MoDeep: A Deep Learning Framework Using Motion Features for Human Pose Estimation

Arjun Jain(✉), Jonathan Tompson, Yann LeCun, and Christoph Bregler

New York University, New York, USA
{ajain,tompson,yann,bregler}@cs.nyu.edu

Abstract. In this work, we propose a novel and efficient method for articulated human pose estimation in videos using a convolutional network architecture, which incorporates both color and motion features. We propose a new human body pose dataset, *FLIC-motion* (This dataset can be downloaded from http://cs.nyu.edu/~ajain/accv2014/.), that extends the FLIC dataset [1] with additional motion features. We apply our architecture to this dataset and report significantly better performance than current state-of-the-art pose detection systems.

1 Introduction

Human body pose recognition in video is a long-standing problem in computer vision with a wide range of applications. However, body pose recognition remains a challenging problem due to the high dimensionality of the input data and the high variability of possible body poses. Traditionally, computer vision-based approaches tend to rely on appearance cues such as texture patches, edges, color histograms, foreground silhouettes or hand-crafted local features (such as histogram of gradients (HoG) [2]) rather than motion-based features. Alternatively, psychophysical experiments [3] have shown that motion is a powerful visual cue that alone can be used to extract high-level information, including articulated pose.

Previous work [4,5] has reported that using motion features to aid pose inference has had little or no impact on performance. Simply adding high-order temporal connectivity to traditional models would most often lead to intractable inference. In this work we show that deep learning is able to successfully incorporate motion features and is able to out-perform existing state-of-the-art techniques. Further, we show that by using motion features alone our method outperforms [6–8] (see Fig. 9(a) and (b)), which further strengthens our claim that information coded in motion features is valuable and should be used when available.

This paper makes the following contributions:

- A system that successfully incorporates motion-features to enhance the performance of pose-detection 'in-the-wild' compared to existing techniques.
- An efficient and tractable algorithm that achieves close to real-time frame rates, making our method suitable for wide variety of applications.

D. Cremers et al. (Eds.): ACCV 2014, Part II, LNCS 9004, pp. 302–315, 2015.
DOI: 10.1007/978-3-319-16808-1_21

- A new dataset called **FLIC-motion**, which is the FLIC dataset [1] augmented with 'motion-features' for each of the 5003 images collected from Hollywood movies.

2 Prior Work

Geometric Model Based Tracking: One of the earliest works on articulated tracking in video was Hogg [9] in 1983 using edge features and a simple cylinder based body model. Several other model based articulated tracking systems have been reported over the past two decades, most notably [10–16]. The models used in these systems were explicit 2D or 3D jointed geometric models. Most systems had to be hand-initialized (except [12]), and focused on incrementally updating pose parameters from one frame to the next. More complex examples come from the HumanEva dataset competitions [17] that use video or higher-resolution shape models such as SCAPE [18] and extensions. We refer the reader to [19] for a complete survey of this era. Most recently such techniques have been shown to create very high-resolution animations of detailed body and cloth deformations [20–22]. Our approach differs, since we are dealing with single view videos in unconstrained environments.

Statistical Based Recognition: One of the earliest systems that used no explicit geometric model was reported by Freeman et al. in 1995 [23] using oriented angle histograms to recognize hand configurations. This was the precursor for the bag-of-features, SIFT [24], STIP [25], HoG, and Histogram of Flow (HoF) [26] approaches that boomed a decade later, most notably including the work by Dalal and Triggs in 2005 [2]. Different architectures have since been proposed, including "shape-context" edge-based histograms from the human body [27,28] or just silhouette features [29]. Shakhnarovich et al. [30] learn a parameter sensitive hash function to perform example-based pose estimation. Many techniques have been proposed that extract, learn, or reason over entire body features, using a combination of local detectors and structural reasoning (see [31] for coarse tracking and [32] for person-dependent tracking).

Though the idea of using "Pictorial Structures" by Fischler and Elschlager [33] has been around since the 1970s, matching them efficiently to images has only been possible since the famous work on 'Deformable Part Models' (DPM) by Felzenszwalb et al. [34] in 2008. Many algorithms that use DPM for creating the body part unary distribution [6,7,35,36] with spatial-models incorporating body-part relationship priors have since then been developed. Johnson and Everingham [37], who also proposed the 'Leeds Sports Database', employ a cascade of body part detectors to obtain more discriminative templates. Almost all best performing algorithms since have solely built on HoG and DPM for local evidence, and yet more sophisticated spatial models. Pishchulin [38] proposes a model that augments the DPM unaries with *Poselet* conditioned [39] priors. Sapp and Taskar [1] propose a model where they cluster images in the pose-space and then find the mode which best describes the input image. The pose of this mode then acts as a strong spatial prior, whereas the local evidence is again based on HoG and gradient features. Following the *Poselets* approach [39], the *Armlets* approach by Gkioxari et al. [40] incorporates edges, contours, and color histograms in addition to the HoG features. They employ a semi-global classifier for part configuration and show good performance on real-world data. However, they only show their results on arms. The major drawback of all these approaches is that both the local evidence and the global structure is hand crafted, whereas we jointly learn both the local features and the global structure using a multi-resolution convolutional network.

