Motion Priors for Pose Estimation and Animation Workflows

Luke Smith

Master Thesis
May 2023

Prof. Dr. Robert W. Sumner

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

cgl
computer graphics laboratory

Abstract

Motion capture, the act of automatically recording the motion of a person or object, has solidified it's position as a key technology in modern animation pipelines due to its numerous benefits. It provides real time, high quality, physically plausible results, however the upfront costs for tailored software and hardware, cumbersome motion capture suits, and other issues leads to a high barrier not only for individual artists but also small teams. At Disney Research|Studios, this barrier is being tackled with a system that aims to capture motion directly from RGB video. The system performs pose estimation on the individual frames, and runs the result through an optimizer to clean up the motion. While promising, the system occasionally struggles with certain artifacts and with some occluded motions. The goal of this thesis is therefore to consider how we might learn a general human motion model that could improve upon these shortcomings. To begin with we investigate a state of the art approach based on Variational Auto Encoders, and attempt to improve upon its issues to make it useable for our purposes. We then consider a relatively new class of models, that of diffusion models, which is proving to be a powerful tool in many domains. We consider the application of such models in the context of motion modelling, and implement and evaluate a baseline model with the view of coming to a go/no-go decision on the feasibility of such models for our purposes.

Zusammenfassung

Motion Capture, die automatische Aufzeichnung der Bewegung einer Person oder eines Objekts, hat sich aufgrund ihrer zahlreichen Vorteile als Schlüsseltechnologie in modernen Animationspipelines etabliert. Sie liefert qualitativ hochwertige, physikalisch plausible Ergebnisse in Echtzeit, doch die Vorabkosten für maßgeschneiderte Software und Hardware, umständliche Motion-Capture-Anzüge und andere Probleme führen zu einer hohen Hürde nicht nur für einzelne Künstler, sondern auch für kleine Teams. Bei Disney Research|Studios wird diese Hürde mit einem System angegangen, das darauf abzielt, Bewegungen direkt aus RGB-Videos zu erfassen. Das System führt eine Posenschätzung für die einzelnen Frames durch und lässt das Ergebnis durch einen Optimierer laufen, um die Bewegung zu bereinigen. Das System ist zwar vielversprechend, hat aber gelegentlich Probleme mit bestimmten Artefakten und mit einigen verdeckten Bewegungen. Ziel dieser Arbeit ist es daher zu untersuchen, wie ein allgemeines menschliches Bewegungsmodell erlernt werden kann, das diese Unzulänglichkeiten beseitigt. Zunächst untersuchen wir einen aktuellen Ansatz, der auf Variational Auto Encoders basiert, und versuchen, seine Probleme zu verbessern, um ihn für unsere Zwecke nutzbar zu machen. Anschließend betrachten wir eine relativ neue Klasse von Modellen, die Diffusionsmodelle, die sich in vielen Bereichen als leistungsfähiges Werkzeug erweisen. Wir betrachten die Anwendung solcher Modelle im Zusammenhang mit der Bewegungsmodellierung und implementieren und bewerten ein Basismodell, um eine Entscheidung über die Durchführbarkeit solcher Modelle für unsere Zwecke treffen zu können.

Contents

List of Figures

HuMoR

The authors of HuMoR [RBH+21] present a novel approach for learning and using a plausible motion prior. They train a conditional VAE that learns a distribution over latent transitions from a given state, and can decode a sample from this distribution to a change of state. They notably use this model as a prior in a 'test time optimisation' (TestOps), which generates plausible sequence motions optimising for an initial state and a sequence of transitions starting from frame by frame estimates. The TestOps can operate on many modalities, 2D/3D joints, point clouds, etc., as the optimisation loss contains a Data Term ϵ_{data} that can be tailored to the modality as the HuMoR state is information rich, containing 3D joints (hence can fit to 2D joints through projection or directly to 3D) and can parametrise the SMPL model (hence the SMPL mesh can be correlated to point clouds).

The performance of HuMoR as described in the paper [RBH+21], alongside it's use in a problem that directly matches our own (TestOps operating on RGB video) lead us to evaluate and investigate the model Section 1.2, and subsequently try to extend and improve 1.3 upon it once the limitations made themselves known.

1.1. Relevant Model Details

A brief overview of the salient features of the HuMoR model and TestOps optimisation procedure is provided here, some details are ommited for clarity and brevity.

1.1.1. Architecture

The HuMoR architecture is that of a C-VAE [SLY15], as can be seen in Figure 1.1. The goal of this architecture is to learn a mapping from a given pose ($\mathbf{x}_{t-1}$) to a distribution over latent

transitions to the next pose (the **prior**), and a mapping from a latent transition sampled from this distribution to a change in pose that can be used to obtain the next state (**decoder**). This is achieved during training through additional **encoder** network that has the full information of the next state ($\mathbf{x}_t$) also available to it, such that it can find the ideal latent transition, and this is used to guide the training of the **prior** network.

The networks operate on states, $\mathbf{x}_t$, consisting of a root translation, root orientation, 3D joint positions, and all their respective velocities, and additionally joint angles. This rich and redundant state representation is a superset of the SMPL models' state [BKL+16] (given some shape parameters β).

