Merging Pose Estimates Across Space and Time

Xavier P. Burgos-Artizzu[1]
xpburgos@caltech.edu
David Hall[1]
dhall@caltech.edu
Pietro Perona[1]
perona@caltech.edu
Piotr Dollár[2]
pdollar@microsoft.com

[1] California Institute of Technology
Pasadena, CA, USA
[2] Microsoft Research
Redmond, WA, USA

Abstract

Numerous 'non-maximum suppression' (NMS) post-processing schemes have been proposed for merging multiple independent object detections. We propose a generalization of NMS beyond bounding boxes to merge multiple pose estimates in a single frame. The final estimates are centroids rather than medoids as in standard NMS, thus being more accurate than any of the individual candidates. Using the same mathematical framework, we extend our approach to the multi-frame setting, merging multiple independent pose estimates across space and time and outputting both the number and pose of the objects present in a scene. Our approach sidesteps many of the inherent challenges associated with full tracking (e.g. objects entering/leaving a scene, extended periods of occlusion, etc.). We show its versatility by applying it to two distinct state-of-the-art pose estimation algorithms in three domains: human bodies, faces and mice. Our approach improves both detection accuracy (by helping disambiguate correspondences) as well as pose estimation quality and is computationally efficient.

1 Introduction

Accurate pose estimation from video is key to many applications such as action recognition [6, 23], motion capture [20] and human computer interaction [21]. By *pose* here we mean the parameters of a model that describes the configuration of an object in the image or, alternatively, the location of a number of object parts or landmarks in the image.

Data driven approaches for pose estimation are maturing and starting to show impressive results on a broad range of recognition tasks [7, 13, 25, 26]. These methods naturally output a set of pose hypotheses and rely on 'non-maximum suppression' (NMS) techniques to merge detections that are associated with the same objects. NMS is well developed for the case of object detection where the goal is to merge rigid object locations (bounding boxes) [9, 11]. However, it is still unclear how to extend it to flexible pose estimates. Applying standard NMS independently to each part location as in [25, 26] fails to explicitly leverage the higher dimensionality of pose parameterization.

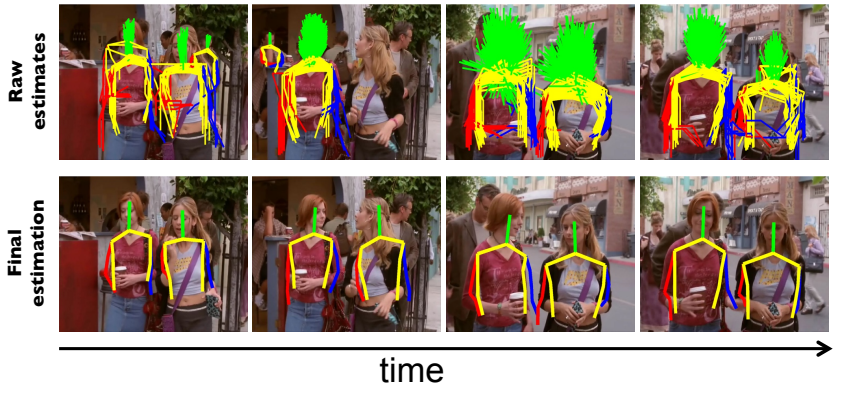


Figure 1: Our approach, Pose-NMS, merges together pose estimates in space and time, outputting a final set of estimates more accurate than any of the individual ones. This is achieved by performing a robust average while simultaneously solving the correspondence problem.

Our first contribution is a principled framework for merging multiple pose estimates in a single frame. This can be viewed as a generalization of NMS beyond bounding boxes. Our proposed approach makes minimal assumptions about the underlying method for pose estimation and generates a final set of pose estimates that are more accurate than any of the individual ones. We achieve this by performing a robust average while simultaneously solving the correspondence problem between pose estimates generated by multiple objects.

Our second contribution is to extend our approach to the multi-frame setting using the same mathematical framework, resulting in pose estimates that are further improved. While our approach is inspired by the recent success of 'tracking by detection' approaches, we sidestep many of the inherent challenges associated with full tracking (e.g. objects entering and leaving a scene, extended periods of occlusion, etc.). Instead we present a principled, simple approach for merging multiple independent pose estimates across space and time and outputting both the number and pose of the objects present in a scene, see Fig. 1.

