with the best configuration (sticks) and the posterior marginal distribution over the body part position in a semi-transparent mask (displayed in second row). ‘Max’ and ‘Var’ features are measured from this distribution (third row). In the case of unimodal distributions, as the above examples (a and b) indicate, the mode almost always corresponds to the correct part location. While in the case of multimodal distributions, typically one of the modes correspond to the correct part location as in example (c), but again they may not as in example (d). Finally in the case of diffuse distributions (e and f), the many modes that barely stand out do not convey any information about correctness of part locations. Thus unimodal distributions are good indicators of correct part configurations as opposed to the other two types. The type of distribution is determined using the ‘Max’ and ‘Var’ features. As the distribution moves from peaked unimodal to more multi-modal and diffuse, the maximum value decreases and the variance increases.

4.3 Human pose estimates on incorrect upper-body detections: typical examples of human pose estimates in a false positive detection window. The pose estimates (MAP) are unrealistic (shown as a stickman, i.e. a line segment per body part) and the posterior marginals (PM) are very diffuse (shown as a semi-transparent overlay: torso in red, upper arms in blue, lower arms and head in green). The pose estimation outputs are from Eichner and Ferrari [26].
4.4 **Pose normalizing the marginal distribution.** Marginal distribution (contour diagram in green) on the left is pose normalized by rotating it by an angle $\theta$ of the body part (line segment in grey) to obtain the transformed distribution on the right. In the inset, the pertinent body part (line segment in grey) is displayed as a constituent of the upper body.  

4.5 **Overlap features.** First row, Left: Three overlapping detection windows. First row, Right: the HPE output is incorrect due to interference from the neighbouring people. This problem can be flagged by the detection window overlap (output from Eichner and Ferrari [26]). Second row: Few more examples of incorrect HPE outputs due to interference from the neighbouring people. 

4.6 **Poses with increasing CPC.** An example per CPC is shown for each of the two reference poses for CPC measures of 0.1, 0.2, 0.3 and 0.5. As can be seen example poses move smoothly away from the reference pose with increasing CPC, with the number of parts which differ and the extent increasing. For 0.1 there is almost no difference between the examples and the reference pose. At 0.2, the examples and reference can differ slightly in the angle of one or two limbs, but from 0.3 on there can be substantial differences with poses differing entirely by 0.5. 

4.7 **Pose notation:** For poses A and B corresponding parts $p$ are illustrated. Each part is described by starting point $s_{a}^{p}$, end point $e_{a}^{p}$ and the angle of the part $\theta_{a}^{p}$ are also illustrated. 

4.8 (a) **Performance of HPE evaluator in regime A:** (no false positives used in training or testing). The ROC curve shows that the evaluator can successfully predict whether an estimated pose has CPC < 0.3. (b) **Performance of HPE evaluator in Regime B:** (false positives included in training and testing). 

4.9 **Example evaluations.** The pose estimates in the first two rows are correctly classified as successes by our pose evaluator. The last two rows are correctly classified as failures. The pose evaluator is learnt using the regime B and with a CPC threshold of 0.3. Poses in rows 1,3 are estimated by Eichner and Ferrari [26], and poses in rows 2,4 are estimated by Yang and Ramanan [103]. 

4.10 **Improving average precision using the pose evaluator.** The figure shows the precision and recall of HPE on the test set described in section 4.3.1. We see that all the methods have a significant positive affect on average precision (AP). Pose evaluator has successfully filtered incorrect pose estimates as shown in 4.10a and fused the pose estimates from Eichner and Ferrari [26], Yang and Ramanan et al [103] as shown in 4.10c. 

4.11 **Pose Representation:** The two poses A and B differ in two parts, the upper and lower left arms. The pose representation based on the angles clearly distinguishes pose A from pose B. The endpoints of each stickmen are consistently numbered. 

5.1 **Pose retrieval overview.** An illustration of the three query modalities: interactive stickman GUI, pose from Kinect, and pose estimated from an image. The output ranked list is obtained instantaneously as the query is varied. This can be tailored in various ways, for example to select only poses in different shots or within a particular movie. The results of the query pose are shown here at the video level. 