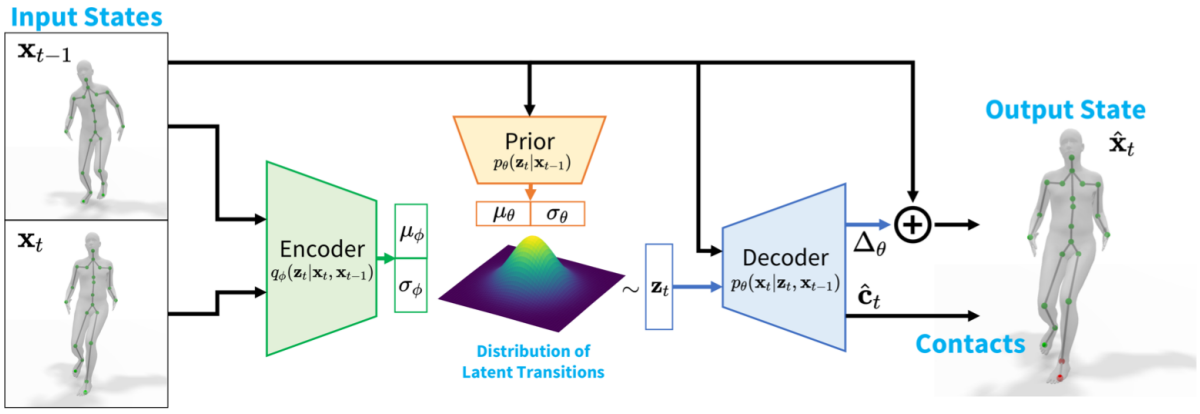


Figure 1.1.: *HuMoR C-VAE Architecture [RBH+21]*

1.1.2. TestOps

TestOps is the term given by the authors of HuMoR to their method of using the HuMoR prior in an optimisation procedure to obtain a smooth motion estimation from, among other things, an RGB video (our focus).

The concept of an autoregressive 'rollout' is important here. As we can see in Figure 1.1, the model output can be used to obtain the next state. This new state can then be autoregressively fed back into the **decoder**, alongside either a given latent transition or one obtain through the **prior** to obtain the next state. The procedure of, starting from a given state $\mathbf{x}_0$, and with a given sequence of latent transitions $\mathbf{z}_{1:T}$, obtaining a sequence of states $\mathbf{x}_{0:T}$ through the autoregressive application of the HuMoR model is called a 'rollout'.

The TestOps procedure is as follows for RGB video. First, frame by frame pose estimation is performed using OpenPose [CHS+19] to obtain a 2d pose estimates. Next, a three stage optimisation procedure is proposed. The first two stages are used to obtain a data that can be used to get a rough estimate of the sequence of states, and the third stage is used to refine this estimate using the HuMoR model.

Stage 1

A global root translation and orientation (in 3d space) is optimised using a loss comparing their projection to the 2d pose estimates.

Stage 2

Next, the full pose parameters (joint positions and angles) are estimated (without the root) through the use of a projection loss to the 2d pose estimates, and of a number of regulariser losses, including smoothness and foot contact losses and an additional pose prior loss ensuring plausible poses using the VPoser [PCG+19] model.

Stage 3 (Initialisation)

Using the root translation and orientation from Stage 1, and the joint positions from stage 2, the velocities of each of these elements can be calculated through finite differences, and all the variables, alongside the joint angles, can be considered the initial estimate $\mathbf{x}_{0:T}$ for the sequence of states. The initial estimates $\mathbf{x}_{0:T}$ are then fed through the **encoder** network to obtain initial estimates of the state transitions $\mathbf{z}_{1:T}$.

Stage 3 (Optimisation)

The final optimisation then operates on a starting state $\mathbf{x}_0$, and a sequence of latent transitions $\mathbf{z}_{1:T}$. For each optimisation step, a rollout (as described above) is performed to obtain a sequence of states $\hat{\mathbf{x}}_{1:T}$, and various losses are applied to these states. The regularisers as described in Section 1.1.2 are applied, a reprojection loss to the 2d pose estimates is used, and finally the HuMoR prior is used to evaluate the likelihood of each transition given the previous state. These losses are summed and the gradients are propagated back to the optimisation variables, $\mathbf{x}_0$ and $\mathbf{z}_{1:T}$. This procedure is repeated for a number of steps, and a sequence of states $\hat{\mathbf{x}}_{0:T}$ is returned after a final rollout.

The computation graph entailed by this Stage 3 procedure is shown in Figure 1.7.

1.2. HuMoR Investigation

1.2.1. Method

Our investigation began with a largely qualitative evaluation of the HuMoR model which had two main aims. First was to stress test the system, to see where it failed, where it succeded, and if the mentioned benefits in the HuMoR paper [RBH+21] were as described. Second was to evaluate the model with the defects of the current Disney Research|Studios system in mind to

see if it could complement it's functionality, notably if it could improve upon occluded motion, joints flipping and confident but false predictions.