By formulating the task in a unified optimization framework we derive an efficient and highly effective approach well suited for computing pose estimates 'on the fly' given short video snippets during which the same objects are observed. Our approach, *Pose-NMS*, makes minimal assumptions about the underlying pose estimation method, making it possible to use any frame-by-frame pose estimation approach and parameterization type.

In Sec. 3 we showcase the versatility of Pose-NMS by applying it to two distinct state-of-the art pose estimation approaches: Deformable Part Models (DPM) for body pose estimation [25] and Cascaded Pose Regression (CPR) [7, 13]. We collected more than 1,000 short video clips from scenes containing three different object types and benchmark our method on three tasks: human body pose estimation, face landmark estimation, and animal pose estimation. Pose-NMS improves both detection accuracy and pose estimation quality in all three cases compared with standard techniques. Code for our approach is available online.

1.1 Related Work

When estimating pose from video instead of single images, previous work can be divided into two main categories: 1) approaches that couple together the tracking and pose estimation stages directly and 2) approaches that first estimate pose independently frame-by-frame and subsequently enforce temporal smoothness across frames.

The first category dominates for markerless human motion-capture and 3D pose estimation from several cameras [20]. Examples include [22], where the authors propose to perform tracking and pose estimation simultaneously through a factored-state Hierarchical HMM, or [18], which proposes to integrate single-frame pose recovery and temporal integration by combining a motion model and observations in a Viterbi-style maximum likelihood approach. These approaches are specifically tailored for each task and are not easily generalized to different pose estimation tasks.

The second category of approaches, more popular in standard 2D pose estimation from monocular video, consist in extending single image detection approaches to video by enforcing temporal consistency across frames. In this context, popular "tracking by detection" techniques can be used. These compute consistent object trajectories over time from single-frame detections [1, 2, 3, 5, 8]. Tracking by detection is quite effective, however, its extension from bounding boxes to flexible pose parameterizations is less well developed.

2 Proposed approach

We now describe our approach in detail. Given a video containing T frames, we assume a pose detector is applied to each frame $1 \leq t \leq T$ independently and returns pose estimates $X^t = \{x_1^t, \ldots, x_{n^t}^t | x_i^t \in \mathbb{R}^D\}$ and their associated confidence scores $S^t = \{s_1^t, \ldots, s_{n^t}^t | s_i^t \in \mathbb{R}\}$, where n^t is the number of estimates in frame t. Estimates x_i^t are parametrized using D dimensions, which vary depending on the task, and may include angular values. The goal is to compute trajectories $Y^t = \{y_1^t, \ldots, y_K^t | y_k^t \in \mathbb{R}^D\}$ that are close to the raw pose estimates in each frame but also smooth over time. Here K is the number of objects present, which needs to be estimated.

2.1 Single-frame

We start by discussing how to merge multiple pose estimates in a single frame (fixed t) assuming that the number of objects K is known. The core of our approach is to reduce the problem to a robust clustering of the raw pose estimates which results in a more accurate estimate of each object's pose while simultaneously solving the correspondence problem.

Let $d(x,y) = \|x-y\|_2^2$ be the squared Euclidean distance. We define the loss of predictions $Y^t = \{y_k^t \in \mathbb{R}^D, 1 \leq k \leq K\}$ in frame t given X^t and S^t as:

$$L_{space}(Y^t) = \frac{1}{s^t} \sum_{i=1}^{n^t} \min_k d(x_i^t, y_k^t) s_i^t, \quad \text{where } s^t = \sum_{i=1}^{n^t} s_i^t. \tag{1}$$

