Pose Encoding for Robust Skeleton-Based Action Recognition

Girum G. Demisse Konstantinos Papadopoulos Djamila Aouada Björn Ottersten
Interdisciplinary Center for Security, Reliability and Trust
University of Luxembourg, 29, avenue J. F. Kennedy, L-1855 Luxembourg
{girum.demisse, konstantinos.papadopoulos, djamila.aouada, bjorn.ottersten}@uni.lu

Abstract

Some of the main challenges in skeleton-based action recognition systems are redundant and noisy pose transformations. Earlier works in skeleton-based action recognition explored different approaches for filtering linear noise transformations, but neglect to address potential nonlinear transformations. In this paper, we present an unsupervised learning approach for estimating nonlinear noise transformations in pose estimates. Our approach starts by decoupling linear and nonlinear noise transformations. While the linear transformations are modelled explicitly the nonlinear transformations are learned from data. Subsequently, we use an autoencoder with L_2-norm reconstruction error and show that it indeed does capture nonlinear noise transformations, and recover a denoised pose estimate which in turn improves performance significantly. We validate our approach on a publicly available dataset, NW-UCLA.

1. Introduction

Over the last few years, a significant progress has been made in computer vision applications using human pose estimates as an input data. Applications range from action recognition [27, 11, 16] to guidance systems for home rehabilitation [2, 6, 4]. Such approaches simplify the problem by restricting their observation to a stick figure of the subject performing the action, usually referred to as the skeleton. They are purely skeleton-based methods [5] when it is the main source of information or hybrid if skeletons are merged with other features [19]. When using skeletons only, the dynamics of a particular action is estimated from a compressed data, estimated poses. The compression reduces the dimensionality of the problem, in effect its complexity. Regardless, however, estimated poses are generally not invariant to differences in intrinsic and extrinsic camera parameters. Moreover, pose estimation methods tend to exhibit high nonlinearity in certain regions of the problem domain, e.g., a slight difference in actual poses might lead to a large difference in the final pose estimates. Overall, such kind of task-irrelevant transformations corrupt the input data and obscure the underlying dynamics which, in most cases, account for overfitting models.

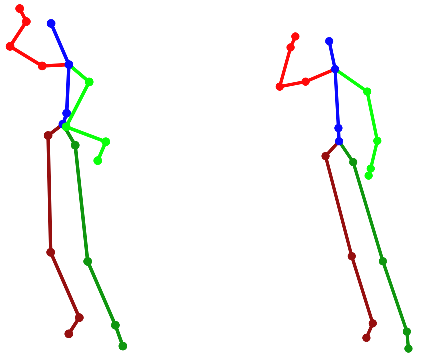

(a) Noisy input (b) Denoised result

Figure 1: **Illustration of the proposed approach**: (a) shows estimated input pose in frontal view, (b) shows an approximate denoised reconstruction of the input in (a).

In this paper, we introduce an unsupervised learning approach to filter coordinate and nonlinear variations from estimated poses. Consequently, we treat observed action sequences as transformed versions of a noise-free underlying dynamics. As such, the goal is to correctly estimate a task-irrelevant transformation, and attempt to recover a standardized and noise-free pose estimate from a noisy and redundant input, see Figure 1. To that end, we propose to use autoencoders for *denoising pose estimates*. The proposed approach starts by assuming an explicit model of pose variation due to scale, location, and rotation. Subsequently, the model is used to normalize rigid transformation of estimated poses. In effect, coordinate variations of an estimated pose are isolated from variations that are due to unknown nonlinear transformations, e.g., nonlinearity of pose estimating algorithms and measurement fluctuations. Next, an autoencoder is used to explicitly filter and correct noisy pose estimates that are caused mainly due to nonlinearity. We evaluate our approach on a publicly available dataset. Results show that denoising pose estimates does indeed have a significant effect in improving performance.

The paper is organized as follows: in Section 2 a brief overview of related works is provided. In Section 3, we describe the general framework of the proposed approach. Section 4 describes experimental setups, the dataset, and results of our approach. The paper concludes with summarizing remarks in Section 5.