Shotton et al. [41] use an ensemble of random trees to perform per-pixel labeling of body parts in depth images. As a means of reducing overall system latency and avoiding repeated false detections, their work focuses on pose inference using only a single depth image. By contrast, we extend the single frame requirement to at least 2 frames (which we show considerably improves pose inference), and our input domain is unconstrained RGB images rather than depth.

Pose Detection Using Image Sequences:

Deep Learning Based Techniques: Recently, state-of-the-art performance has been reported on many vision tasks using deep learning algorithms [42–47]. References [48–50] also apply neural networks for pose recognition, specifically Toshev et al. [48] show better than state-of-the-art performance on the 'FLIC' and 'LSP' [51] datasets. In contrast to Toshev et al., in our work we propose a translation invariant model which improves upon their method, especially in the high-precision region.

3 Body-Part Detection Model

We propose a Convolutional Network (ConvNet) architecture for the task of estimating the 2D location of human joints in video (Sect. 3.2). The input to the network is an RGB image and a set of *motion features*. We investigate a wide variety of motion feature formulations (Sect. 3.1). Finally, we will also introduce a simple Spatial-Model to solve a specific sub-problem associated with evaluation of our model on the FLIC-motion dataset (Sect. 3.3).

3.1 Motion Features

The aim of this section is to incorporate features that are representative of the true *motion-field* (the perspective projection of the 3D velocity-field of moving surfaces) as input to our detection network so that it can exploit motion as a cue for body part localization. To this end, we evaluate and analyze four motion features which fall under two broad categories: those using simple derivatives of the RGB video frames and those using optical flow features. For each RGB image pair f_i and $f_{i+\delta}$, we propose the following features:

- RGB image pair - $\{f_i, f_{i+\delta}\}$
- RGB image and an RGB difference image - $\{f_i, f_{i+\delta} - f_i\}$

- Optical-flow[1] vectors - $\{f_i, \mathrm{FLOW}(f_i, f_{i+\delta})\}$
- Optical-flow magnitude - $\{f_i, ||\,\mathrm{FLOW}(f_i, f_{i+\delta})||_2\}$

The RGB image pair is by far the simplest way of incorporating the relative motion information between the two frames. However, this representation clearly suffers from a lot of redundancy (i.e. if there is no camera movement) and is extremely high dimensional. Furthermore, it is not obvious to the deep network what changes in this high dimensional input space are relevant temporal information and what changes are due to noise or camera motion. A simple modification to this representation is to use a difference image, which reformulates the RGB input so that the algorithm sees directly the pixel locations where high energy corresponds to motion (alternatively the network would have to do this implicitly on the image pair). A more sophisticated representation is optical-flow, which is considered to be a high-quality approximation of the true *motion-field.* Implicitly learning to infer optical-flow from the raw RGB input would be non-trivial for the network to estimate, so we perform optical-flow calculation as a pre-processing step (at the cost of greater computational complexity).

FLIC-motion Dataset: We propose a new dataset which we call **FLIC-motion**[2]. It is comprised of the original FLIC dataset of 5003 labeled RGB images collected from 30 Hollywood movies, of which 1016 images are held out as a test set, augmented with the aforementioned motion features.

We experimented with different values for δ and investigated the above features with and without camera motion compensation; we use a simple 2D projective motion model between f_i and $f_{i+\delta}$, and warp $f_{i+\delta}$ onto f_i using the inverse of this best fitting projection to approximately remove camera motion. A comparison between image pairs with and without warping can be seen in Fig 1.

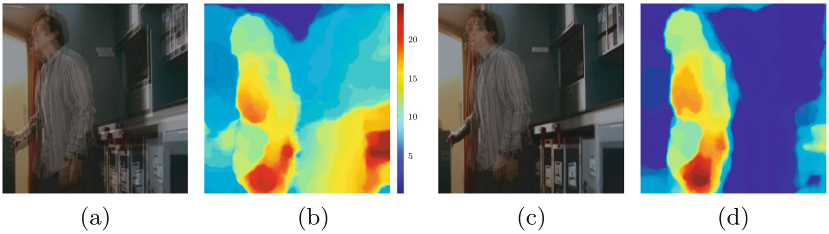


Fig. 1. Results of optical-flow computation: (a) average of frame pair, (b) optical flow on (a), (c) average of frame pair after camera compensation, and (d) optical-flow on (c)

To obtain $f_{i+\delta}$, we must know where the frames f_i occur in each movie. Unfortunately, this was non-trivial as the authors Sapp et al. [1] could not provide

[1] We use the algorithm proposed by Weinzaepfel et al. [47] to compute optical-flow.