L_{space} encourages the predictions y_k^t to be close to pose estimates x_i^t. One shortcoming of the above loss function is that a single prediction y_k^t can account for x_i^t that are fairly far apart (and conversely that distant x_i^t can affect y_k^t). y_k^t should be able to account for a large number of nearby detections but not any distant ones. We can modify the loss function accordingly by defining a bounded distance measure $d_{bd}(x,y) = \min(z, \|x-y\|_2^2)$. d_{bd} is like the squared Euclidean distance except it attains a maximum value of z. The resulting loss is:

$$L_{space}(Y^t) = \frac{1}{s^t} \sum_{i=1}^{n^t} \min_k d_{bd}(x_i^t, y_k^t) s_i^t. \tag{2}$$

The above is the same as the loss in Eqn. (1) except d has been replaced by d_{bd}. Now, once an estimate x_i^t is far enough from any location y^t, it simply attains the maximum penalty z

and no longer affects y_k^t. The constant z is application dependent. In practice we always set z to the average object width in pixels.

We now discuss intuition behind the optimizing procedure for L_{space}. First, consider the first loss given in Eqn. (1). Note that this is precisely the loss of (weighted) **k-means** clustering with $k = K$ and Y^t being the cluster centers. In other words, if we assume K is known, we could run weighted k-means giving us a reasonable solution for a single frame t. However, upon replacing d with the bounded distance d_{bd}, k-means is no longer applicable.

We describe a simple variation of k-means (which we call **bounded k-means**) for the loss defined in Eqn. (2). We drop the t superscript for the following discussion. In every phase of k-means, first cluster memberships are determined, and then each cluster center is set to the mean of the points belonging to the cluster. In standard k-means, a point x_i belongs to cluster k if $d(x_i,y_k) < d(x_i,y_j), \forall j \neq k$. In bounded k-means, we use the same update procedure, except x_i belongs to cluster k if both $d(x_i,y_k) < d(x_i,y_j), \forall j \neq k$ and $d(x_i,y_k) < z$.

It is well known that given a set of points x_i with associated weights w_i, setting $\mu = \sum_i w_i x_i / \sum_i w_i$ minimizes $\sum_i w_i \|x_i - \mu\|_2^2 = \sum_i w_i d(x_i,\mu)$. It is this property that results in a decrease of the loss in each phase of k-means. The above no longer holds for d_{bd}. However, suppose that for a given y, $d(x_i,y) \leq z \; \forall i$. Now, it is simple to show that using the weighted mean μ of x_i, d_{db} has the property that: $\sum_i w_i d_{bd}(x_i,\mu) \leq \sum_i w_i d_{bd}(x_i,y)$. Proof:

$$\sum_i w_i \min(z, \|x_i - \mu\|_2^2) \leq \sum_i w_i \|x_i - \mu\|_2^2 \leq \sum_i w_i \|x_i - y\|_2^2 = \sum_i w_i \min(z, \|x_i - y\|_2^2). \tag{3}$$

The last step follows because we only considered y for which $d(x_i,y) \leq z$.

In other words, replacing y with the weighted mean μ of the x_i that are within a distance of z of y decreases the loss (or keeps it constant), resulting in a decrease of the loss in each phase. Note, however, that μ is not guaranteed to be the true minimum of $\sum_i w_i d_{bd}(x_i,\mu)$; it is only guaranteed to be superior to any y with $d(x_i,y) \leq z \; \forall i$. Hence, although each step of bounded k-means is guaranteed to decrease the loss, it is not guaranteed to be optimal (in standard k-means each step is optimal although the alternating optimization procedure is not). Running bounded k-means with different initializations alleviates this problem.

Finally note that if the pose includes angular data, the distance function and optimization procedure need to be further modified, see supplementary material for details.

2.2 Multi-frame

In video, we need to ensure that pose predictions are consistent across frames. To extend the approach discussed above to multiple frames, we add a second term to the loss, encouraging predictions of the same object y_k^t to remain close together between adjacent frames:

$$L_{time}(Y^{t-1}, Y^t) = \frac{1}{K} \sum_{k=1}^{K} d(y_k^{t-1}, y_k^t), \tag{4}$$

where d is again the squared Euclidean distance. Putting things together, the overall loss is:

$$L(Y) = \sum_{t=1}^{T} L_{space}(Y^t) + \lambda \sum_{t=2}^{T} L_{time}(Y^{t-1}, Y^t). \tag{5}$$

λ is user specified and controls the amount of temporal smoothing. Setting $\lambda = 1$ results in the spatial and temporal terms receiving about equal importance.