2. Related works

Several of the earlier skeleton-based approaches are tailored to address data variation due to known task-irrelevant transformations. In [27, 11, 16, 9, 25, 5], data variation due to scale, location and rotation, are addressed by normalizing the data with respect to a given standard. Meanwhile, variations in an action dynamics are modelled with probabilistic [30, 29] or deterministic [32, 16, 17] temporal models. Concurrently, several approaches improve on accuracy and robustness by further reducing pose estimates to body parts [27, 11, 32, 1] and focusing on a particular body part that is correlated with a given action. Nevertheless, in most of the above approaches, the addressed data variations are modelled explicitly, e.g., data variations due to different camera parameter are generally modelled as linear transformations. However, there is no explicit model to address highly nonlinear variations in pose estimates. In that respect, a general purpose data compression approach would ideally be able to capture such variations.

Some of the earliest works in general purpose data compression are proposed in the context of signal reconstruction. In [8, 10], it was shown that under a mild assumption on a known and underdetermined linear system, selecting a solution that minimizes the L_1 norm is equivalent to selecting the sparsest solution. Thereby, eliminating redundancy. Subsequently, sparsity and other smoothness based data representations are generalized to problems where the transformation function is neither linear nor known [24, 20, 26]. In a similar spirit, several works are proposed in sequential data representation learning. In [23, 15, 3], an encoder that maps a sequence to a fixed size high-dimensional vector is presented with an application in machine translation. In [22] an encoding of a video is presented with an action recognition application from videos. Nevertheless, our approach draws its main motivation from sparsity-based compression methods that assume fixed data size as opposed to a sequence.

3. Model description

Let $x \in \mathbb{R}^n$ be the joint positions of an actual pose in a given world coordinate system, and let $\tilde{x}$ be an estimate of the pose from an image or depth map defined as

$$f(x) = \tilde{x}, \tag{1}$$

where $f(\cdot)$ is an unknown transformation function. Given the above formulation, our main goal is to solve for the original pose x without an explicit knowledge or model of the transformation function $f(\cdot)$. Consequently, in this section, we first describe the underlying framework (autoencoders) of our model and proceed to the description of the proposed approach.

3.1. Autoencoders

An autoencoder is a deep learning based framework that is closely related to Independent Component Analysis (ICA) [7]. Given a set of data points $\tilde{x}_{i=1}^{n}$ an autoencoder solves for what is known as a reconstruction error which, using the L_2 norm, is written as follows

$$\underset{\Theta_g, \Theta_h}{\arg\min} \sum_{i=1}^{n} \|\tilde{x}_i - g(h(\tilde{x}_i))\|_2. \tag{2}$$

The functions g (decoder) and h (encoder) are mostly modelled as feedforward networks, hence are parameterized with connection weights and biases which we denoted altogether with Θ_g, Θ_h. In general, the main goal of (2) is to identify the underlying transformation of the dataset, formalized as $g \circ h$. However, in most cases, (2) does not have a unique solution, e.g., a trivial solution would be an identity that will lead to zero reconstruction error. As a result, apart from identifying a suitable reconstruction error function, it is important to regularize the cost with general and domain-specific priors so that non-trivial transformations can be learned.

In the next subsection, we describe details of an autoencoder-based learning architecture that is conditioned to correct and compress noisy pose estimates.

3.2. Proposed approach

The main two goals of our approach are to achieve robustness to noise transformations and remove redundancy in pose estimates. Subsequently, we address coordinate normalization and its generalization, pose denoising, as follows.

Nonlinear pose variation: Variation in pose estimate due to scale, location and rotation are mostly modelled as linear transformations by considering the vectorized form of a pose estimate as a rigid-object. Consequently, given a pose estimate with n joints, $\tilde{x} = (J_1, \cdots, J_n)$, location is filtered by fixing a given reference point p_c as

$$\tilde{x} = (J_1 - p_c, \cdots, J_n - p_c). \tag{3}$$

In [25, 16], p_c is computed as the mean vector of the hip joints. Similarly, we compute p_c as the mean vector of the two hip joints and the torso joint. Meanwhile, scale is normalized by standardizing the vectorized form of the pose estimate $\tilde{x}$ to unit norm. Finally, rotational variation is approximated by estimating the camera pose with respect to a fixed world coordinate system. To that end, let J_{hl} be the left hip joint position of a centered pose estimate. An orthogonal vector to J_{hl}, in the direction of the torso joint J_t, is then estimated as

