Motion Denoising using Deep Learning

Tariq Berrada
CentraleSupélec || ENS
Mathematics & Data Science - SDI || MVA
tariq.berrada@student-cs.fr

Abstract

Recent progress in Deep Learning has made it possible to solve many complex tasks. One notable example is 3D human pose estimation, where the goal is to estimate a 3D animation of one or multiple persons from either image or video files. However, many problems persist in state-of-the-art solutions, such as jittering and self-occlusion. The goal of this work is to study different methods for denoising such models by proposing architectures that handle such artefacts. While most mo-cap data is presented as a sequence of axis-angle rotations of joints with respect to their parents in a rig responsible for controlling the 3D mesh, we posit that compressing such data renders regression tasks easier as physically implausible poses are eliminated from the compressed latent space. Next we explore the accuracy of different methods, mainly interpolation, optimization and dictionary learning on the task of animation denoising. Finally, we propose a novel denoiser architecture making use of LSTMs and a sinusoidal decomposition of the latent code's trajectory to learn accurate latent mappings of noisy-to-smooth animations.

1. Introduction

3D pose estimation has become a popular problematic among the Deep Learning community, especially with the rise of multiple annotated datasets [7][11] that facilitate the access to 3D estimation tasks. A recurrent problematic in state-of-the-art approaches is the presence of jittering in the output animations [4][3][6][2], caused by the ineffectiveness of the MSE loss usually deployed in regression tasks in dealing with non-smoothness within temporal data.
To tackle this problem, multiple approaches have been proposed [6], which rely either on processing the output of the neural networks with smoothing filters such as [1] or the use of better pose initializations to reduce the jittering induced by the iterative regression steps towards the predicted pose [6]. However, the problem still persists, although to a lesser extent. In this work we argue that such methods are unsuitable for such a task and that better representation spaces enable learning better parameterizations of motion.

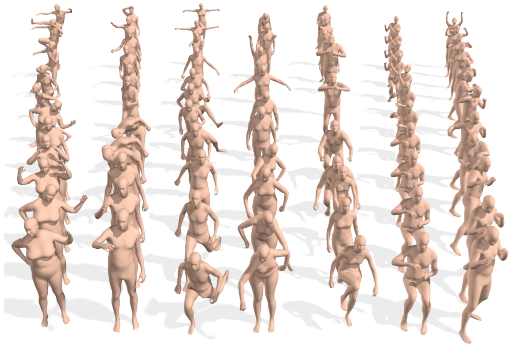

Figure 1. Preview of AMASS dataset.

We begin this work by encoding mocap data using a variational autoencoder to compress the data into a smooth manifold of plausible poses that exclude unrealistic combinations. We then test out denoising methods using different procedures such as interpolation and gradient descent. Finally we present a dictionary learning approach using sinusoidal wave functions to parameterize motion paths with the variational autoencoder's latent space and a denoiser model making use of LSTMs and the pretrained variational autoencoder to estimate smooth motion sequences.

2. Related Work

Recently, most state-of-the-art pose estimation models make use of an adversarial training approach that tries to regress poses in an iterative manner while taking into account the overall temporal context by making use of GRU/LSTM blocks and self-attention [3][6] to learn realistic motion sequences. [3] show that making use of self-attention within the discriminator greatly reduces the temporal inconsistencies and the temporal noise. [6] are able to reduce the overall noise with respect to [3] by encoding a motion prior using a variational autoencoder. However, their model is only meant to provide coarse motion estimates that serves as initializations for the regressor net-

work. Both models make use of AMASS dataset [7], that provides a large spectrum of human mocap using the SMPL standard [5], where the human mesh is parameterized with a set of angle-axis rotations of keypoints with respect to their parents in the skeleton hierarchy.
This work will be mainly based on AMASS dataset [7]. AMASS is a large database of human motion unifying different optical marker-based motion capture datasets by representing them within a common framework and parameterization.
One notable advantage of AMASS is that it's based on SMPL [5], which provides a realistic human body model parameterized by a common shape vector $\beta \in \mathbb{R}^{10}$ and a pose vector $\theta \in \mathbb{R}^{24\times3+3}$.