To acheive this goal, a selection of videos were taken in the Disney Research|Studios lab containing a variety of motion; fast, slow, and abnormal, and a including number of occluded scenes. The HuMoR system was ran on these videos and the results were investigated. The TestOps is performed in 3 stages, where only the 3rd makes use of the HuMoR motion model, we could therefore compare the stage 2 results to the stage 3 results to see where the model was providing an improvement over a more classical optimisation that includes smoothness and pose prior losses.

1.2.2. Advantages of HuMoR

We see a number of situations in which the model shows a clear improvement over the stage where the HuMoR model is not used.

In an occluded situation, as shown in Figure 1.2, where the 2d pose predictions don't have any information on the legs, the model manages to produce a realistic sitting motion.

(a) OpenPose **(b)** Stage 2 **(c)** Stage 3

Figure 1.2.: *HuMoR achieving occluded sitting*

We also note in many of the less difficult videos, that HuMoR produces clean motion and deals with many abnormal movements without obvious issues. It therefore doesn't seem to regress the easy situations, but can improve the more difficult situations, notably occlusions.

1.2.3. Drawbacks of HuMoR

We noted that while HuMoR manages to sit when occluded, it often fails to walk. This seems to be due to the fact that OpenPose often predicts both legs on the frames just before the occlusions where only one leg is actually un-occluded, as can be seen in Figure 1.3. This results in a sequence of poses where the frames before an occlusion indicate that the person is no longer walking, hence making it significantly more difficult for HuMoR to begin walking again during the occlusion. This issue is thus probably largely due to HuMoRs' TestOps' dependence on OpenPose and thus it's inheritence of OpenPoses' failure points.

We also found that the choice of axis angle representation for various rotations led to the emergence of common known issue that arising from the discontinuities present in the representation [ZBL+18]. The shortest path between certain angles in axis angle space can lead to a 360 degree rotation, as seen in Figure 1.4, among other issues.

(a) OpenPose **(b)** Stage 2 **(c)** Stage 3

Figure 1.3.: *Occluded walking failure point, 2 legs predicted instead of 1*

(a) **(b)** **(c)**

Figure 1.4.: *Axis angle issue: dubious rotations*

We found that the TestOps system would occasionally get itself into a tangle, as seen in Figure 1.5, we assume due to the autogregressive nature resulting in a potentially catastrophic cumulation of errors, and found this happended most notably for a sequence containing someone rolling on the floor. It is interesting to note that these sorts of motions were not often, if ever, present in the training data.

(a) OpenPose **(b)** Stage 2 **(c)** Stage 3

Figure 1.5.: *HuMoR in a tangle*

Next, we saw overly smoothed motion for certain clips, most notably for particularly stylised motion such as dancing. We also found that if there is a single frame without Open Pose predictions then as of that frame the TestOps fails. Finally, and most importantly, we found that the TestOps was extremely slow. It took around 20mins per 2s clip, and that we could batch at most 4 2s clips together, so 20mins per 8s on a Nvidia GeoForce 1080ti with 11Gbs memory.

We concluded that there were a number of potential benifits, notably in occluded situations, and that many of the issues, including the axis angle representation issue, the dependence on humor, and smooth motion, have a good chance of being fixed with different design choices. The glaring issue however was the unresonable amount of time the system took, rendering it unusable for our purposes as we wish the system to be close to real time.

1.2.4. Profiling

To investigate this speed issue, we profiled the code, as can be seen in Figure 1.6. We found that the program spent 90% of it's time in the Stage 3 optimiser closure, 56% of it's time performing the backwards step and 32% of it's time in the rollout function. Hence it was clear that the speed issue was due to the slow act of rolling out and the large computation necessary to perform the backwards step on the large computation graph resulting from the rollout.