[2] This dataset can be downloaded from http://cs.nyu.edu/~ajain/accv2014/.

us with the exact version of the movie that was used for creating the original dataset. Corresponding frames can be very different in multiple versions of the same movie (4:3 vs wide-screen, director's cut, special editions, etc.). We estimate the best similarity transform S between f_i and each frame f_j^m from the movie m, and if the distance $|f_i - Sf_j^m|$ is below a certain threshold (10 pixels), we conclude that we found the correct frame. We visually confirm the resulting matches and manually pick frames for which the automatic matching was unsuccessful (e.g. when enough feature points were not found).

3.2 Convolutional Network

Recent work [48,49] has shown ConvNet architectures are well suited for the task of human body pose detection, and due to the availability of modern Graphics Processing Units (GPUs), we can perform Forward Propagation (FPROP) of deep ConvNet architectures at interactive frame-rates. Similarly, we realize our detection model as a deep ConvNet architecture. The input is a 3D tensor containing an RGB image and its corresponding motion features, and the output is a 3D tensor containing *response-maps*, with one response-map for each joint. Each response-map describes the per-pixel energy for the presence of the corresponding joint at that pixel location.

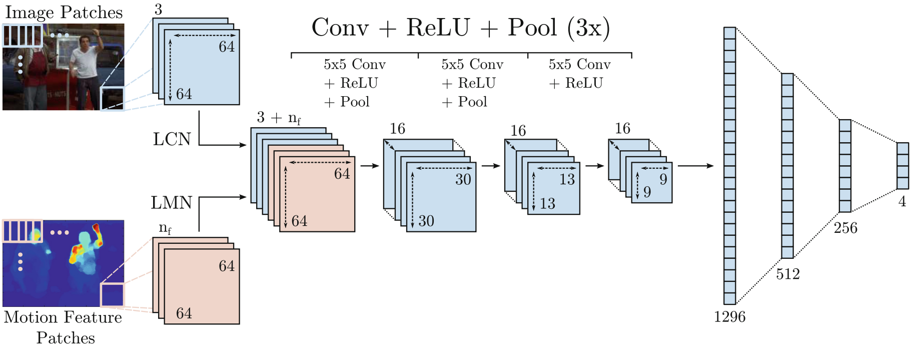


Fig. 2. Sliding-window with image and flow patches

Our ConvNet is based on a *sliding-window* architecture. A simplified version of this architecture is shown in Fig. 2. The input patches are first normalized using Local Contrast Normalization (LCN [52]) for the RGB channels and a new normalization method for the motion features we call *Local Motion Normalization* (LMN). We formulate LMN as the local subtraction with the response from a Gaussian kernel with large standard deviation followed by a divisive normalization. The result is that it removes some unwanted background camera motion as well as normalizing the local intensity of motion (which helps improve network generalization for motions of varying velocity but with similar pose). Prior to processing through the convolution stages, the normalized motion channels are concatenated along the feature dimension with the normalized RGB channels, and the resulting tensor is processed though 3 stages of convolution.

The first two convolution stages use rectified linear units (ReLU) and Max-pooling, and the last stage incorporates a single ReLU layer. The output of the last convolution stage is then passed to a three stage fully-connected neural-network. The network is then applied to all 64 × 64 sub-windows of the image, stepped every 4 pixels horizontally and vertically. This produces a dense response-map output, one for each joint. The major advantage of this model is that the learned detector is translation invariant by construction.

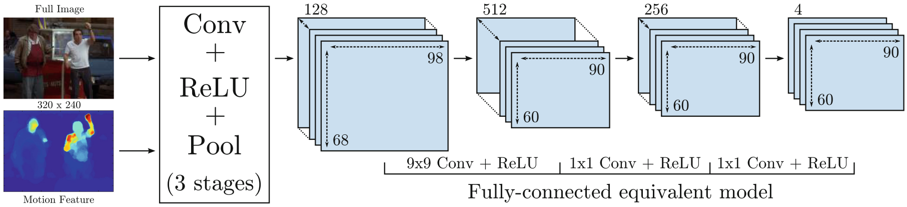


Fig. 3. Efficient sliding window model

Because the layers are convolutional, applying two instances of the network in Fig. 2 to two overlapping input windows leads to a considerable amount of redundant computation. Recent work [53, 54] eliminates this redundancy and thus yields a dramatic speed up. This is achieved by applying each layer of the convolutional network to the entire input image. The fully connected layers for each window are also replicated for all sub-windows of the input. This formulation allows us to back-propagate though this network for all windows simultaneously. Due to the two 2 × 2 subsampling layers, we obtain one output vector every 4 × 4 input pixels. An equivalent efficient version of the sliding window model is shown in Fig. 3.