$$J_{hl}^{\perp} = J_t - \Big(\frac{(J_{hl})^T}{\|J_{hl}\|_2} J_t\Big) J_{hl}, \tag{4}$$

where $J_{hl}^{\perp}$ denotes a vector that is orthogonal to J_{hl}. The third and final orthogonal vector is then estimated as

$$(J_{hl}, J_{hl}^{\perp})^{\perp} = J_{hl} \otimes J_{hl}^{\perp}, \tag{5}$$

where $\otimes$ denotes cross product. The orthonormal version of the above three vectors, $\mathbf{M} = (J_{hl}, \quad J_{hl}^{\perp}, \quad (J_{hl}, J_{hl}^{\perp})^{\perp})$, constitute the camera position estimate with respect to a fixed world coordinate frame. Finally, a given pose estimate is standardized to a fixed coordinate orientation as follows

$$\tilde{x} = \mathbf{M}^T \times \tilde{x}, \tag{6}$$

where $(\cdot)^T$ denotes matrix transpose. Henceforth, we denote a pose estimate coordinate transformation function as κ, and the standardization of scale, location and coordinate orientation as κ^{-1}. Subsequently, we rewrite (1) as

$$f(x) = \kappa \circ \tilde{f}(x) = \tilde{x}, \tag{7}$$

and use κ^{-1} to filter coordinate transformations. As such, the problem is now to solve for the data source x with an estimate of $\tilde{f}$ by optimizing

$$\arg\min_{\tilde{f},x} \|k^{-1}(\tilde{x}) - \tilde{f}(x)\|_2. \tag{8}$$

Subsequently, we use autoencoders to solve for $\tilde{f}$ by approximating it with the decoder, and the data source x by the latent variable $h(\kappa^{-1}(\tilde{x}))$, written as

$$\arg\min \sum_{i=1}^{n} \|k^{-1}(\tilde{x}) - g \circ h(k^{-1}(\tilde{x}))\|_2. \tag{9}$$

Consequently, the encoder h, defined in (2), models $\tilde{f}^{-1}$ which represents the nonlinear transformation of a noisy pose estimate to noise-free data. While, the decoder g represents a denoised yet uncompressed approximation as shown in Figure 1.

Autoencoders for pose denosing: The proposed learning architecture for solving (9) is composed of an encoder and decoder feedforward networks where the encoder is defined as

$$h = \text{relu}(\mathbf{W}_l^e \tilde{x}_l + b_l^e), \tag{10}$$

and the decoder as

$$g = \tanh(\mathbf{W}_l^d h(\tilde{x}_l) + b_l^d), \tag{11}$$

l is used to identify the layers. We have chosen to use rectified linear units (Relu) to strictly enforce sparsity through hard-nonlinearity, instead of imposing an L_1 norm constraint on the encoder. However, unlike other common nonlinearities Relu is unbounded (does not saturate) opening possibilities for learning biased representations, see [13] for details. Consequently, similar to [21] we reparametrize the connecting weights as

$$\mathbf{W}_l^i = s_l \frac{\mathbf{W}_l^i}{\|\mathbf{W}_l^i\|_2}, \tag{12}$$

where i denotes connections per hidden unit (a row in the weight matrices), s_l is scalar. In this work, however, s_l is estimated per layer not per hidden unit.

In order to further ensure non-trivial transformation learning, we reduce the dimensionality of the latent variable h. In such a case, the network has to learn to compress the input data into a much smaller dimensional latent variable h in such a way that it can reconstruct the original pose from it.