We now discuss the full optimization procedure for the loss in Eqn. (5). Given an initial solution Y, we iteratively refine Y at a single frame t in a manner that guarantees the loss $L(Y)$ will decrease. For notational simplicity assume $1 < t < T$. We can rewrite Eqn. (5) to include only terms that depend on Y^t, while keeping the rest of Y fixed, as:

$$L(Y^t) = \frac{1}{s^t}\sum_{i=1}^{n^t} \min_k d_{bd}(x_i^t, y_k^t) s_i^t + \lambda \frac{1}{K}\sum_{k=1}^{K}\left(d(y_k^t, y_k^{t-1}) + d(y_k^t, y_k^{t+1})\right). \tag{6}$$

Next, we replace $\min_k$ with an assignment a_{ik}^t and d_{bd} with d:

$$L'(Y^t) = \frac{\bar{s}^t}{s^t} z + \frac{1}{s^t}\sum_{i=1}^{n^t} a_{ik}^t d(x_i^t, y_k^t) s_i^t + \lambda \frac{1}{K}\sum_{k=1}^{K}\left((d(y_k^t, y_k^{t-1}) + d(y_k^t, y_k^{t+1})\right) \tag{7}$$

$$\text{where } a_{ik}^t = \mathbf{1}\left[d(x_i^t, y_k^t) \leq d(x_i^t, y_j^t)\ \forall j \text{ and } d(x_i^t, y_k^t) < z\right]. \tag{8}$$

In the above $\mathbf{1}$ is the indicator function and $\bar{s}^t$ is the sum of scores of all points x_i^t not assigned to any cluster. As the assignments a_{ik}^t are now fixed, L' serves as an upper bound on L, that is $L(Y^t) \leq L'(Y^t)$. Instead of optimizing $L(Y^t)$ directly w.r.t. Y^t, which is difficult because of the nonlinearity introduced by the min operator, we instead optimize the upper bound $L'(Y^t)$.

Although we omit details here, $L'(Y^t)$ can be easily re-written in the following form:

$$L'(Y^t) = \sum_k \sum_j \tilde{s}_j^k d(y_k^t, \tilde{x}_j^k), \tag{9}$$

where $\tilde{s}_j^k$ and $\tilde{x}_j^k$ are set appropriately. Once in this form, we can compute:

$$y_k^t = \sum_j \tilde{s}_j^k \tilde{x}_j^k \Big/ \sum_j \tilde{s}_j^k, \tag{10}$$

which is guaranteed to minimize $L'(Y^t)$. For further intuition behind the approach, and its relation to k-means, we refer readers to Sec. 2.1. The above gives us a procedure to start with any random solution Y and improve it gradually one frame at a time. We alternate iterating forward (optimizing from $t = 1$ to $t = T$) and backward (from $t = T$ to $t = 1$) until convergence (typically a few passes suffices). To avoid local minima, several restarts with random initializations are performed, similarly to standard k-means.

2.3 Variable *K*

Our approach can be extended to automatically estimate the number of objects K present in a given image or video by iteratively estimating one trajectory at a time. We set $K = 1$ and use the approach described in Sec. 2.2 to find the single best pose trajectory given X and S. Then, all estimates x_i^t and corresponding scores s_i^t near the returned trajectory Y are removed and the method iterates, finding a single trajectory at each round and stopping when the average number of remaining estimates is less than 1 per frame. Estimates are removed if $d(x_i^t, y^t) < z$. We refer to our full approach as Pose-NMS. Our Matlab code runs between 10-25 fps on a standard 3.4Ghz CPU, depending on K and D. Source code is available online.

Pose-NMS is a versatile approach. It can be used to merge pose estimates in single images or in short sequences, controlling the desired amount of temporal consistency through parameter λ. In scenarios where the number of objects is fixed over extended periods of time (*e.g.* animals in a cage), it can be used to perform joint optimization over $K > 1$, resulting in an effective 'tracking by repeated pose estimation' approach. For more classical tracking scenarios (variable number of objects entering/leaving the scene), the iterative method described above can be used to find all relevant trajectories in short sequences.