Note that an alternative model (such as in Tompson et al. [50]) would replace the last 3 convolutional layers with a fully-connected neural network whose input context is the feature activations for the entire input image. Such a model would be appropriate if we knew a priori that there existed a strong correlation between skeletal pose and the position of the person in the input frame since this alternative model is not invariant with respect to the translation of the person within the image. However, the FLIC dataset has no such strong pose-location bias (i.e. a subject's torso is not always in the same location in the image), and therefore a sliding-window based architecture is more appropriate for our task.

We extend the single resolution ConvNet architecture of Fig. 3 by incorporating a *multi-resolution* input. We do so by down-sampling the input (using appropriate anti-aliasing), and then each resolution image is processed through either a LCN or LMN layer using the same normalization kernels for each bank producing an approximate Laplacian pyramid. The role of the Laplacian Pyramid is to provide each bank with non-overlapping spectral content which minimizes

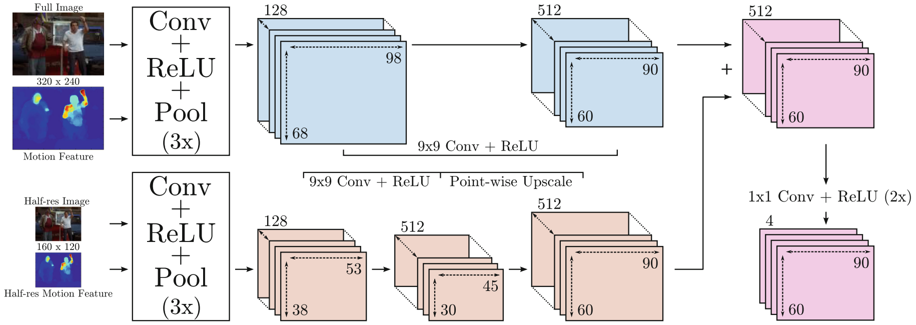


Fig. 4. Multi-resolution efficient sliding window model

network redundancy. Our final, multi-resolution network is shown in Fig. 4. The outputs of the convolution banks are concatenated (along the feature dimension) by point-wise up-scaling of the lower resolution bank to bring the feature maps into canonical resolution. Note that in our final implementation we use 3 resolution banks.

We train the Part-Detector network using supervised learning via Back Propagation and Stochastic Gradient Descent. We minimize a mean squared error criterion for the distance between the inferred response-map activation and a ground truth response-map, which is a 2D Gaussian distribution centered at the target joint location and with small standard deviation (1px). We use Nesterov momentum to reduce training time [55] and we randomly perturb the input images each epoch by randomly flipping and scaling the images to prevent network overtraining and improve generalization performance.

3.3 Simple Spatial Model

Our model is evaluated on our new FLIC-motion dataset (Sect. 3.1). As per the original FLIC dataset, the test images in FLIC-motion may contain multiple people, however, only a single actor per frame is labeled in the test set. As such, a rough torso location of the labeled person is provided at test time to help locate the "correct" person. We incorporate this information by means of a simple and efficient Spatial-Model.

The inclusion of this stage has two major advantages. Firstly, the correct feature activation from the Part-Detector output is selected for the person for whom a ground-truth label was annotated. An example of this is shown in Fig. 5. Secondly, since the joint locations of each part are constrained in proximity to the single ground-truth torso location, then (indirectly) the connectivity between joints is also constrained, enforcing that inferred poses are anatomically viable (i.e. the elbow joint and the shoulder joint cannot be to far away from the torso, which in turn enforces spatial locality between the elbow and shoulder joints).

The core of our Spatial-Model is an empirically calculated *joint-mask*, shown in Fig. 5(b). The joint-mask layer describes the possible joint locations, given

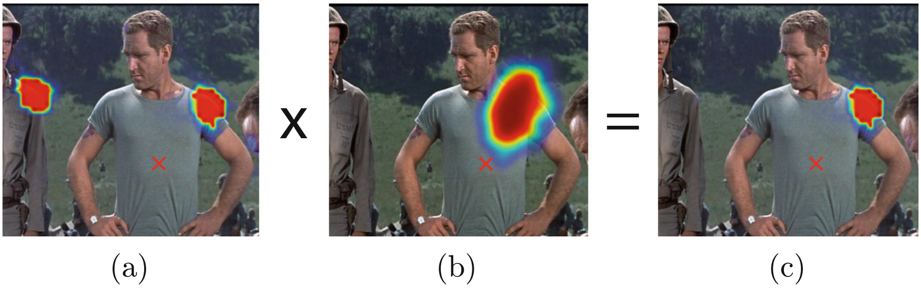


Fig. 5. Simple spatial model used to mask-out the incorrect shoulder activations given a 2D torso position

that the supplied torso position is in the center of the mask. To create a mask layer for body part A, we first calculate the empirical histogram of the part A location, x_A, relative to the torso position x_T for the training set examples; i.e. $x_{\text{hist}} = x_A - x_T$. We then turn this histogram into a Boolean mask by setting the mask amplitude to 1 for pixels for which $p\left(x_{\text{hist}}\right) > 0$. Finally, we blur the mask using a wide Gaussian low-pass filter which accounts for body part locations not represented in the training set (but which might be present in the test set).