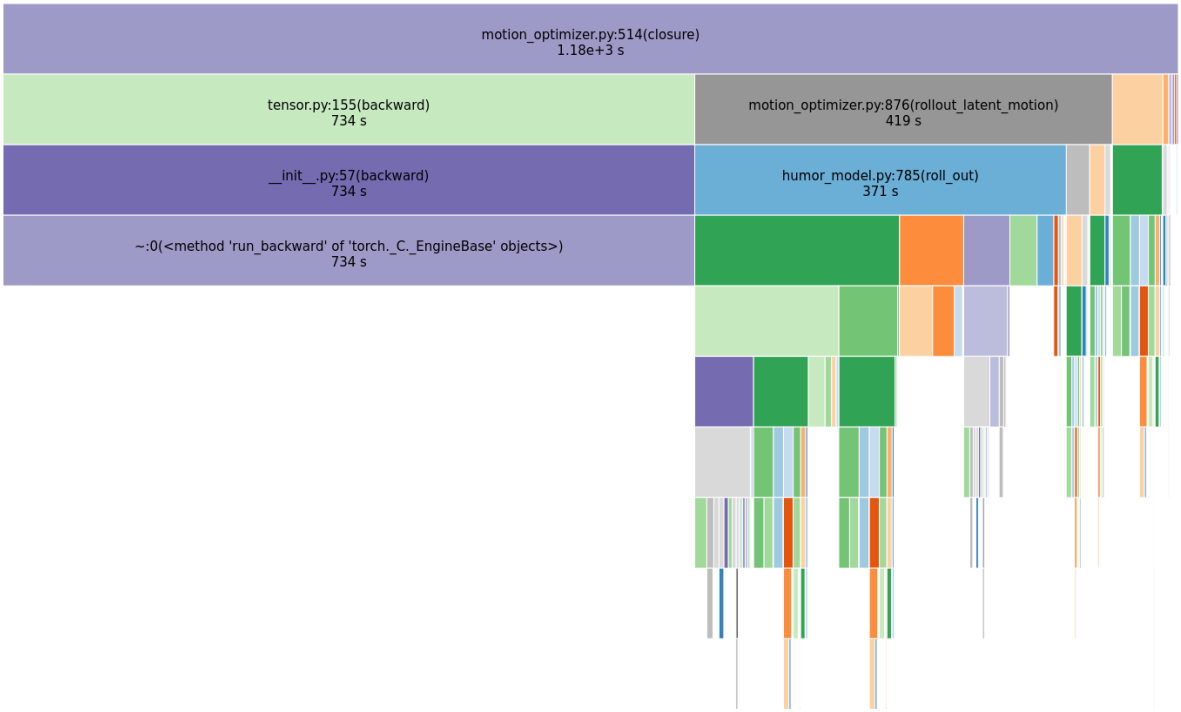


Figure 1.6.: *TestOps profiling*

1.2.5. Conclusion

We came to the final conclusion that the HuMoR motion model alongside the TestOps optimisation showed potential, but that it must be sped up before we can conclusively say if it is of particular use to us. We note that to speedup the method, we must primarily focus on improving/removing the rollout. Thus, in the next section, an attempt to speed up the TestOps is made.

1.3. Improving HuMoR TestOps

This section presents the attempt that was made at modify the TestOps of the HuMoR paper [RBH+21] in such a way that we attain comparable results but in a significantly shorter timespan.

TODO: See if I have introduced the terms properly (decoded sequence, x's, x''s, etc.)

1.3.1. Speeding up the TestOps

Realising that the main speed limitations are due to the concept of rollout over the entire sequence, we decided to break this long range dependence. Through short overlapping rollouts, it is possible to acheive more parrallelisation and smaller computation graphs, and our intuition suggests that the need for such a large context window is not necessary, that only a small number of autoregressed frames are needed to be able to propagate information sensibly and to get meaningful gradients. Anecdotaly, it would seem intuitive that if a video contains sitting, then walking, then sitting again, the second act of sitting would not depend in on the early act of sitting, hence rolling out over the whole sequence seems unnecessary.

The HuMoR TestOps rollout is presented graphically in Figure 1.7. x is the state, and z are the latent variables, the optimised variabels are highlighted in pink, and the losses are coloured, gray x's indicate they are calculated during the optimisation from the optimised variables. As can be seen, first all the states are autoregressed from the initial state and the latent variables, and then the losses are applied. This can be seen to result in very long range dependencies in the computation graph, thus making the gradients time consuming to calculate.

TODO: Update Figure 1.8 and remove the 'NEW' norm, labels on green loss, and potentially red loss?

The most obvious way to break this autogregression is to maintain and optimise a separate sequence of x's, updating this seperate sequence with reference to the x''s decoded from the latent variables as in Figure 1.8. We can take into account the decoded x''s in several manners:

- Copy over
  - Directly replace the optimised sequence of x's with the decoded x''s.
- Blend
  - Perform a weighted addition of the optimised x's and decoded x''s.
- Loss term
  - Add a loss term on the difference between the optimised x's and decoded x''s.

All decoding steps can now be performed in parrallel, rather than sequentially, which greatly reduces the computation time.

This new method also allows us to experiment with different amounts of rollout. We can once again decode the decoded sequence of x''s to get x'', and so on and so forth. These extra sequences are all decoded with the same latent variables thus the updates to the latent variables will take into account the losses on all the decoded sequences. Graphically this can be seen in Figure 1.9, where each next line . TODO: describe the figure better

The information contained in each x can flow forwards through the short range rollouts, and over the iterations of the optimiser. Note however that the information flow will be slower than the HuMoR TestOps, as in the TestOps all subsequent x's are regressed from the initial x_0. In the HuMoR TestOps information can flow backwards from any future frame's x in one update through the gradients, as all x's are linked in the computation graph through the auto-regression (though our intuition suggests that the information does not actually need to flow so far, as