5.2 **Screen-shot of the pose retrieval system.** The elements of the web page such as the three query modes, options for selecting the database and the level of retrieval and video player, amongst other things, are indicated by text annotations and pointers in light red color. The interactive stickman can be moved around and the results are instantly updated as a ranked list. Clicking on a thumbnail plays a video of that shot.
LIST OF FIGURES

5.3 **Pose retrieval evaluation:** The poses displayed in the second row are used to evaluate the performance of the system. The corresponding numbers below are the precision values of the top 12 results. The third and fourth row respectively are the precision values using the ‘Filter II (Fl2)’ and ‘Hybrid pose estimates (Hpm)’ methods. The mean precision over the ten pose queries is 31.7% and 32.5% respectively.

5.4 **Pose retrieval examples:** For the query poses displayed in the first row, five retrieved results are displayed.

6.1 **Discovered deep poselets:** Six deep poselets and instances belonging to them are shown. For each deep poselet, an average image marked with stickman and example instances are displayed. A deep poselet is composed of subset of body parts in a particular pose as indicated by the stick figure on the average image. The body parts and their poses in each example instance matches its corresponding deep poselet.

6.2 **Deep poselet method:** The proposed deep poselet method has four parts: (a) Discovery: First, poselets of various body joint configurations (illustrated in the figure) are discovered by clustering in the pose space (b) Training: These poselets are then trained using convolutional neural networks. (c) Detection: Each poselet has been observed to have a localized area within the upper body bounding box. We term this area as “Expected poselet area (EPA)”. The poselet detection is performed within this area. (d) Post processing: The EPA of several poselets intersect (e.g., all poselets belonging to the left arm). Thus within the same area, several poselets have detections while only small number of them are correct. Using linear regression we re-score the poselets detections using the context of other poselet detections. (Best viewed in color)

6.3 **Spatial reasoning:** For a given test sample, three deep poselet detections and their scores are shown as belonging to the area marked by an orange rectangle. Detections 1 and 3 are partially correct as the pose of the left upper arm matches that of the test sample. Detection 2 is the correct one. Typically many such deep poselet detections, often mutually exclusive, have significant overlap. Using spatial reasoning, these detections are rescored such that correct ones (detection 2) get a score of nearly 1 and the partially or totally incorrect ones (detection 1 and 3) get a score of nearly 0. The image also shows that area around the left arm (orange rectangle) has 15 unique deep poselets while area around the right arm (pink rectangle) has 13 unique deep poselets.

6.4 **Deep poselets vs HOG poselets:** The graphs show the performance of three deep poselets on test data. The red curve in each graph corresponds to HOG poselet while the green curve corresponds to the deep poselet. As can be seen, the deep poselet outperforms the HOG poselet.

6.5 **Pose search performance:** The distribution of query performances by various retrieval methods are shown. Each bar in the graph shows the percentage of queries (Y-axis) having an average precision (X-axis). Thus the more the number of queries on the right side of the graph the better the method. This is also reflected by the mean of the distribution (mAP) of various methods given in the top right corner. It is clear that the proposed method significantly outperforms other methods.
6.6 **Top deep poselet detections**: Three deep poselets and top detections by them are shown. For each deep poselet, every fifth detection is displayed. In the top 50 detections, while there are no mistakes in deep poselet (a), there are 4 mistakes in deep poselet (b) and 20 mistakes in deep poselet (c). In the deep poselets (b) and (c), the first mistakes occur at ranks 20 and 10 respectively. It can be seen that the performance of deep poselets improve as the number of training samples increases.

6.7 **Example retrievals**: Top retrievals and AP curves for three queries are displayed. For the top retrievals every fifth sample from the top in retrieved list is displayed. The first mistake occurs at ranks 11, 4 and 33 respectively for the above queries.

7.1 **CNN architecture**: This architecture is a minor variation of the CNN architecture proposed in [58]. The number of layers and the number of the parameters are depicted.

7.2 **Triplet architecture**: A training sample to this network contains a tuple of three images. The three images are reference image, a positive image which contains the similar pose as reference image, and a negative image which contains a dissimilar pose to the reference image. The images are forward propagated and passed to the loss function which measures how well the network separates reference-positive pair and the reference-negative pair. This architecture can be visualized as containing three networks, each for an image, with same CNN architecture and same set of parameters at any given time.

