Character Pose Design in Latent Space For Animation Edition

Léon Victor, Alexandre Meyer

Character Pose Design in Latent Space For Animation Edition

L. Victor[1], A. Meyer[2]

[1] Univ Lyon, INSA Lyon, LIRIS, CNRS, France
[2] Univ Lyon, Université Claude Bernard Lyon 1, LIRIS, CNRS, France

Abstract

In order to create appealing animation, animators define the key poses of a character by manipulating its underlying skeletons' joints. To look plausible, a human pose must respect many ill-defined constraints and the resulting realism greatly depends on the author's eye for details. Computer animation software propose tools to help in this matter, relying on various algorithms to automatically enforce some of these constraints. The increasing availability of motion capture data has raised interest in data-driven approaches to pose design, with the potential of shifting more of the task of assessing realism from the artist to the computer. In this paper, we propose such a method, relying on neural networks to learn the constraints from the data and to create an alternative representation of the pose space. We then demonstrate one application of this space by performing pose edition through optimization of a pose's latent representation.

1. Introduction

Producing and editing character animation manually is an essential task for animators in computer-generated imagery (CGI) industries such as movies, television and games. Animators traditionally edit animation by posing the character's skeleton on selected key frames between which the computer interpolates. Most animation software such as Blender or Maya provide interactive tools to help editing poses, allowing users to manipulate specific skeleton joints and automatically updating the pose so as not to break any constraint. Internally, Inverse Kinematics (IK) algorithms are used, considering the skeleton as an ensemble of articulated kinematic chains, often having the lengths of the limbs as constraints. Some of the algorithms in this family are able to also tackle constraints on joints orientation or interpenetration of body part, but this usually relies on careful manual specification of each joint's constraints. In practice, an animator still needs an important knowledge of the human body only acquired through hours of practice.

On the other hand, progress in motion capture technologies has made animation data more available than ever, offering a large source of examples viable poses. Neural networks have recently proven to be powerful tools in modelling such complex data; and leveraging this source of information to help in pose design by learning the many constraints of the human body represents an interesting field of research.

In this paper, we train a set of neural networks on a large pose dataset, creating an alternate (latent) representation of the pose space. Exploring this compact space allows us to optimize using straightforward methods without having to explicitly specify skeleton constraints.

2. Related Work

In most scenarios animators edit keyframes by moving individually the skeleton's joints, as if posing a puppet in space. In most animation software the main constraint considered is the constant distance between joints, which is guaranteed by full-body IK solvers.

Inverse Kinematics is a common process in robotics, engineering and computer graphics used to determine the joint parameters of a kinematic chain so as to have its end effector at a desired position. Many IK solutions have been studied over the years [ALCS18]. IK equations are usually solved through approximated linearizations or heuristics, but can also be tackled by data-driven approaches. Numerical methods require a set of iterations to achieve a satisfactory solution formulated by a cost function to be minimized. The numerical family can generally be divided into three sub-categories: Jacobian [SK16], Newtonians [CGBR96] and Heuristics. Most software implement heuristic methods such as Cyclic Coordinate Descent (CCD) [SLGS01] or Forward-Backward Reaching IK (FABRIK) [AL11] due to their simplicity and extensibility. The main drawback of these solvers is that they manipulate kinematic chains without taking into account many morphological aspects that make a pose more or less comfortable.

Although data-driven pose edition is promising, it has not been explored much in the literature. [WTR11] *et al.*propose a method for natural character posing from a large motion database. It employs adaptive KD-clustering to select a representative frame from a large motion database and employs sparse approximations to accelerate training and posing. Huang *et al.*in [HWF*17] present a method based on the formulation of multi-variate Gaussian distribution models (MGDMs), which learn the joint constraints of a

kinematic skeleton from motion capture data. Some work has also been dedicated to more direct interfaces, translating doodles and sketches to motion. Garcia *et al.*[GRC19] propose a method transforming doodle of trajectories (position and orientation over time) into sequences of actions and then into detailed character animations using a dataset of parameterized motion clips automatically fitted to the trajectory.