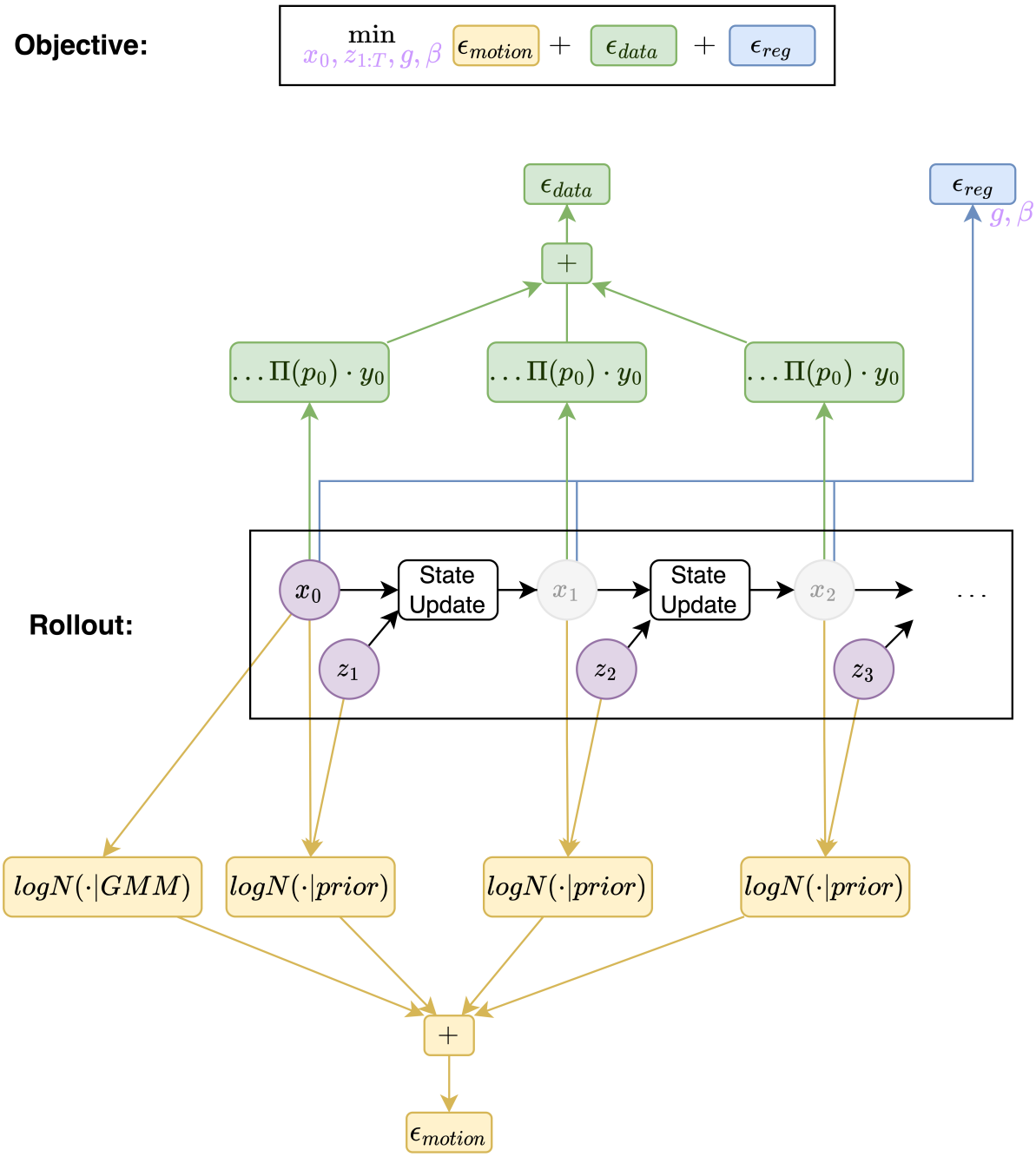


Figure 1.7.: *TestOps Computation Graph*

previously discussed). In the new method however, the information flow will be much more limited, as the rollouts will be significantly shorter, though again the information should flow through the iterations fo the optimiser.