3 Results

We collected 1,000 short clips to benchmark our approach on three different tasks: human bodies (Sec. 3.1), human face landmarks (Sec. 3.2), and mice (Sec. 3.3). We manually annotated pose on the last frame of each clip to measure how much pose estimation gets improved on frame T given the previous $T-1$ frames. Clip lengths vary from 1-10s.

To measure performance, we decouple object detection from pose estimation since pose estimation can only be correct if the object has been correctly detected. For detection, we convert pose estimates back to bounding boxes (biggest bounding box containing all object parts) and use the standard PASCAL [15] criteria which considers as true positives all detection bounding boxes overlapping ground-truth bounding boxes by more than 0.5. Then, we report pose estimation quality **only** on the objects correctly detected. Note that reported pose quality will depend on detector recall and cannot be fully separated from detection accuracy.

We briefly discuss tested approaches next:

- **NMS**: Standard NMS scheme based on bounding boxes [25]. Does not take into account pose or temporal information during non-maximum suppression.
- **TrackByDetect**: NMS on bounding boxes followed by temporal smoothing of the remaining pose estimates using Pose-NMS. Serves as a stronger baseline due to use of temporal and pose information; however, less information is available for Pose-NMS due to the initial NMS step. Related to standard tracking-by-detection schemes.
- **Pose-NMS (T=1)**: Single frame variant of Pose-NMS. Applies to images or videos.
- **Pose-NMS**: Our full approach. In all cases, during multi-frame optimization we used $\lambda = 1$, which results in a good trade-off between detection accuracy and pose trajectory smoothness. All other parameters are kept constant unless otherwise noted.

3.1 Buffy stickmen

The Buffy Stickmen dataset [14, 16] is one of the most widely used datasets for human body pose estimation. Pose is encoded as the beginning/end points of 5 body parts (head, shoulder, elbow, wrist and hip), converting humans into 'stick' figures, see Fig. 2(c). All frames are obtained from videos of a TV show; however, the original dataset does not contain any temporal information. Instead, we use it to compare standard NMS with our single-frame approach (T=1). In order to benchmark on video, we extend the Buffy Stickmen dataset by collecting 50 short clips using the same episodes as the original test set.

For the pose estimation method, we use the deformable part model from Yang *et al.* [24, 25], the current state-of-the-art on the Buffy dataset. We downloaded the latest version of the code directly from the author's website and trained it on the training set using the default settings. The original approach has an NMS step in which all redundant pose candidates are removed. We replace this step by our approach and compare results. Yang *et al.* [25] also provided ground-truth bounding boxes to be able to report detection performance correctly.

In [25], a new pose quality metric called *Percentage of Correct Keypoints (PCK)* is proposed to improve the original *Percentage of Correct Parts (PCP)* [16]. PCK is similar to the protocol used in the PASCAL Person Layout Challenge [15]. It assumes that the person bounding box is given at test time and considers a keypoint as correctly detected if it falls within $0.2 \cdot \max(h,w)$ pixels of the ground-truth keypoint, where h and w are the height and width of the image. To avoid assumptions about object locations being given at test time, we only report PCK back on the people correctly detected, where detection accuracy is computed from bounding boxes as explained before.

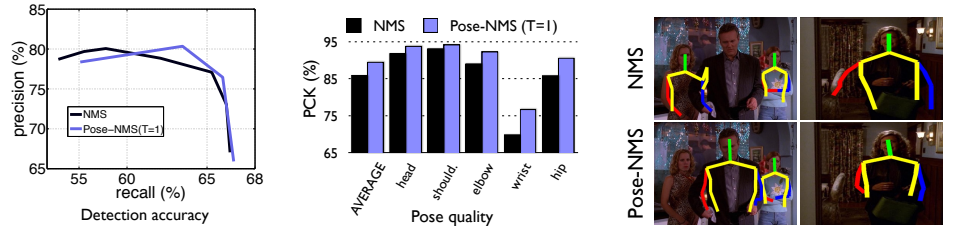
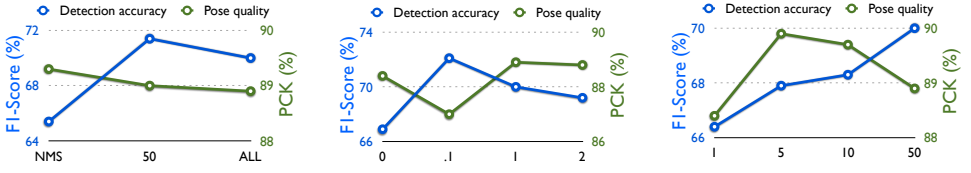