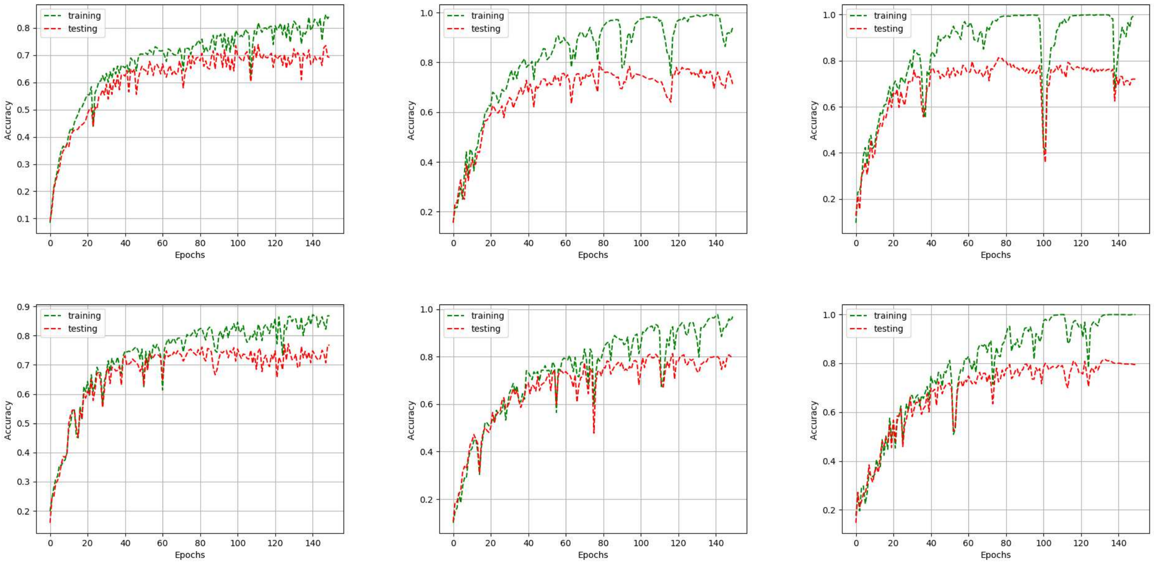
3.3. Robust action recognition

The dynamics of an action recognition system is modelled using LSTM (Long short-term memory) [14]. Together with a nonlinear pose transformation model, described in Section 3.2, LSTM completes the general architecture of the proposed *robust action recognition* system.

The most common architecture in representation learning is to use an unsupervised learning to initialize parameters of a supervised learning [12]. Here, however, we simply denoise poses and treat them as inputs for the supervised learning. Hence, there is neither supervision in learning to denoise the poses nor the learned transformation function is adjusted by class-specific error later on. Subsequently, given a denoised version of the poses, the outputs of the LSTM cells are projected to the class labels using a single layer feedforward network.

4. Experimental results

In this section, we describe the dataset we have used for the experimental analysis, Northwestern-UCLA (NW-UCLA) dataset [28], the experimental setups, and analysis of the results.



(a) $\mathbf{M}_1$ = (30,1)

(b) $\mathbf{M}_2$ = (30,4)

(c) $\mathbf{M}_3$ = (60,4)

Figure 2: **Training vs testing accuracy**: The figure shows training (in green) vs testing (in red) accuracy against optimization iterations. The impact of pose normalization κ^{-1} and pose denoising $\tilde{f}^{-1}$ are shown in two separate rows. The first row is the result of models when the data is filtered using κ^{-1} and the second row when the data is filtered using $\tilde{f}^{-1}$. Each column shows results of their respective model. Note that, the training accuracy trails the test accuracy closely in case of $\tilde{f}^{-1}$ as opposed to κ^{-1} as the model size increases.

Northwestern-UCLA dataset: The dataset contains 10 types of action sequences taken from three different point of views. Ten different subjects performed each action up to 10 times, creating variability in the dynamics of an action. Furthermore, the dataset is collected simultaneously using three cameras installed at different positions and orientations. Hence, the variable between the data collected by two different cameras is principally due to redundant noise transformations. As a result, it is particularly suited to evaluate the proposed approach.

Experimental setups: We follow a similar experimental setup as described in [31]– we use the data from the first two cameras for training, and use the data from the third for testing. However, in order to show the impact of the proposed approach as a preemptive action against overfitting, we evaluate our approach using temporal models with different capacities. To get a baseline performance, we use pose normalization, κ^{-1}.

Subsequently, we use filtered pose estimates to train and test three different LSTM models. Each model is different from another only by the number of hidden units and layers. To that end, denoting the number of hidden layers by L, number of hidden units by H and a model by (H,L), we have used an LSTM model $\mathbf{M}_1$ = (30, 1), and $\mathbf{M}_2$ = (30,4), and finally $\mathbf{M}_3$ = (60, 4). A mini-batch size of 20 is kept for all models with the same learning rate (0.001) and epoch number (140). Each model is trained and tested on a pose estimate filtered by the following approaches

1. κ^{-1}: Here, we simply standardize the poses, without accounting for the nonlinear variations, and achieve a baseline performance.