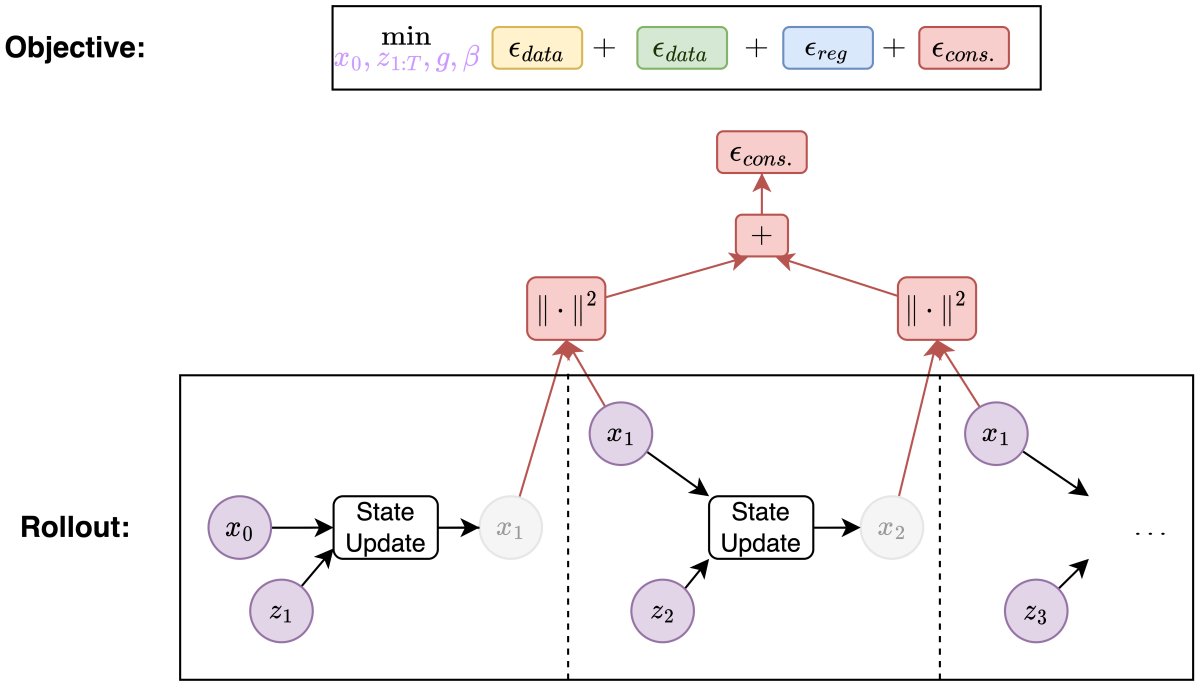


Figure 1.8.: *Decoupled Computation Graph*

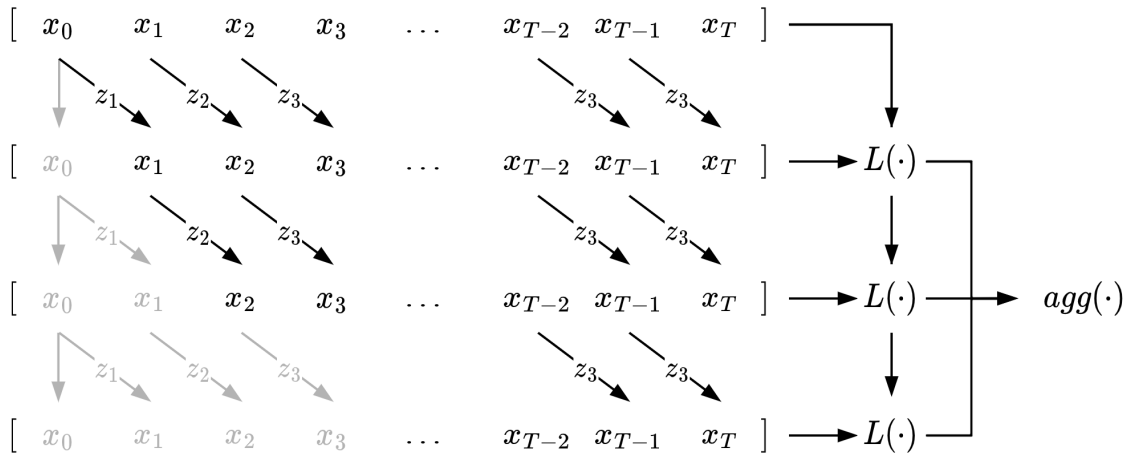


Figure 1.9.: *Decoded sequences*

1.3.2. Implementation notes

A number of issues were encountered during the implementation that are worth mentioning. After implementing the new method, attempts were made to get the optimiser to produce sensible results. The general approach was to use a small number of losses, so as to get a better intuition of each loss, with the goal of better understanding how to balance the losses in the optimiser, but some errors in individual losses were encountered.

Firstly, as expected, the Axis-Angle representation became a problem. When running through functions to convert to root-local reference frame for the HuMoR model for decoding (as it

operates on said reference frame), and then back to world reference frame for the opftimiser, the Axis-Angle vector flipped direction. Though the flipped vector still represented the same rotation, the angle representations of the same angle in the optimised sequence of x's and the decoded sequence of x''s were different, hence encorporating the information from decoded sequence cause the angle to move somewhere between the two flipped values which was no longer a sensible angle representation, thus resulting in root flips and other strange effects.

Secondly, an issue due to the SMPL model was found. When using a 2d reprojection loss (comparing the projected SMPL joints to the OpenPose 2d predictions) on a sequence where only the left arm and face had any OpenPose predictions, the legs were being moved by the optimiser. This was wrong as there were only losses on the arm and face joints, hence no gradients should flow to the legs. It was eventually found that the skinning weights of the SMPL model [BKL+16] contain spurious long range connections. 3% of the LBS (linear blend skinning) weights in the SMPL model are non-zero but less than $1e-2$, which seems rather too low to have any meaningful effect on the skinning, but which allows for spurious gradients to flow.
For the comparison to the OpenPose predictions the OpenPose skeleton must be obtained from the SMPL mesh. Certain joints are regressed, other are simply taken directly as vertices from the mesh, as in the case of the nose joint in the OpenPose skeleton which is taken to be vertex 332 [SMP]. We noted that this vertex 332 is skinned 99.8% by the 'head' joint, but also 0.2% by the right hip, left knee and right knee. This issue was solved by pruning all weights below 1e-2. It is interesting to note that there are known spurious connections in the SMPL blend shapes [OBB20], but that we have found no reference to spurious connections in the LBS weights.