During test time, this joint-mask is shifted to the ground-truth torso location and the per-pixel energy from the Part-Model (Sect. 3.2) is then multiplied with the mask to produce a filtered output. This process is carried out for each body part independently.

It should be noted that while this Spatial-Model does enforce some anatomic consistency, it does have limitations. Notably, we expect it to fail for datasets where the range of poses is not as constrained as the FLIC dataset (which is primarily front facing and standing up poses).

4 Results

Training time for our model on the FLIC-motion dataset (3957 training set images, 1016 test set images) is approximately 12 hours, and FPROP of a single image takes approximately 50 ms[3]. For our models that use optical flow as a motion feature input, the most expensive part of our pipeline is the optical flow calculation, which takes approximately 1.89 s per image pair. (We plan to investigate real-time flow estimations in the future).

Section 4.1 compares the performance of the motion features from Sect. 3.1. Section 4.2 compares our architecture with other techniques and shows that our system significantly outperforms existing state-of-the-art techniques. Note that for all experiments in Sect. 4.1 we use a smaller model with 16 convolutional features in the first 3 layers. A model with 128 instead of 16 features for the first 3 convolutional layers is used for results in Sect. 4.2.

[3] Analysis of our system was on a 12 core workstation with an NVIDIA Titan GPU.

4.1 Comparison and Analysis of Proposed Motion Features

Figure 6 shows a selection of example images from the FLIC test set which highlights the importance of using motion features for body pose detection. In Fig. 6(a), the elbow position is occluded by the actor's sling, and no such examples exist in the training set; however, the presence of body motion provides a strong cue for elbow location. Figure 6(b) and (d) have extremely cluttered backgrounds and the correct joint location is locally similar to the surrounding region (especially for the camouflaged clothing in Fig. 6(d)). For these images, motion features are essential in correct joint localization. Finally, Fig. 6(c) is an example where motion blur (due to fast joint motion) reduces the fidelity of RGB edge features, which results in incorrect localization when motion features are not used.

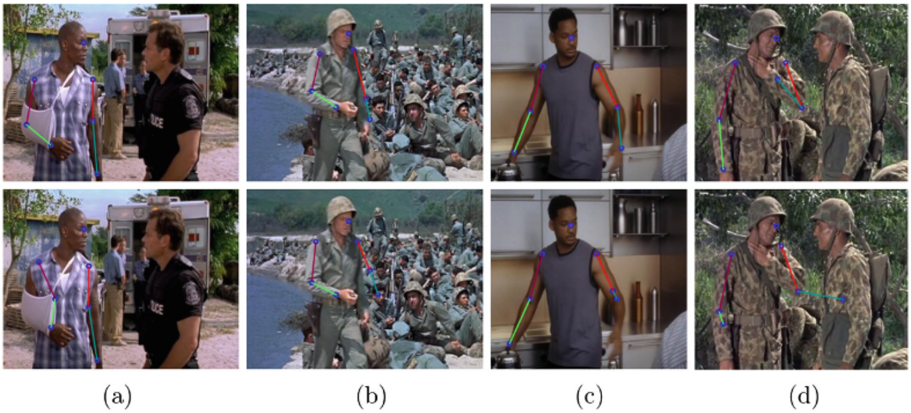
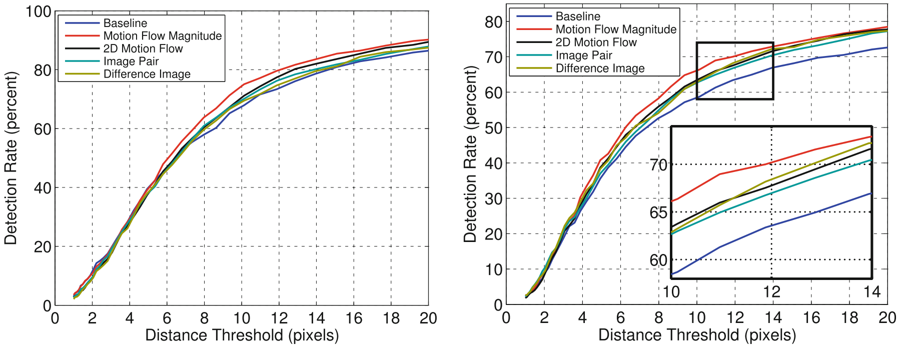


Fig. 6. Predicted joint positions on the FLIC test-set. Top row: detection with motion features (L2 motion flow). Bottom row: without motion features (baseline).