Figure 2: Results on (static) Buffy Stickmen dataset. Pose-NMS performs slightly better for detection but consistently improves the quality of pose estimates around 5%.



(a) Amount of prior NMS (b) Temporal smoothness λ. (c) Temporal information T.

Figure 3: Parameter trade-offs in terms of detection accuracy (F1-Score, blue) and pose estimation quality (PCK, green) on the Video Buffy dataset. (a) Keeping all pose candidates (no NMS) prior to applying Pose-NMS improves detection accuracy while maintaining similar pose quality. (b,c) Increasing temporal smoothing λ and the amount of prior temporal information T trades-off detection accuracy and pose quality. We kept all original pose candidates and used $\lambda = 1$ and $T = 50$ in all reported experiments.

3.1.1 Original Buffy (no temporal information)

The testing part of the dataset consists of 276 pose-annotated video frames over 3 episodes of the Buffy TV show. Fig. 2 shows the results of our single-frame approach (T=1) against standard NMS. Pose-NMS reaches 3% higher precision at similar recall rates. More importantly, it consistently improves the quality of pose estimates by more than 5% on all body parts. This shows that Pose-NMS, unlike standard NMS, is capable of generating a final set of pose estimates that are more accurate than any of the original ones.

3.1.2 Video Buffy

To better illustrate how our approach can improve results when full temporal information is available, we extended the original Buffy Stickmen dataset with video. We collected all uncut scenes with a duration longer than 2s (50 frames) containing people standing from the same 3 episodes used for testing in original dataset, resulting in 50 clips.

Pose-NMS significantly improves detection accuracy while maintaining similar pose quality when compared with running NMS prior to our multi-frame optimization, see Fig. 3(a). This shows the benefits of our approach compared with standard techniques. Strengthening temporal consistency by increasing λ beyond $\lambda = .1$ decreases detection accuracy (recall decreases as objects that are not consistently detected across frames are suppressed), but increases pose estimation quality, see Fig. 3(b). The effect of varying number of frames T for Pose-NMS is shown in Fig. 3(c). In contrast to increasing λ, increasing the available temporal information improves detection accuracy (all observed objects get included) but somewhat degrades pose quality.

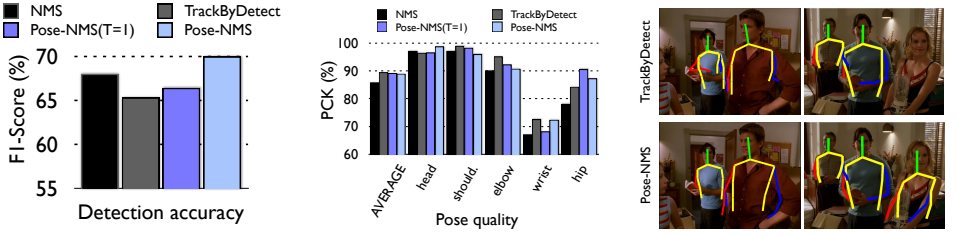


Figure 4: Results on the Video Buffy dataset. Single-frame Pose-NMS improves pose quality around ~5% at similar detection accuracy compared with standard NMS. Full (multi-frame) Pose-NMS improves detection accuracy 6% and maintains similar pose quality compared to running NMS prior to our tracking phase.

Fig. 4 shows the full results on the Video Buffy dataset. Pose-NMS with (T=1) improves pose quality around ~5% at somewhat lower detection accuracy compared with standard NMS. Full Pose-NMS improves detection accuracy 6% and maintains similar pose quality compared to TrackByDetect. Adding multi-frame reasoning to Pose-NMS significantly improves detection accuracy while achieving similar pose quality.