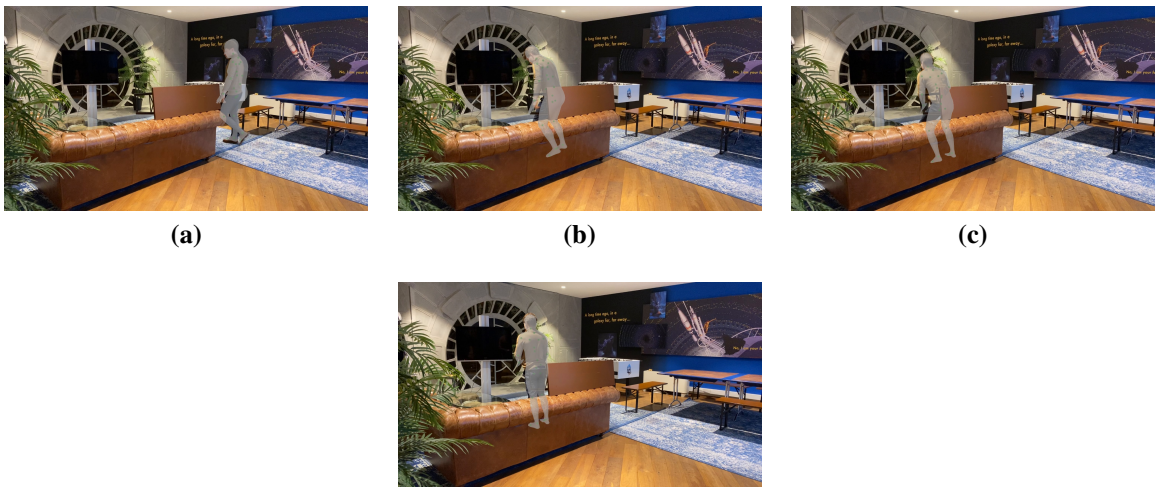
1.3.3. Experiments

TODO: explain why the initial rollout was bad, and the compounding of errors forward TODO: maybe we should discuss more in the humor investagtion section about the bad initial rollout TODO: Check everything is explicitely named, e.g have a graphic clearly naming the optimised x's and the decoded x's, and what the z's are

To begin with, we looked into the initial z's that were obtained from projecting the Stage 2 results through the HuMoR encoder. We found that in occluded situations these z's encode a sitting motion without any optimisation as seen in Figure 1.10, that simply the projection leads them to encode the necessary occluded motion. However we note that when autoregressively rolled out over a longer sequence the results, in Figure 1.12, deviate largely from the initial sequence of x's seen in Figure 1.11. This shows us that though the initial z's encode some sensible motion locally, and can be used to accurately recover the next frame, when they are chained together small deviations from the x motion accumulate to create an overall deviation. For example if the arm is moved slightly too much by a single z, then all the future frames will have the arm in slightly the wrong position, and multiple z's are wrong in the same direction, then the arm will raise up. This deviating sequence is the initial starting rollout in the Stage 3 of the HuMoR TestOps in which the rollout is optimised, though it seems unfortunate to us that from a sensible sequence of x's obtained from Stage 2 we get such a bad sequence through the autoreggresive rollout. We do however note that though while this starting point represents a sequence that deviates largely, the z's might not need to move so far to avoid the accumulation

effect. The intuition in our case was that we are starting from some sensible x's and z's, and therefore that it should be possible to refine the x's by take into account the HuMoR model through the z's whilst updating the z's.

Figure 1.10.: *Stage 2 z's encode a sitting motion*

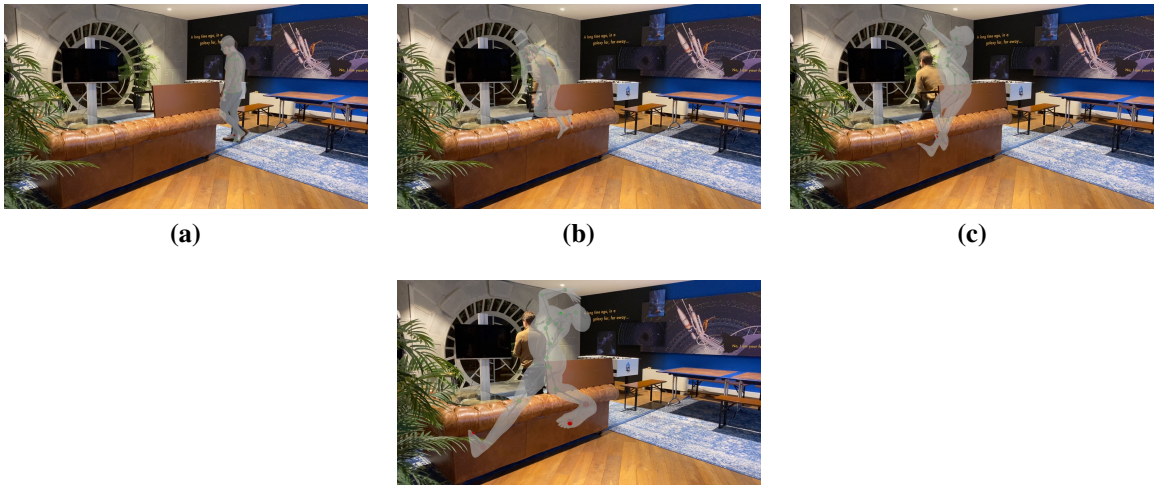


(a) **(b)** **(c)**

(d)