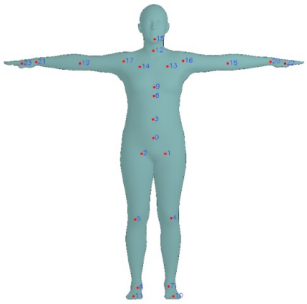

Figure 2. SMPL standard mesh with keypoint pointers.

AMASS regroups multiple sequences of different motions, totaling in 2992.34 minutes of motion performed by 484 actors and representing 14245 different motions, with motion defined by a hierarchical rotation logic following the SMPL[3] standard.

In SMPL [5], a pose is defined within a hierarchical structure of keypoints on the mesh skeleton. The pose is defined by 24 vectors in 3D representing the rotations of a joint with respect to its parent in the hierarchy. One additional vector encodes the global rotation of the root joint with respect to the world frame. The shape of the mesh is defined by the SMPL vector $\beta \in \mathbb{R}^{10}$

Hence an animation of n frames is encoded with β and $\Theta \in \mathbb{R}^{n\times(24\times3+3)}$.
For our study, we will fix the shape vector β to a standard model for all animations, as we will be more focused on the pose parameters θ.

3. Encoding Mocap Data

Using AMASS dataset, we develop a variational autoencoder model to represent a compressed functional space of plausible poses. Both encoder and decoder blocks

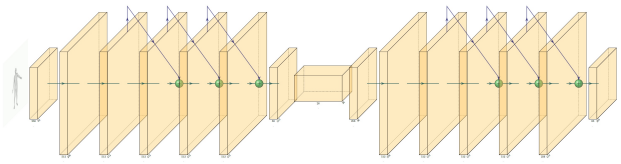

Figure 3. VAE architecture

have 5 fully connected layers with 512 hidden units. The latent space has a dimensions of 28 (while a pose θ has a dimension of $21 \times 3 = 63$). Each block has 3 residual connections and *leaky_relu* activations.

The loss function for training this model is comprised of four terms and can be written as follows :

$$\mathcal{L} = \lambda_1 L_{KL} + \lambda_2 L_\theta + \lambda_3 L_{3D} + \lambda_4 L_\nabla \quad (1)$$

Where L_{KL} is the Kullback Leibler divergence term imposing structure on the learned latent space, L_θ is a MSE loss on the pose prediction parameters (SMPL rotations) $L_\theta = ||\theta_p - \hat{\theta}||_2$, L_{3D} is a MSE on the 3D joint positions, that can be computed by applying the regressor provided by SMPL to the pose parameters $L_{3D} = ||R(\theta_p) - \hat{x}||_2$.
Finally, to enforce the learned latent space to present smooth motions when interpolating, we add an additional constraint on the latent space to present a regular motion amplitude while interpolating.
For each training sample x_i, we sample a random direction on the latent space (i.e a random unit-norm vector $d_i \in \mathbb{R}^{28}$), then penalize deviations centered around x_i along direction d_i at a distance α :

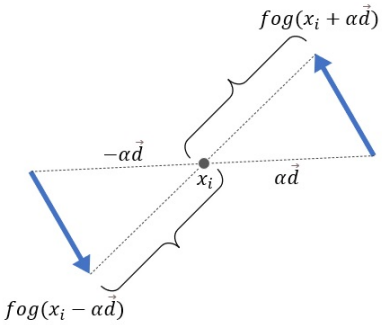

Figure 4. Illustration of the impact of the regularization term on the latent space. *The loss equalizes the distances between brackets*

$$L_\nabla = \left|\left||f \circ g(x) - f \circ g(x - \alpha d)| - |f \circ g(x) - f \circ g(x + \alpha d)|\right|\right|_2 \quad (2)$$

The term α represents the distance with which this constraint is applied. It's calculated by taking the mean over the distance in the latent space between two successive frames over multiple animations from the training set.
Training is done using SGD with Nesterov momentum enabled, the learning rate is set to 10^{-3}, the batch size is 4096 and momentum is set to 0.9. In Figure 5 we can visualize some reconstruction results after training the model.