3.2 Faces

We next test the performance of Pose-NMS on face landmark estimation 'in the wild'. Since to the best of our knowledge there is no currently available 'in the wild' face landmark dataset that includes video, we collected a new dataset. We downloaded 33 HD movies shot on the streets of 23 different countries from YouTube from the series '50 people one question', where 50 random people are selected and interviewed. They represent a realistic and challenging benchmark for face landmark estimation due to the variety of filming conditions, locations and people's expressions. From these 33 ten minute films we extracted 450 clips with durations varying between 1 and 10 seconds and annotated face landmarks on the final frame of each clip using 29 keypoints as in the *LFPW* dataset [2].

For the face landmark estimation method we use Cascaded Pose Regression (CPR), first introduced in [3] and later extended to be the current state-of-the art on several face datasets [7]. As code is not publicly available, we reimplemented the method ourselves, and trained it using the 2K faces from the training set of the HELEN dataset [19]. Since this method requires the face bounding box to be previously detected, we trained a face detector using code from a state-of-the-art pedestrian detector [12] and 4K faces from the Multi-PIE [17] and HELEN [19] datasets.

The usual procedure for CPR given a single image is to initialize it from the most confident bounding box returned by the face detector *after* NMS. Instead, when applying our approach, we run CPR starting from each bounding box detected around the object independently and keep all resulting pose estimates.

Fig. 5 shows results. The metric used to report landmark estimation quality is the average distance between all estimated landmarks and ground-truth landmarks, normalized with respect to the interocular distance. Pose-NMS improves both detection and pose estimation 2-3%. We only report pose quality of correctly detected faces, as before. The number of pose estimation failures is reduced by 5% (not shown), using 0.1 as the threshold as proposed in [10]. This again demonstrates that Pose-NMS effectively uses all pose candidates to output a more precise final pose estimation.

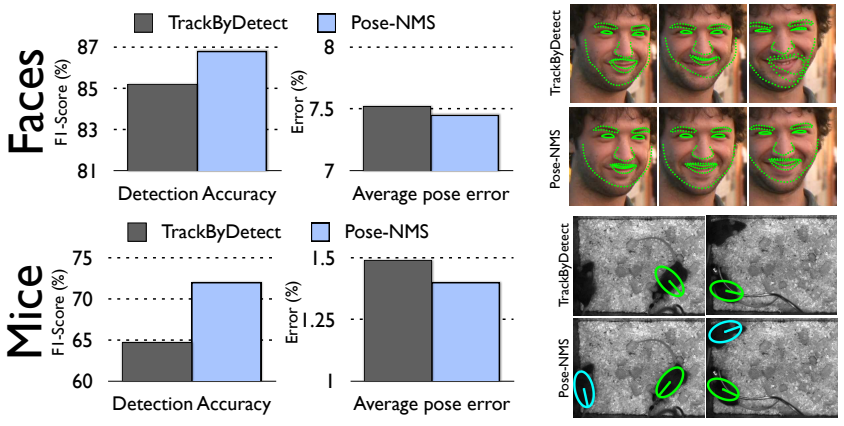


Figure 5: Results on Faces and Mice. Our approach improves detection by 3-7% and pose estimation quality between 2-7%. See text for details.

3.3 Mice

As a third and final task, we test our approach for estimating the pose of two mice. We use videos from the *Caltech Resident-Intruder Mouse (CRIM13)* dataset, published in [6]. These videos represent a challenging task for pose estimation due to the high amount of inter-object occlusions that result from the frequent social interactions between the two mice. We downloaded the 133 ten minute top-view testing videos from which we extracted 550 clips ranging from 1-10s in duration. For each, we annotated the final frame by placing direction sensitive ellipses around each mouse body as in the original work from [13], see Fig. 5.