Neural-networks based generative models of motion have also received a lot of attention due to their low memory usage, scalability in terms of data, and time efficiency at runtime. Deep learning has been effectively applied to generate realistic motion in a number of difficult cases including navigation [HKS17] and interactions with the environment [SZKS19].

Latent variable models are a family of machine learning models whose purpose is to simplify complex data by learning simpler representations. Among latent variable models, auto-encoders have been used for motion by Holden *et al.*[HSK], projecting motion onto the latent space and back to fix issues such as noise in a motion capture clip. Generative Adversarial Networks (GANs) [GPAM*14] attempt to solve the purely generative aspect by training a generator network to translate random samples from a fixed space to real data points. The quality of the generated point is provided by a critic network, and training both networks simultaneously gradually improves the realism of the outputs. GANs have also been used in motion synthesis [LA18].

Data-driven IK and pose editing can relieve animators from time-consuming, back-and-forth pose adjustments by providing a smart space of edition. Recently, neural-network-based approaches have demonstrated major advances in human motion modelling and demonstrated their ability to construct interesting latent spaces. In this paper, we propose a new latent space of human poses. The space is built using multiple neural networks, is able to scale to large amounts of data and is efficient at run-time. We demonstrate that it is well adapted to many tasks such as editing and designing poses for animation production.

3. Proposed method

Pose data is usually defined by each joint's position or rotation relative to its parent in the skeleton hierarchy. However only a small subset of this representation corresponds to valid poses and wrong parameterizations result in unrealistic poses. We propose to build a latent space only representing valid poses using a scaffolding of GANs and autoencoders trained on a large dataset. We then show how simple optimizations on a pose's latent representation can help in satisfying user-defined constraints such as reaching an end effector's target.

We first present our model's architecture ensuring the latent space respects two important properties. First the bijectivity of the mapping to and from the latent space, as we want to be able to retrieve an encoded pose with a high fidelity. Second, to ensure a smooth optimization process the latent space also needs to be convex and fully defined.

We then illustrate the usefulness of such a space with a full-body inverse kinematics solver which satisfies user-defined constraints. By optimizing in this space we ensure that no unrealistic poses are considered, and so remove the possibility of breaking the pose, as illustrated in Fig. 1.

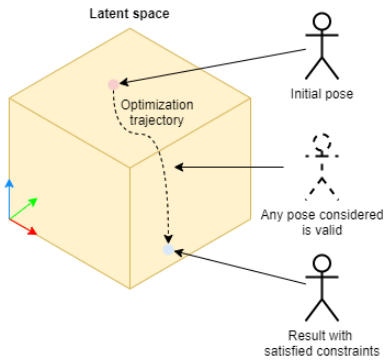

Figure 1: *Optimizing in latent space garanties that any pose considered will be valid*

3.1. Materials

We train the models using a dataset of human poses, obtained by processing multiple available motion-capture datasets from the literature: Emilya [FP14] ; CMU [CMU] ; the MPI Emotional Body Expressions Database for Narrative Scenarios [VdlRBM14] . Each animation clip is retargeted to a standard skeleton following the scheme proposed by [HSKJ15]. The global translation is removed, and each joint's position is calculated relative to the pelvis (root) joint which is fixed in place. The unified skeleton is composed of 21 joints; using the joints' positions in space, a posture is described by $3 \times 21 = 63$ float values concatenated in a single vector. The dataset is then formed by the individual poses in each clip among which we sample randomly during training. We normalize each pose by subtracting the mean and dividing by the standard deviation of each feature.

3.2. Pose latent space

This section describes the proposed neural-network-based construction of the latent space. The architecture is based on the nesting of an autoencoder and a GAN, whose weights are optimized while minimizing dedicated loss functions.