7.3 **Pose search performance**: The distribution of query performances by various retrieval methods are shown. Each bar in the graph shows the percentage of queries (Y-axis) having an average precision (X-axis). Thus the more the number of queries on the right side of the graph the better the method. This is also reflected by the mean of the distribution (mAP) of various methods given in the top right corner. It is clear that the proposed method significantly outperforms other methods.

7.4 **Deep pose embedding network’s performance**

7.5 **Deep poselet analysis**: For the two query-retrieval pairs, the top and bottom poselets based on prominence scores are displayed for all three poselet categories. For the first pair (above), the retrieval is incorrect. This analysis clearly shows that the top poselet in positive-negative category has misfired for the retrieval image. Similarly the top poselet in negative-negative category also misfired. For the second pair (below), the retrieval is correct and all the prominence scores reflect it.
List of Tables

Table | Page
---|---
3.1 **Our dataset statistics.** The number of images and stickmen annotations for the two newly introduced datasets *viz.*, Movie stickman dataset and Buffy2 dataset are given above. While the total number of stickman annotations are 12414, only 4163 of them have all the parts visible. | 24
3.2 **Off-the-shelf dataset statistics.** Samples from publicly available datasets are added to the newly introduced datasets to increase the dataset size. The number of annotations do not include the flipped versions. | 24
4.1 **Train-test split.** For our experiments, two publicly available datasets and two newly introduced datasets are used. The train-test stickmen annotations along with other statistics are given above. | 39
4.2 **Pose estimation evaluation (PCP).** The PCP $^1$ of the four HPE algorithms (Andriluka et al. [5], Eichner and Ferrari [26], Sapp et al. [80], Yang and Ramanan [103]) averaged over each dataset (at PCP threshold 0.5, section 4.2.2). The numbers in brackets are the results reported in the original publications. The differences in the PCPs are due to different versions of UBD software used. The performances across the datasets indicate their relative difficulty. | 40
4.3 **Pose estimation evaluation (CPC).** The table shows the percentage of samples which were estimated accurately (CPC $< 0.3$) on the training and test sets, as well as overall, for the four HPE algorithms (Andriluka et al. [5], Eichner and Ferrari [26], Sapp et al. [80], Yang and Ramanan [103]). These accurate pose estimates form the positive samples for training and testing the evaluator. | 40
4.4 **Performance of the Pose Evaluator in Regime B.** The pose evaluator is used to assess the outputs of four HPE algorithms (Andriluka et al. [5], Eichner and Ferrari [26], Sapp et al. [80], Yang and Ramanan [103]) at three different CPC thresholds. The evaluation criteria is the area under the ROC curve (AUC). BL is the AUC of the baseline and PA is the AUC of our pose evaluator. | 41
5.1 **Dataset statistics.** We consider 18 movies for the retrieval system. For each video, the number of images, shots, tracks and human pose estimates are reported. Movies with * are abbreviations for ‘Buffy the Vampire Slayer’ , ‘Four weddings and a Funeral’, ‘Living in oblivion’, ‘Lost in Translation’, ‘My Cousin Vinny’, ‘Desperately Seeking Susan’ and ‘A fish called Wanda’. | 52

$^1$We are using the PCP measure as defined by the source code of the evaluation protocol in [26], as opposed to the more strict interpretation given in [71].
<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.2</td>
<td><strong>Approximate nearest neighbor search vs Exhaustive search.</strong> The recall of the top 100 ground truth matches is averaged over 1000 queries.</td>
<td>56</td>
</tr>
<tr>
<td>6.1</td>
<td>The contributions of various datasets before adding the flipped versions.</td>
<td>67</td>
</tr>
<tr>
<td>6.2</td>
<td>Performance of five randomly chosen deep poselets on various CNN features over the test data.</td>
<td>68</td>
</tr>
<tr>
<td>6.3</td>
<td>Comparison of our method with state-of-art poselet methods on the test data.</td>
<td>68</td>
</tr>
<tr>
<td>6.4</td>
<td>Pose search performance (mAP) and pose representation’s dimensions of various methods.</td>
<td>71</td>
</tr>
<tr>
<td>7.1</td>
<td>Pose search performance (mAP) and pose representation’s dimensions of various methods.</td>
<td>80</td>
</tr>
</tbody>
</table>