Figure 1.11.: *Stage 2 x's*

With some preliminary experiments in which we began optimising x's and z's we found that the most successful approach to blending the decoded and optimised x's was to have an L1 loss on the distance between them. We saw that when simply copying over, the propagation of small errors forward was too strong (as desribed in Figure 1.12) and the system failed to recover as the long range gradients present in the HuMoR TestOps method weren't present, and



(a) **(b)** **(c)**

(d)

Figure 1.12.: *Deviation due to rollout of Stage 2 z's*

we found when blending rather than completely copying over that the weighting of the blending was another parameter that was difficult to choose, hence having a loss was the path of least resistence.

Next we experimented with jointly optimising x and z with a larger variety of losses. It was noted that the sitting down motion was not acheived consistenly. However as previously mentioned, the initial z's encode a sitting motion, thus we conject that the x's and z's were fighting and this sitting motion was lost, that is to say the z's move to match the x's motion in which the skeleton simply clips into the floor without bending legs, thus the z's begin encoding less leg bending. To avoid this issue in future experiments, the z's were detached from the consistency loss making the optimised x's match the decoded x's.

This led us on to some experiments in which we either optimise z whilst fixing x, or vice versa. With the z's fixed at the stage 2 results, we managed to acheive a sensible sitting motion for a clip containing occluded sitting, but when we applied the same optimiser settings to a longer clip the results were no longer satisfactory. We also noted that when it did work, it required a large number of iterations, in the 1000s, which took 10mins or more, thus the speedup was not as evident as hoped (though our method would continue scaling better). In the inverse case, fixing x to the final HuMoR optimised states and optimising z, we noted that we consistently converge in the direction of the final HuMoR z's, which indicates that we are optimising in the right direction, with the best results acheived at a rollout of 5 decoding steps.

All these experiments were performed with varying numbers of decoding steps, 1, 2, 5, and 10. We found that the effect of more stable results but slower compute times with longer rollouts did manifest as intuition would suggest.

These experiments suggest that the next direction to pursue would be an optimisation scheme in which we flip flop optimising z then x. However considering our difficulty in achieving consistent results for the optimisation of the x's with fixed z's, our feeling that this might not be the optimal solution to the problem, my desire to try a new direction, and the time constraints, we deemed it time to move on and try a new method.

1.3.4. Conclusion

We found that jointly optimising the x's and z's did not prove an easy task to solve, which may indicate that

- it is a particularly difficult problem, and therefore that the optimiser struggles to satisfy the constraints and find a good minimum for both x and z at the same time
- the coupling of the x's and z's in the HuMoR decoder is an important issue. We found it to be at least a minor issue as it was necessary to decouple the z's from the consistency loss (matching the optimised xs to the decoded x''s) to avoid the z's being modified in such a way as to fit the unoptimised x's at the beginning of the optimisation
- the optimiser is badly balanced

This at the very least suggests that a new decoder that does not couple x's and z's would be useful and worth investigating if a similar method is pursued.

In the investigations we found it possible to acheive sensible results on a given clip (for example fixing the unoptimised initial z's, we can a plausible sitting motion in a small video of sitting in occlusion) but that the same settings for the optimiser failed to acheive a good result on an alternative clip. This may indicate that

- again, this is a difficult problem that the optimiser will struggle to solve
- the optimiser is badly balanced

as before.

While many of the issues encountered may simply be due to a badly balanced optimiser, considerable effort was made to mitigate this potential issue, and therefore it is our impression that this problem is at the very least a difficult one. We have improved our intuition, fixed several issues, and understood more about the problem and our goals. We therefore can make a more informed decision with respect to which direction to pursue next.
We found that taking a system that was designed with autogregression in mind, and trying to parrallelise it, proved more difficult than expected. The experiments lead us to beleive that jointly optimising a sequence of poses and latent variables obtained through a single frame decoder may be a more difficult formulation of the problem than others. We therefore conclude that a method in which we model longer motion sequences may be more fruitful.

TODO: Double check through Notes/Current_experiments.md to see if there's anything else we can learn from the experiments

Appendix

Appendix intro

A.1. Section

TODO:

Bibliography

[BKL+16] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image, 2016.

[CHS+19] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[OBB20] Ahmed A. A. Osman, Timo Bolkart, and Michael J. Black. STAR: sparse trained articulated human body regressor. *CoRR*, abs/2008.08535, 2020.

[PCG+19] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[RBH+21] Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J. Guibas. Humor: 3d human motion model for robust pose estimation. In *International Conference on Computer Vision (ICCV)*, 2021.

[SLY15] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[SMP]

[ZBL+18] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks, 2018.