Figure 7(a) and (b) show the performance of the motion features of Sect. 3.1 on the FLIC-motion dataset for the Elbow and Wrist joints respectively. For evaluating our test-set performance, we use the criterion proposed by Sapp et al. [1]. We count the percentage of the test-set images where joint predictions are within a given radius that is normalized to a 100 pixel torso size. Surprisingly, even the simple frame-difference temporal feature improves upon the baseline result (which we define as a single RGB frame input) and even outperforms the 2D optical flow input (see Fig. 6(b) inset).

Note that stable and accurate calculation of optical-flow from arbitrary RGB videos is a very challenging problem. Therefore, incorporating motion flow features as input to the network adds non-trivial localization cues that would be very difficult for the network to learn internally with limited learning capacity. Therefore, it is expected that the best performing networks in Fig. 7 are those that incorporate motion flow features. However, it is surprising that using the magnitude of the flow vectors performs as well as - and in some cases outperforms - the full 2D motion flow. Even though the input data is richer, we hypothesize that when using 2D flow vectors the network must learn invariance to the direction of joint movement; for instance, the network should predict the same head position whether a person is turning his/her head to the left or right on the next frame. On the other hand, when the L2 magnitude of the flow vector is used, the network sees the high velocity motion cue but cannot over-train to the direction of the movement.



(a) FLIC-motion: Elbow (b) FLIC-motion: Wrist

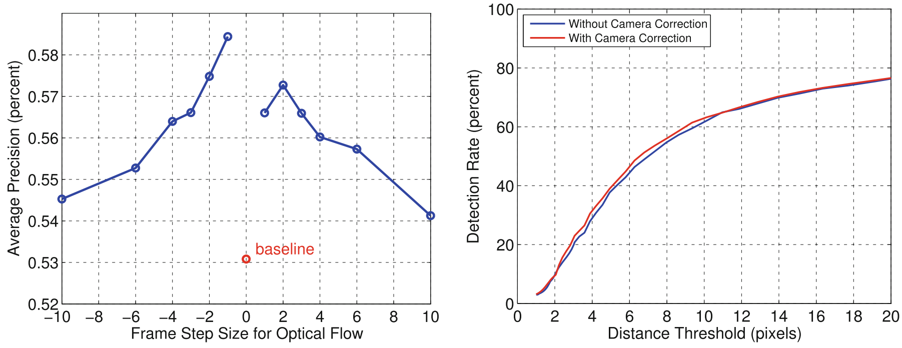
Fig. 7. Model performance for various motion features

Figure 8(a) shows that the performance of our network is relatively agnostic to the frame separation (δ) between the samples for which we calculate motion flow; the average precision between 0 and 20 pixel radii degrades 3.9 % from -10 pixels offset to -1 pixel offset. A frame difference of 10 corresponds to approximately 0.42 s (at 24fps), and so we expect that large motions over this time period would result in complex non-linear trajectories in input space for which a single finite difference approximation of the pixel velocity would be inaccurate. Accordingly, our results show that performance indeed degrades as a larger frame step is used.

Similarly, we were surprised that our camera motion compensation technique (described in Sect. 3.1) does not help to the extent that we expected, as shown in Fig. 8(b). Likely this is because either LMN removes a lot of constant background motion or the network is able to learn to ignore the remaining foreground-background parallax motion due to camera movement.

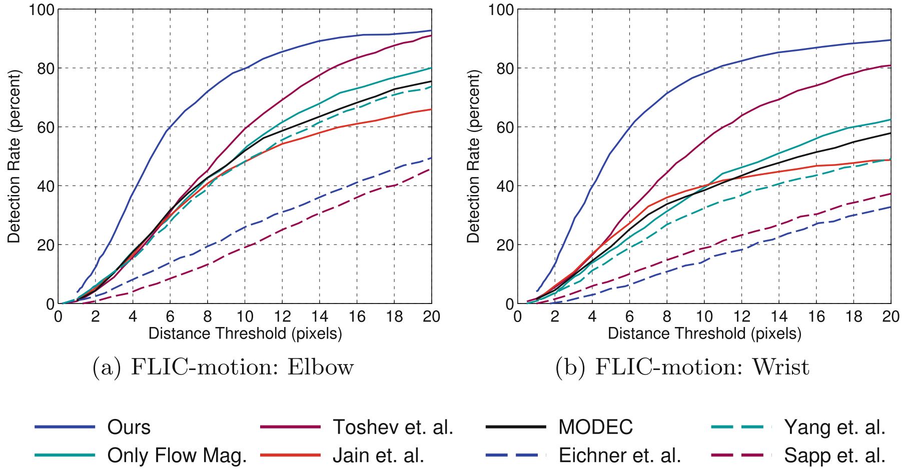
4.2 Comparison with Other Techniques

Figure 9(a) and (b) compares the performance of our system with other state-of-the-art models on the FLIC dataset for the elbow and wrist joints respectively.