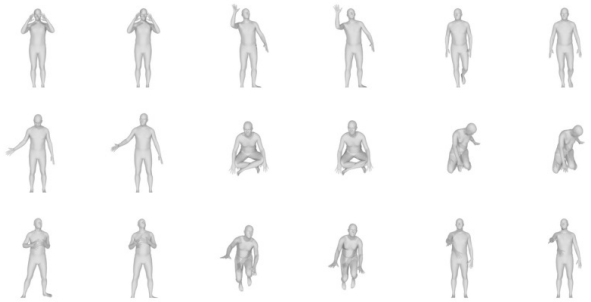

Figure 5. Reconstruction results on the trained variational autoencoder (*poses are shown in set of twos, with left being the ground truth and right the reconstruction).*

3.1. Reconstruction

To evaluate the interpolation properties of the latent space, we conduct experiments on a few animations and their reconstructions.
In figure 7 we can observe two properties :
In row 1 we reconstruct an animation from the test set frame by frame then plot the TSNE embedding of the pose sequences in 2D for both the ground truth and the reconstruction. The lack of noise in the embedding of the predictions indicates smoothness in the predicted animation. In row two we plot the correspondences between the target and predicted poses, the quasi-linear plot obtained validates the accuracy of the frame-by-frame reconstructions.
Finally, we verify the uniformity in the latent space by plotting the TSNE embedding of the latent code corresponding to the frame-by-frame reconstruction of the animation,as reported in Figure 6.

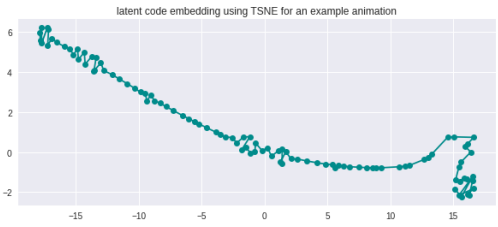

Figure 6. Latent code embedding of example animation.

Qualitatively, we can see that two successive latent codes remain at a similar distance throughout the animation, which is the goal behind the L_∇ term in our training loss and enables interpolation through the latent space to generate plausible animations.

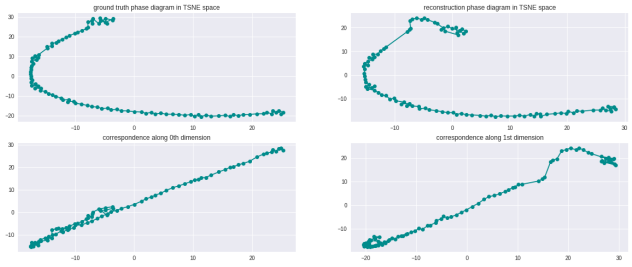

Figure 7. Results of experiments on distance distributions in the VAE latent space.

4. Denoising

In this section, we explore different methods for denoising animations using the embedding space available with our trained variational autoencoder.
For our experiments, we will generate noisy animations by adding i.i.d normally distributed random noise $\epsilon_i \in \mathcal{N}(0, \sigma)$ to the sequences in the test set. The value of $\sigma = 0.1$ is obtained with a qualitative analysis, we try to find an optimal value for corrupting the original signal while still maintaining the overall structure of the motion.

4.1. Interpolation

Figure 8. Illustration of interpolation results on noisy animation

The first method we experiment with is to down-sample the noisy animation and try different interpolation methods to obtain a smoother path for the latent code, which should result in smoother animations.
In the following we experiment with three different interpolation methods after keeping only 1 out of 10 frames. Results are summarised_below : *nearest-neighbor*, *linear* and *cubic*, we then report the MSE error and the norm of the gradients over the animation. Results are summarized in the table below

Method	MSE	Gradient norm
Noisy	0.01	0.02
Nearest	0.209	0.0026
Linear	0.0211	0.0019
Quadratic	0.0215	0.0022
Cubic	0.0216	0.0022

Curiously, interpolating has a negative effect on the MSE, which means that the simplest paths in the latent space don't necessarily correspond to the most realistic animations. On the other hand, we can see that the gradient amplitude is decreased ten-fold with respect to the noisy animation, which further validates the regularization property of our latent space.