2. $\tilde{f}^{-1}$: Here, we use an autoencoder of three hidden layers with the number of hidden units corresponding to (40,30,20); the decoder is composed of (20,30,40) hidden layers. Hence, the final pose representation is a 20-dimensional sparse vector.

4.1. Results

The goal of the described experimental setup is to mainly evaluate the impact of the proposed approach in two scenarios: 1) where overfitting is less likely, and 2) where overfitting is more likely. In that regard, the network model $\mathbf{M}_1$ = (30,1) is representative of a model with a much smaller number of parameters thus likely to not overfit a dataset. On the contrary, the models $\mathbf{M}_2$ = (30, 4) and $\mathbf{M}_3$= (60,4) represent models that are more likely to overfit a dataset in comparison to $\mathbf{M}_1$.

Consequently, in using $\mathbf{M}_1$ for modelling the dynamics of an action, the testing accuracy trails the training accu-

Methods	Accuracy (%)
HBRNN-L [11]	78.52
Lie group [25]	74.20
Actionlet ensemble [27]	76.00
Ensemble TS-LSTM [16]	89.22
Enhanced skeleton visualization [18]	86.09
Our appraoch	
Denoised-LSTM $\mathbf{M}_1$ = (30,1)	76.81
Denoised-LSTM $\mathbf{M}_2$ = (30,4)	80.25
Denoised-LSTM $\mathbf{M}_3$ = (60,4)	79.57

Table 1: **Performance comparison**: The table shows results of recent and earlier works on Northwestern-UCLA dataset. Mainly due to the proposed approach, the base LSTM-model performed comparably to most of the specialized models.

Models	Filters accuracy (%)	
	κ^{-1}	$\tilde{f}^{-1}$
$\mathbf{M}_1$	71.84	76.81
$\mathbf{M}_2$	71.36	80.25
$\mathbf{M}_2$	72.94	79.57

Table 2: **Experimental result**: The table summarizes the performance of different models while trained and tested using different input data filters.

racy closely regardless of which data filter is used, κ^{-1} or $\tilde{f}^{-1}$. However, as the model capacity is increased from $\mathbf{M}_1$ to $\mathbf{M}_2$ and to $\mathbf{M}_3$, the testing accuracy starts to diverge from the training accuracy depending on the filter. As such, it characterizes an overfitting model. However, as shown in Figure 2, using the proposed nonlinear filter $\tilde{f}^{-1}$, the difference between testing accuracy and training accuracy is stabilized as the models capacity is increased. This fact is shown much more clearly in Figure 3 and Table 2, through improved accuracy and stable training/testing performance difference. As a result, the experimental results indicate that using the proposed approach $\tilde{f}^{-1}$ on top of pose standardization κ^{-1} does indeed add robustness and improves performance.

Although the proposed approach is not designed to address variability in action dynamics, the performance boost due to the pose encoding resulted in a comparable performance while using low capacity models as compared to works presented in [11, 16], see Table 1. Finally, we show a qualitative result of accurately denoised poses and failure cases in Figure 4.

5. Conclusion

In this paper, we have introduced an approach for filtering nonlinear variations in pose estimates. Our approach began by decoupling pose estimate variation due to coordinate transformation from nonlinear pose variation. Subsequently, the later is modelled using an encoder in encoder-decoder (autoencoder) framework. We have shown that the proposed model does indeed capture redundancy in pose representation and remove noise. Consequently, helps to avoid dataset overfitting when large capacity models are used, thereby improving performance. Nevertheless, exploring alternative architectures can potentially improve robustness and improve performance, e.g., overcomplete autoencoders. Furthermore, the proposed approach can be integrated with any high capacity model, e.g., Ensemble LSTM [16], to improve performance and mitigate a potential overfitting. The integration can be purely unsupervised, as presented here, or semi-supervised, where the learned representation is used to initialize a supervised network's parameters.