(a) FLIC-motion: Wrist (b) FLIC-motion: Wrist

Fig. 8. Model performance for (a) varying motion feature frame offsets (b) with and without camera motion compensation



(a) FLIC-motion: Elbow (b) FLIC-motion: Wrist

Fig. 9. Our model performance compared with our model using only flow magnitude features (no RGB image), Toshev et al. [48], Jain et al. [49], MODEC [1], Eichner et al. [6], Yang et al. [7] and Sapp et al. [8].

Our detector is able to significantly outperform all prior techniques on this challenging dataset. Note that using only motion features already outperforms [6–8]. Also note that using only motion features is less accurate than using a combination of motion features and RGB images, especially in the high accuracy region. This is because fine details such as eyes and noses are missing in motion features. Toshev et al. [48] suffers from inaccuracy in the high-precision region, which we attribute to inefficient direct regression of pose vectors from images. MODEC [1], Eichner et al. [6] and Sapp et al. [8] build on hand crafted HoG features. They all suffer from the limitations of HoG (i.e. they all discard color information, etc.). Jain et al. [49] do not use multi-scale information and evaluate their model in a sliding window fashion, whereas we use the 'one-shot' approach. Finally, we believe that increasing the complexity of our simple spatial model will improve performance of our model, specifically for large radii.

5 Conclusion

We have shown that when incorporating both RGB and motion features in our deep ConvNet architecture, our network is able to outperform existing state-of-the-art techniques for the task of human body pose detection in video. We have also shown that using motion features alone can outperform some traditional algorithms [6–8]. Our findings suggest that even very simple temporal cues can greatly improve performance with a very minor increase in model complexity. As such, we suggest that future work should place more emphasis on the correct use of motion features. We would also like to further explore higher level temporal features, potentially via learned spatiotemporal convolution stages and we hope that using a more expressive temporal-spatial model (using motion constraints) will help improve performance significantly.

Acknowledgments. The authors would like to thank Tyler Zhu for his help with the data-set creation. This research was funded in part by the Office of Naval Research ONR Award N000141210327.

References

1. Sapp, B., Taskar, B.: Modec: multimodal decomposable models for human pose estimation. In: CVPR (2013)
2. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
3. Johansson, G.: Visual perception of biological motion and a model for its analysis. Percept. Psychophys. **14**, 201–211 (1973)
4. Ferrari, V., Marin-Jimenez, M., Zisserman, A.: Progressive search space reduction for human pose estimation. In: CVPR (2008)
5. Weiss, D., Sapp, B., Taskar, B.: Sidestepping intractable inference with structured ensemble cascades. In: NIPS (2010)
6. Eichner, M., Ferrari, V.: Better appearance models for pictorial structures. In: BMVC (2009)
7. Yang, Y., Ramanan, D.: Articulated pose estimation with flexible mixtures-of-parts. In: CVPR (2011)
8. Sapp, B., Toshev, A., Taskar, B.: Cascaded models for articulated pose estimation. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part II. LNCS, vol. 6312, pp. 406–420. Springer, Heidelberg (2010)
9. Hogg, D.: Model-based vision: a program to see a walking person. Image Vis. Comput. **1**, 5–20 (1983)
10. Rehg, J.M., Kanade, T.: Model-based tracking of self-occluding articulated objects. In: Computer Vision (1995)

11. Kakadiaris, I.A., Metaxas, D.: Model-based estimation of 3d human motion with occlusion based on active multi-viewpoint selection. In: CVPR (1996)
12. Wren, C.R., Azarbayejani, A., Darrell, T., Pentland, A.P.: Pfinder: Real-time tracking of the human body. IEEE Trans. Pattern Anal. Mach. Intell. **19**, 780–785 (1997)
13. Bregler, C., Malik, J.: Tracking people with twists and exponential maps. In: CVPR (1998)
14. Deutscher, J., Blake, A., Reid, I.: Articulated body motion capture by annealed particle filtering. In: CVPR (2000)
15. Sidenbladh, H., Black, M.J., Fleet, D.J.: Stochastic tracking of 3D human figures using 2D image motion. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1843, pp. 702–718. Springer, Heidelberg (2000)
16. Sminchisescu, C., Triggs, B.: Covariance scaled sampling for monocular 3d body tracking. In: CVPR (2001)
17. Sigal, L., Balan, A., Black, M.J.: HumanEva: synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. Int. J. Comput. Vis. **87**, 4–27 (2010)
18. Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., Davis, J.: Scape: shape completion and animation of people. In: TOG (2005)
19. Poppe, R.: Vision-based human motion analysis: an overview. Compu. Vis. Image Underst. **108**, 4–18 (2007)
20. De Aguiar, E., Stoll, C., Theobalt, C., Ahmed, N., Seidel, H.P., Thrun, S.: Performance capture from sparse multi-view video. ACM Trans. Graph. **27**, 1–9 (2008)
21. Jain, A., Thormählen, T., Seidel, H.P., Theobalt, C.: Moviereshape: tracking and reshaping of humans in videos. In: TOG (2010)
22. Stoll, C., Hasler, N., Gall, J., Seidel, H., Theobalt, C.: Fast articulated motion tracking using a sums of gaussians body model. In: ICCV (2011)
23. Freeman, W.T., Roth, M.: Orientation histograms for hand gesture recognition. In: International Workshop on Automatic Face and Gesture Recognition (1995)
24. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**, 91–110 (2004)
25. Laptev, I.: On space-time interest points. Int. J. Comput. Vis. **64**, 107–123 (2005)
26. Dalal, N., Triggs, B., Schmid, C.: Human detection using oriented histograms of flow and appearance. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 428–441. Springer, Heidelberg (2006)
27. Mori, G., Malik, J.: Estimating human body configurations using shape context matching. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2352, pp. 666–680. Springer, Heidelberg (2002)
28. Agarwal, A., Triggs, B., Rhone-Alpes, I., Montbonnot, F.: Recovering 3D human pose from monocular images. IEEE Trans. Pattern Anal. Mach. Intell. **28**, 44–58 (2006)
29. Grauman, K., Shakhnarovich, G., Darrell, T.: Inferring 3d structure with a statistical image-based shape model. In: ICCV (2003)
30. Shakhnarovich, G., Viola, P., Darrell, T.: Fast pose estimation with parameter-sensitive hashing. In: ICCV (2003)
31. Ramanan, D., Forsyth, D., Zisserman, A.: Strike a pose: Tracking people by finding stylized poses. In: CVPR (2005)
32. Buehler, P., Zisserman, A., Everingham, M.: Learning sign language by watching TV (using weakly aligned subtitles) (2009)
33. Fischler, M.A., Elschlager, R.: The representation and matching of pictorial structures. IEEE Trans. Comput. **22**, 67–92 (1973)

34. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: CVPR (2008)
35. Andriluka, M., Roth, S., Schiele, B.: Pictorial structures revisited: people detection and articulated pose estimation. In: CVPR (2009)
36. Dantone, M., Gall, J., Leistner, C., Gool., L.V.: Human pose estimation using body parts dependent joint regressors. In: CVPR (2013)
37. Johnson, S., Everingham, M.: Learning effective human pose estimation from inaccurate annotation. In: CVPR (2011)
38. Pishchulin, L., Andriluka, M., Gehler, P., Schiele, B.: Poselet conditioned pictorial structures. In: CVPR (2013)
39. Bourdev, L., Malik, J.: Poselets: body part detectors trained using 3d human pose annotations. In: ICCV (2009)
40. Gkioxari, G., Arbelaez, P., Bourdev, L., Malik, J.: Articulated pose estimation using discriminative armlet classifiers. In: CVPR (2013)
41. Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., Moore, R.: Real-time human pose recognition in parts from single depth images. ACM (2013)
42. Zeiler, M., R., F.: Visualizing and understanding convolutional neural networks. In: arXiv preprint arXiv:1311.2901. (2013)
43. Razavian, A.S., Azizpour, H., Sullivan, J., Carlsson, S.: Cnn features off-the-shelf: an astounding baseline for recognition (2014)
44. Yaniv Taigman, Ming Yang, M.R., Wolf, L.: Deepface: closing the gap to human-level performance in face verification. In: CVPR (2014)
45. Deng, L., Abdel-Hamid, O., Yu, D.: A deep convolutional neural network using heterogeneous pooling for trading acoustic invariance with phonetic confusion. In: ICASSP (2013)
46. Sermanet, P., Kavukcuoglu, K., Chintala, S., LeCun, Y.: Pedestrian detection with unsupervised multi-stage feature learning. In: CVPR (2013)
47. Weinzaepfel, P., Revaud, J., Harchaoui, Z., Schmid, C.: Deepflow: large displacement optical flow with deep matching. In: ICCV (2013)
48. Toshev, A., Szegedy, C.: Deeppose: Human pose estimation via deep neural networks. In: CVPR (2014)
49. Jain, A., Tompson, J., Andriluka, M., Taylor, G., Bregler, C.: Learning human pose estimation features with convolutional networks. In: ICLR (2014)
50. Tompson, J., Stein, M., LeCun, Y., Perlin, K.: Real-time continuous pose recovery of human hands using convolutional networks. In: TOG (2014)
51. Johnson, S., Everingham, M.: Clustered pose and nonlinear appearance models for human pose estimation. In: BMVC (2010)
52. Collobert, R., Kavukcuoglu, K., Farabet, C.: Torch7: a matlab-like environment for machine learning. In: BigLearn, NIPS Workshop (2011)
53. Giusti, A., Ciresan, D.C., Masci, J., Gambardella, L.M., Schmidhuber, J.: Fast image scanning with deep max-pooling convolutional neural networks. In: CoRR (2013)
54. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. In: ICLR (2014)
55. Sutskever, I., Martens, J., Dahl, G., Hinton, G.: On the importance of initialization and momentum in deep learning. In: ICML (2013)