4.2. Gradient descent

The second method we experiment with is gradient descent, the idea is to minimize an objective function comprised of both a reconstruction term and a regularization term. As such we define our objective function as follows :

$$\mathcal{L}_{\text{opt}} = ||\theta_p - \hat{\theta}||_2^2 + \lambda||\Delta\theta_p||_1 \tag{3}$$

The first term in the objective function corresponds to the MSE between the noisy animation and the optimization results. The second is a penalty on the laplacian of the rotation parameters, which is meant to smooth out the pose sequences and is computed as :

$$\Delta\theta_i[j] = \theta_i[j+1] - 2\theta_i[j] + \theta_i[j-1] \tag{4}$$

The hyper-parameter λ is a weighting factor for determining the contribution of each term to the optimization results. For the optimization, we use Adam with a learning rate of 0.05 over 10000 iterations and $\lambda = 0.1$. We divide the learning rate by a factor of 5 halfway through training and by a factor of two three quarters into training.

	Noisy	Optimized
MSE (mean)	0.00065	0.0026
MSE (std)	0.099	0.13
Laplacian (mean)	0.06	0.004
Laplacian (std)	0.085	0.025

We recover results similar to those obtained with interpolation (Figure 9), the noise amplitude is greatly reduced but the MSE increases (although to a lesser extent than with interpolation). This leads us to postulate that gradient descent is a method adapted for noise reduction in such cases. However, this method is still very localized (for each frame, we only look at the previous and next neighbor to make a decision), and therefore leads to wander about alternative representation spaces that can model sequences in a longer range.

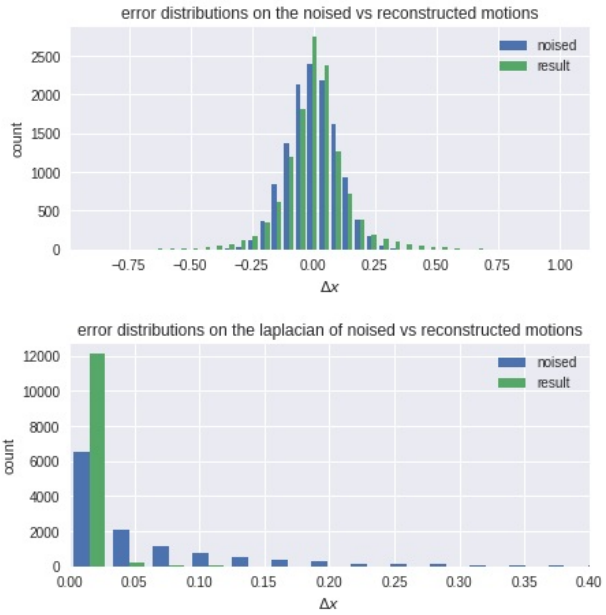

Figure 9. Error distributions for both MSE and the laplacians on the noisy vs optimized pose sequences.

4.3. Dictionary Learning for Denoising

With the results obtained in the previous section, we experiment with Dictionary Learning [9][8] to enforce the smoothness of the animations while keeping the error with respect to the poses low. To do this, we define a sinusoidal basis of functions over our optimization timeframe and search for a sparse decomposition of each of the latent codes in this basis to find an adequate reconstruction of the noisy animation.