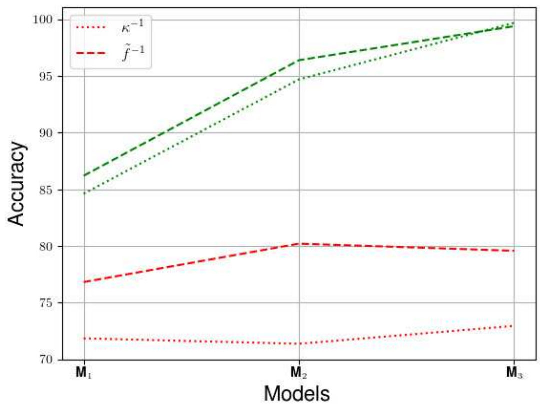


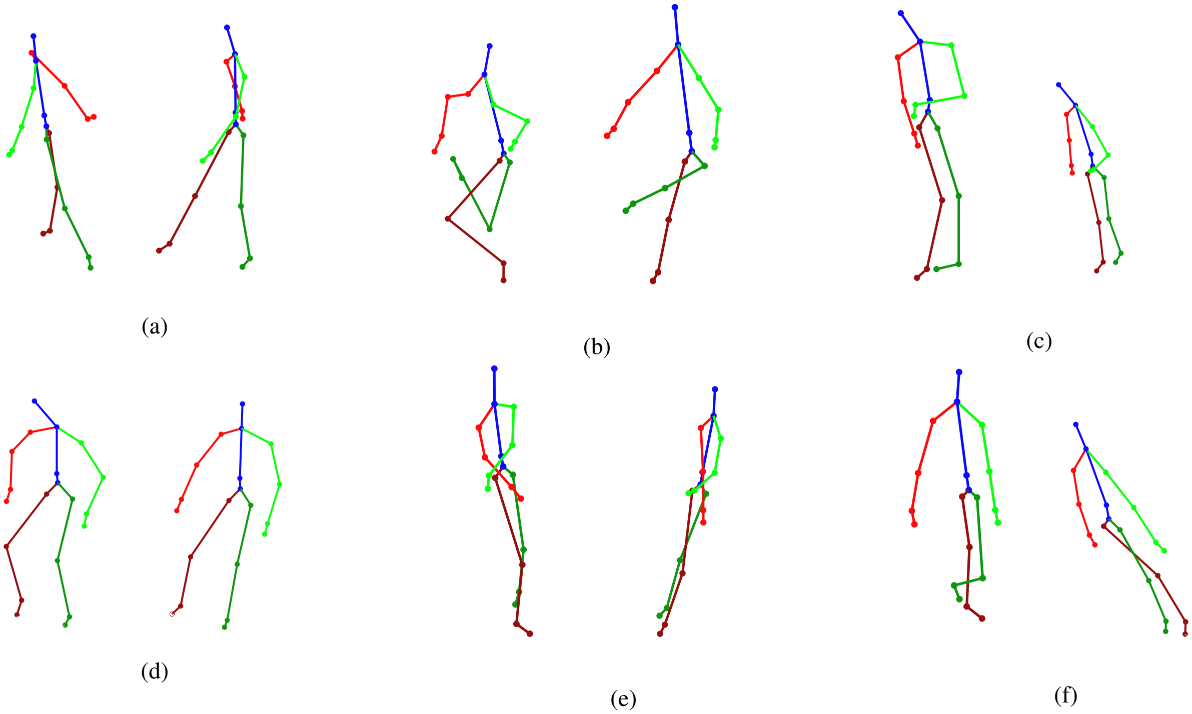
Figure 3: **Performance vs model capacity**: The figure shows the difference between testing accuracy (in red) and training accuracy (in green) for different model sizes. Note that, the difference increases much faster in case of κ^{-1} as compared to $\tilde{f}^{-1}$.

Acknowledgement

We thank Renato Baptista for his technical support and helpful comments. This work was funded by the European Unions Horizon 2020 research and innovation project STARR under grant agreement No.689947, and by the National Research Fund (FNR), Luxembourg, under the project C15/IS/10415355/3D-ACT/Bjorn Ottersten.