3.2.1. Architecture

Neither of the two major approach to data synthesis with neural networks, auto-encoders and GANs, is suitable to fit the requirements highlighted in section 3. Auto-encoder's latent spaces are unconstrained while GANs do not provide a way of encoding a real data point. Thus, we use the hybrid approach proposed by Lazarou [Laz20] to create a latent space of images in which they illustrate good interpolation properties. We adapt their architecture to human poses to take advantage from both autoencoders and GANs. The method uses four separate networks, illustrated in Fig. 2: an encoder E, a generator (also called decoder) G, a pose discriminator

D_{pose} and a latent vector discriminator D_{lat}. The encoder and decoder are trained to map from pose to latent space and back, and use feedback from the discriminators to shape the desired latent space.

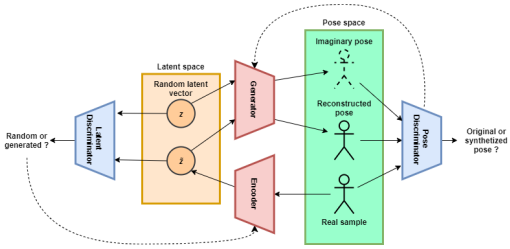

Figure 2: *High level overview of the different models and connections between them.*

The skeleton is a hierarchical structure made of several kinematic chains, and each joint's parameterization has an impact on other joints down the chain. In order to explicitly model this spacial structure, our networks use Structured Prediction Layers (SPL) [AKH19] in place of fully-connected layers. SPL splits dense connections in multiple smaller layers, connected themselves following the kinematic chains of the skeleton.

The encoder network is composed of a single SPL layer of 252 neurons and ReLU activations, followed by a fully connected layer with 32 neurons and a tanh activation. The decoder is the opposite with a fully-connected layer with 252 neurons, ReLU activations, and a SPL layer with 63 outputs units. The pose discriminator is the same as the encoder except for the last activation which is a Sigmoid function. The latent discriminator is a multi layers perceptron with 4 fully connected layers of 256 neurons with Leaky ReLU activations. The last activation is a Sigmoid function.

3.2.2. Loss functions

The encoder and decoder networks are trained to learn a bijective function mapping respectively a latent point to a pose and vice versa; i.e $G(z) = x$ and $E(x) = z$; where x is a pose vector and z is a randomly sampled vector from the latent distribution. The pose discriminator D_{pose} outputs a single scalar representing the probability for a given pose to be "real" over being generated. It is trained to minimize the loss function in Eq. 1.

$$L_{D_{pose}} = log(1 - D_{pose}(x)) + log(D_{pose}(G(z))) \tag{1}$$

In the same manner D_{lat} discriminates between random latent samples and encoded poses. It is trained to minimize Eq. 2

$$L_{D_{lat}} = log(1 - D_{lat}(E(x))) + log(D_{lat}(z)) \tag{2}$$

The convexity of the latent space is dictated by the interaction between the discriminators and the autoencoder. As G tries to fool D_{pose} it is encouraged to turn any point from the latent space in a valid pose. At the same time the latent discriminator encourages the encoder to encode poses into latent vectors indistinguishable from random ones, and so to use the latent space to its full extent. We therefore build a convex latent space in which the real pose are evenly spread. E and G are thus trained conjointly both to reconstruct a pose with fidelity and to fool the discriminators by minimizing Eq. 3.

$$L_{AE} = L_{recon} + L_{gan} \tag{3}$$

$$L_{recon} = \|(G(E(x)) - x)^2\| + \|(G(E(z)) - z)^2\| \tag{4}$$

$$\begin{aligned} L_{gan} =& log(D_{pose}(x)) + log(1 - D_{pose}(G(z))) \\ &+ log(D_{lat}(E(x))) + log(1 - D_{lat}(z)) \end{aligned} \tag{5}$$

3.2.3. Training

The networks are trained for 200 epochs with a batch size of 256 poses. We use the Adam optimizer [KB17] with a learning rate of 0.001 for the pose discriminator and 0.0002 for the encoder, decoder and latent discriminator. In order to regularize the performance of each networks during the training process, the encoder-decoder and latent discriminator are trained with respectively two and three times as much samples as the pose discriminator.