To generate each of the 28 components of the latent code, we make use of sinusoidal functions defined as :

$$x[n] = \sum_{i=1}^{N_h} a_i \sin\left(2\pi(if_0)\frac{n}{F_s} + \phi_i\right) + b[n] \tag{5}$$

Where f_0 is the fundamental frequency, F_s is the sampling frequency, a_i and ϕ_i is the amplitude and phase corresponding to sinusoidal i, N_h is the number of harmonics used in the decomposition and n the discretized timestep. With this we can recreate for each latent dimension, a smooth path that will minimize the reconstruction loss. We therefore define the following objective function :

$$L_{\text{sin}} = ||f(z_p) - \hat{\theta}||_2 + \beta\sum_{i=1}^{N_h}||a_i||_1 \tag{6}$$

where f is the encoder of our VAE model, z_p is the latent code where each component is defined using eq. 5, $\hat{\theta}$

is the noisy pose sequence and β is a weight factor. When working with a high number of harmonics, using an l_1 regularization term is necessary to obtain satisfying results. Additionally, Shannon's sampling theorem has to be respected and f_0 should be smaller than $\frac{F_s}{2\times N_h}$, otherwise all the signals in the basis will cover more than one full period during the timeframe $0 : N_h$ and the reconstructed codes will be periodic, which is not wanted in most cases.
For our experiments we set β to 0.02 and initialize the amplitudes and biases from a normal distribution with 0 mean and a standard deviation of 0.03. We use Adam optimizer with a learning rate of 0.001 over 2000 iterations. Results are reported in table below :

Method	Noisy	Dictionary Learning
MSE	0.0008	0.0018
Gradient norm	0.0026	0.0010
Laplacian norm	0.0056	0.0010

The results obtained follow the same trend as those of the previous sections, we note that the laplacian is greatly reduced in comparison to pixelwise gradient descent.

5. Variational Denoiser

While the previous section clearly showcases the efficiency of adopting a custom basis for modelling animations, these method can still be too limited by the optimization constraint, which only learns from one sequence at a time. Consequently, in this section we develop a neural denoiser model. The goal of this model is to learn to extract animations from their noisy counterparts by leveraging the access to smooth animations as ground truth during training and also by making use of the correlations normally present in the behaviors of the curves corresponding to different body parts (the motion of the left leg might mirror that of the right leg for example).

5.1. Architecture

We experimented with different architectures for our model all based on recurrent networks (we used LSTMs) to extract temporal features from the animations. With this, we generate a noised dataset in the same way as in previous sections and make use of the original animation sequences as ground truth. We then train an LSTM-based model to predict the ground truth animation given its noisy counterpart. To have more stable training, the noise amplitude σ is progressively increased during training.
The architecture is illustrated in Figure 10. For a given noisy animation $\{\theta_i\}_{i\in\{1,\ N_h\}}$, the animation is first fed to a 3-layer bidirectional LSTM encoder followed by an ELU activation, afterwards the output hidden states are combined using two successive fully connected layers with ELU activation. The features at the output of the second fully connected block are then used to predict the parameters of the

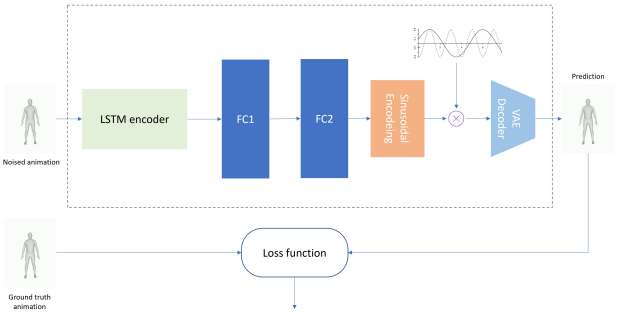

Figure 10. Denoiser model architecture

sinusoidal decomposition defined in the previous section 5. Finally the inferred latent codes are fed to the pretrained VAE decoder (that is fixed during the training of the denoiser) which results in the predicted animation.
For training this model, we make use of the relevant loss functions from the previous sections :

$$\mathcal{L}_{\text{D}} = ||f(z_p) - \hat{\theta}||_2 + \beta_1||R \circ f(z_p) - \hat{x}||_2 + \beta_2 \sum_{i=1}^{N_h} ||a_i||_1 \tag{7}$$

Where f is the VAE decoder, R is the SMPL [5] regressor for transforming pose rotations into their corresponding 3D joint positions and β_i are weight factors for tuning the contribution of each term to the total loss.
For training this model, we use SGD with Nesterov momentum with a learning rate of 0.1 and momentum set at 0.9.