References

[1] M. Antunes, D. Aouada, and B. Ottersten. A revisit to human action recognition from depth sequences: Guided svm-sampling for joint selection. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–8. IEEE, 2016. 2


(a)

(b)

(c)

(d)

(e)

(f)

Figure 4: **Pose denoising**: The figures show different examples of pose denoising. In each pair of the examples the pose on the left is *raw input* and the pose on the right is the *denoised* result. Although most of the examples show reasonably well approximated and denoised poses, figures in (e) and (f) are shown as examples of bad approximation.

[2] M. Antunes, R. Baptista, G. G. Demisse, D. Aouada, and B. Ottersten. Visual and human-interpretable feedback for assisting physical activity. In *European Conference on Computer Vision Workshop (ECCVW)*, 2016. 1

[3] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 2

[4] R. Baptista, M. Antunes, D. Aouada, and B. Ottersten. Video-based feedback for assisting physical activity. In *12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP)*, 2017. 1

[5] R. Baptista, M. Antunes, D. Aouada, and B. Ottersten. Anticipating suspicious actions using a small dataset of action templates. In *13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP)*, 2018. 1, 2

[6] R. Baptista, M. Antunes, A. E. R. Shabayek, D. Aouada, and B. Ottersten. Flexible feedback system for posture monitoring and correction. In *IEEE International Conference on Image Information Processing (ICIIP)*, 2017. 1

[7] A. J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7(6):1129–1159, 1995. 2

[8] E. J. Candes and T. Tao. Decoding by linear programming. *IEEE transactions on information theory*, 51(12):4203–4215, 2005. 2

[9] K. Cho and X. Chen. Classifying and visualizing motion capture sequences using deep neural networks. In *Computer Vision Theory and Applications (VISAPP), 2014 International Conference on*, volume 2, pages 122–130. IEEE, 2014. 2

[10] D. L. Donoho. For most large underdetermined systems of linear equations the minimal 1-norm solution is also the sparsest solution. *Communications on pure and applied mathematics*, 59(6):797–829, 2006. 2

[11] Y. Du, W. Wang, and L. Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1110–1118, 2015. 1, 2, 5

[12] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11(Feb):625–660, 2010. 3

[13] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 315–323, 2011. 3

[14] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 3

[15] N. Kalchbrenner and P. Blunsom. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, 2013. 2

[16] I. Lee, D. Kim, S. Kang, and S. Lee. Ensemble deep learning for skeleton-based action recognition using temporal sliding

lstm networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1012–1020. IEEE, 2017. 1, 2, 3, 5

[17] J. Liu, A. Shahroudy, D. Xu, and G. Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In *European Conference on Computer Vision*, pages 816–833. Springer, 2016. 2

[18] M. Liu, H. Liu, and C. Chen. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition*, 68:346–362, 2017. 5

[19] K. Papadopoulos, M. Antunes, D. Aouada, and B. Ottersten. Enhanced trajectory-based action recognition using human pose. In *IEEE International Conference on Image Processing (ICIP)*, 2017. 1

[20] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pages 833–840. Omnipress, 2011. 2

[21] T. Salimans and D. P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems*, pages 901–909, 2016. 3

[22] N. Srivastava, E. Mansimov, and R. Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852, 2015. 2

[23] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014. 2

[24] Y. W. Teh, M. Welling, S. Osindero, and G. E. Hinton. Energy-based models for sparse overcomplete representations. *Journal of Machine Learning Research*, 4(Dec):1235–1260, 2003. 2

[25] R. Vemulapalli, F. Arrate, and R. Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 588–595, 2014. 2, 3, 5

[26] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010. 2

[27] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1290–1297. IEEE, 2012. 1, 2, 5

[28] J. Wang, X. Nie, Y. Xia, Y. Wu, and S.-C. Zhu. Cross-view action modeling, learning and recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2649–2656, 2014. 3

[29] D. Wu and L. Shao. Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 724–731, 2014. 2

[30] L. Xia, C.-C. Chen, and J. Aggarwal. View invariant human action recognition using histograms of 3d joints. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 20–27. IEEE, 2012. 2

[31] X. Yang and Y. Tian. Super normal vector for activity recognition using depth sequences. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 804–811, 2014. 4

[32] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, X. Xie, et al. Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. In *AAAI*, volume 2, page 8, 2016. 2