3.3. Pose edition in the latent space

The latent space built in the previous sections can then be used as an interface for interactive character posing. As the generator is trained to fool the discriminator, its output should be guaranteed to be a realistic pose: the skeleton constraints are implicitly learnt and it is no longer necessary to manually parameterize each joint in the skeleton.

This section describes the optimization process used in interactive time to help in pose design. The optimization is used as a full-body inverse kinematic solver in place of classical IK ones. The initial configuration includes a starting pose and the desired positions of any number of targets; no configuration of the skeleton is necessary and any joint can be associated with a target. The solver only uses the trained encoder and decoder models: the initial pose is encoded in the latent space, and the resulting vector is optimized until the decoded pose's end effectors are close enough to their targets.

For simplicity's sake the optimization processed is performed via gradient descent to minimize Eq. 6 where $t_{0..n}$ are the targets positions, $j_{i..n}$ the associated joint's position and $distance(p_1, p_2)$ the euclidean distance function.

$$L = \sum_{i=0}^{n} distance(t_i, j_i) \tag{6}$$

4. Results

We illustrate our full-body editor by comparing with a staple solution from the literature: Forward Backward Reaching Inverse Kinematics (FABRIK). We implement an adapted solution for our base skeleton, without manually specifying joints constraints. Figure 3 shows a comparisons of the poses obtained by latent optimization and FABRIK with the same targets set.

FABRIK working on kinematics chains with no prior on the human skeleton, it may end up with unrealistic poses, whereas our

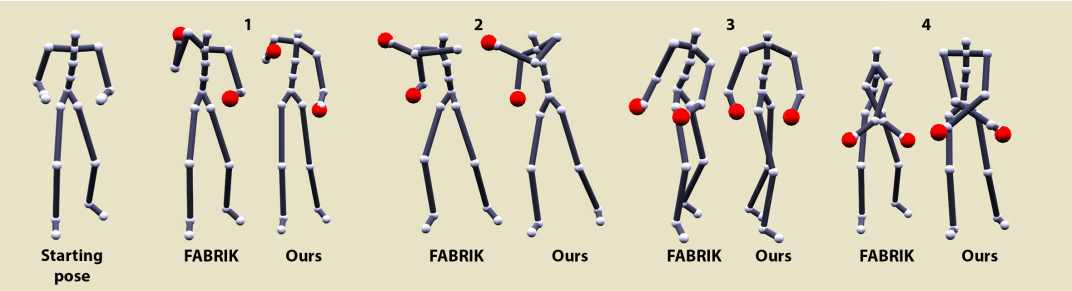

Figure 3: *Examples of pose edition using latent optimization and FABRIK. Targets are shown in red.*

optimization process exploring the latent space results in poses satisfying the constraints without breaking the implicit skeleton rules: the distance between limbs is constant, self-occlusion is avoided and the poses appear natural. In (1) and (2) the skeleton leans on one side in order to reach a target above its shoulder, giving way to its arm. The legs also move slightly so as to appear in balance. In (3), our method makes the skeleton twists it upper body to face the two targets on its side. (4) illustrates the limits of FABRIK without specifying many constraints: when trying to reach targets on opposite sides, the shoulders move forward in an unnatural way. With our methods, the torso structure is implicitly learned and the algorithm finds a more suitable solution.

5. Conclusion and perspectives

We propose to build a latent space of poses by training two encoding and decoding neural networks guided by two discriminators. We enforce its convexity and use this property to our advantage to optimize poses to respect some constraints. We illustrate the results of this approach by using it in a full-body IK solver, optimizing a latent pose representation to avoid unrealistic poses. Further work will focus on adapting our method to a variety of morphologies and skeletons.