6. Evaluation

A quantitative analysis of the performance of our models is presented below.
We report the gradient amplitude aswell as the MSE on our test set for all of the aforementioned methods :

	MSE	Gradient norm
Noisy	0.01	0.02
Linear	0.0211	0.0019
Quadratic	0.0215	0.0022
Cubic	0.0216	0.0022
Gradient descent	0.0205	0.004
Dictionary learning	0.019	0.0011
Denoiser model	0.0143	0.0013

In this experiment we fixed the same noise amplitude $\sigma = 0.1$ for all models and averaged over 100 animations of length 60 from our test set. The denoiser model performs best in terms of MSE and output gradients that are lower only to the sparse dictionary learning method, this is probably a result of not implementing the l_1 weight decay in the training procedure of the denoiser which imposes sparse

decompositions and therefore smoother paths for the latent codes.
Overall we can see a clear effect of our decomposition method and our intuition was correct about leveraging the temporal data, the VAE latent space and the decomposition model to learn denoised animations since the model clearly outperforms all others (except maybe for dictionary learning) and results in a noticeable improvement over the noisy set, especially for high noise amplitudes.

7. Conclusion

In this article, we presented different possibilities for denoising mocap data. We propose a variational autoencoder architecture to encode the space of plausible human poses while imposing additional regularity to enable the possibility of interpolating through the model's latent space. We then explored different reconstruction methods for animation denoising that leverage the compressed and smooth structure of this latent space.
We have shown the impact of different representation spaces on the reconstruction by making use of dictionary learning approaches to impose a sinusoidal decomposition on the smoothed latent codes. Finally, we leveraged our findings by proposing a novel denoising architecture using LSTMs and the proposed sinusoidal basis decomposition.
With this, we posit that dictionary learning approaches can be very advantageous in pose estimation tasks where the pose parameters have an underlying regularity. Our ideas can therefore be implemented directly into a pose estimation pipeline to predict functional parametrizations instead of frame-by-frame features. However, we leave this endeavour for future work.

Note

- For each section except 6, we used different animations and noise amplitudes for our tests, which is why the results might differ from one test to the other.

- For a qualitative analysis of these methods and for the generated video files, refer to the github repo. The animations are part of the notebook that can be run in colab.

`https://github.com/TariqBerrada/Motion-Denoising`

References

[1] Géry Casiez, Nicolas Roussel, and Daniel Vogel. 1€ Filter: A Simple Speed-based Low-pass Filter for Noisy Input in Interactive Systems. In *CHI'12, the 30th Conference on Human Factors in Computing Systems*, pages 2527–2530, Austin, United States, May 2012. ACM. 1

[2] Imry Kissos, Lior Fritz, Matan Goldman, Omer Meir, Eduard Oks, and Mark Kliger. Beyond weak perspective for monocular 3d human pose estimation. *CoRR*, 2020. 1

[3] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. VIBE: video inference for human body pose and shape estimation. *CoRR*, 2019. 1

[4] Nikos Kolotouros, Georgios Pavlakos, Michael J. Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. *CoRR*, abs/1909.12828, 2019. 1

[5] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015. 2, 5

[6] Zhengyi Luo, S. Alireza Golestaneh, and Kris M. Kitani. Meva: 3d human motion estimation via motion compression and refinement. *CoRR*, 2020. 1

[7] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: archive of motion capture as surface shapes. *CoRR*, 2019. 1, 2

[8] Julien Mairal, Francis Bach, J. Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. volume 382, page 87, 01 2009. 4

[9] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. Supervised dictionary learning, 2008. 4

[10] N. Dinesh Reddy, Minh Vo, and Srinivasa G. Narasimhan. Occlusion-net: 2d/3d occluded keypoint localization using graph networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[11] Timo von Marcard, Roberto Henschel, Michael Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *European Conference on Computer Vision (ECCV)*, sep 2018. 1