As for faces, we use Cascaded Pose Regression (CPR) [13] for pose estimation, and code from [12] as the object detector (both trained on frames from the 104 training videos of CRIM13). As before, we run CPR starting from each bounding box around the object independently and keep all resulting pose estimates. The metric used to report performance is the distance between the estimated and ground-truth ellipses, normalized with respect to human annotator's variance, such that human error would be equal to 1, as proposed in [13].

We show results in Fig. 5. Pose-NMS improves both detection accuracy and pose quality by 7%. The number of pose estimation failures are reduced by 7% (using 5 as threshold), demonstrating once again the benefits of our approach.

4 Conclusions

We proposed a principled framework for merging multiple independent pose candidates both in a single frame and across multiple frames by performing a joint optimization. Our appoach generates a final set of pose estimates that are more accurate than any of the original ones. In scenarios where the number of objects is fixed over extended periods of time, Pose-NMS can be used as an effective 'tracking by repeated pose estimation' approach.

Our proposed approach makes minimal assumptions about the underlying pose estimation method resulting in a highly efficient, versatile and effective method. We used it together with two distinct state-of-the art pose estimation approaches on three different pose estimation tasks, showing that it improves both the detection accuracy as well as pose estimation quality. All source code is available online.

Acknowledgments

This work is funded by the Gordon and Betty Moore Foundation and ONR MURI Grant N00014-10-1-0933.

References

[1] M. Andriluka, S. Roth, and B. Schiele. Monocular 3d pose estimation and tracking by detection. In *CVPR*, 2010.

[2] K. Schindler B. Leibe and L. Van Gool. Coupled detection and trajectory estimation for multi-object tracking. In *ICCV*, 2007.

[3] B. Babenko, M.-H. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. *PAMI*, 33(8):1619–1632, 2011.

[4] P. Belhumeur, D. Jacobs, D. Kriegman, and N. Kumar. Localizing parts of faces using a concensus of exemplars. In *CVPR*, 2011.

[5] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *PAMI*, 33(9):1806–1819, 2011.

[6] X.P. Burgos-Artizzu, P. Dollár, D. Lin, D.J. Anderson, and P. Perona. Social behavior recognition in continuous videos. In *CVPR*, 2012.

[7] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. In *CVPR*, 2012.

[8] D. A. Forsyth D. Ramanan, and and A. Zisserman. Tracking people by learning their appearance. *PAMI*, 29(1):65–91, 2007.

[9] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[10] M. Dantone, J. Gall, G. Fanelli, and L. VanGool. Real-time facial feature detection using conditional regression forests. In *CVPR*, 2012.

[11] C. Desai, D. Ramanan, and C.C. Fowlkes. Discriminative models for multi-class object layout. In *ICCV*, 2009.

[12] P. Dollár, S. Belongie, and P. Perona. The fastest pedestrian detector in the west. In *BMVC*, 2010.

[13] P. Dollár, P. Welinder, and P. Perona. Cascaded pose regression. In *CVPR*, 2010.

[14] M. Eichner, M. Marin-Jimenez, A. Zisserman, and V. Ferrari. 2d articulated human pose estimation and retrieval in (almost) unconstrained still images. *IJCV*, 99(2):190–214, 2012.

[15] M. Everingham, L. Van Gool, C. Williams, J. Winn, and Zisserman A. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010.

[16] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *CVPR*, 2008.

[17] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-pie. In *FG*, 2008.

[18] M. Hofmann and D.M. Gavrila. Multi-view 3d human pose estimation combining single-frame recovery, temporal integration and model adaptation. In *CVPR*, 2009.

[19] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang. Interactive facial feature localization. In *ECCV*, 2012.

[20] T. B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *CVIU*, 81(33):231–268, 2001.

[21] V.I. Pavlovic, R. Sharma, and T.S. Huang. Visual interpretation of hand gestures for human-computer interaction: a review. *PAMI*, 19(7):677–695, 1997.

[22] P. Peursum, S. Venkatesh, and G. West. Tracking-as-recognition for articulated full-body human motion analysis. In *CVPR*, 2007.

[23] R. W. Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976–990, 2010.

[24] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, 2011.

[25] Y. Yang and D. Ramanan. Articulated human detection with flexible mixtures of parts. *PAMI*, 2013.

[26] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, 2012.