References

[AKH19] AKSAN E., KAUFMANN M., HILLIGES O.: Structured Prediction Helps 3D Human Motion Modelling. 10. 3

[AL11] ARISTIDOU A., LASENBY J.: FABRIK: A fast, iterative solver for the Inverse Kinematics problem. *Graphical Models 73*, 5 (Sept. 2011), 243–260. 1

[ALCS18] ARISTIDOU A., LASENBY J., CHRYSANTHOU Y., SHAMIR A.: Inverse Kinematics Techniques in Computer Graphics: A Survey. *Computer Graphics Forum 37* (2018). 1

[CGBR96] COHEN M., GUENTER B., BODENHEIMER B., ROSE C.: Efficient generation of motion transitions using spacetime constraints. In *SIGGRAPH 96* (1996), Association for Computing Machinery. 1

[CMU] Cmu graphics lab motion capture database. *http://mocap.cs.cmu.edu/*. 2

[FP14] FOURATI N., PELACHAUD C.: Emilya: Emotional body expression in daily actions database. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)* (2014), European Languages Resources Association (ELRA). 2

[GPAM*14] GOODFELLOW I. J., POUGET-ABADIE J., MIRZA M., XU B., WARDE-FARLEY D., OZAIR S., COURVILLE A., BENGIO Y.: Generative Adversarial Networks. *arXiv:1406.2661 [cs, stat]* (June 2014). arXiv: 1406.2661. 2

[GRC19] GARCIA M., RONFARD R., CANI M.-P.: Spatial Motion Doodles: Sketching Animation in VR Using Hand Gestures and Laban Motion Analysis. In *MIG 2019 - ACM SIGGRAPH Conference on Motion, Interaction and Games* (2019). 2

[HKS17] HOLDEN D., KOMURA T., SAITO J.: Phase-functioned neural networks for character control. *ACM Trans. Graph. 36*, 4 (2017). 2

[HSK] HOLDEN D., SAITO J., KOMURA T.: A deep learning framework for character motion synthesis and editing. 1–11. 2

[HSKJ15] HOLDEN D., SAITO J., KOMURA T., JOYCE T.: Learning motion manifolds with convolutional autoencoders. In *SIGGRAPH ASIA 2015 Technical Briefs on - SA '15* (2015), ACM Press. 2

[HWF*17] HUANG J., WANG Q., FRATARCANGELI M., YAN K., PELACHAUD C.: Multi-Variate Gaussian-Based Inverse Kinematics. *Computer Graphics Forum* (2017). 1

[KB17] KINGMA D. P., BA J.: Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]* (Jan. 2017). arXiv: 1412.6980. 3

[LA18] LIN X., AMER M. R.: Human Motion Modeling using DVGANs. *arXiv:1804.10652 [cs]* (Apr. 2018). arXiv: 1804.10652. 2

[Laz20] LAZAROU C.: Autoencoding Generative Adversarial Networks. *arXiv:2004.05472 [cs, stat]* (Apr. 2020). arXiv: 2004.05472. 2

[SK16] SICILIANO B., KHATIB O.: *Springer Handbook of Robotics*, 2nd ed. Springer Publishing Company, Incorporated, 2016. 1

[SLGS01] SHIN H. J., LEE J., GLEICHER M., SHIN S. Y.: Computer puppetry: An importance-based approach. *ACM Transcations on Graphics 20*, 2 (apr 2001). 1

[SZKS19] STARKE S., ZHANG H., KOMURA T., SAITO J.: Neural State Machine for Character-Scene Interactions. 14. 2

[VdlRBM14] VOLKOVA E., DE LA ROSA S., BÜLTHOFF H. H., MOHLER B.: The MPI Emotional Body Expressions Database for Narrative Scenarios. *PLoS ONE 9*, 12 (Dec. 2014), e113647. 2

[WTR11] WU X., TOURNIER M., REVERET L.: Natural Character Posing from a Large Motion Database. *IEEE Computer Graphics and Applications 31*, 3 (May 2011), 69–77. 1