# Non-local Pose Means for Denoising Motion Capture Data

Christopher J. Dean*
Computer Graphics Group
Victoria University, Wellington, New Zealand
Email: christopherjosephdean@gmail.com

J.P. Lewis*
SEED, Electronic Arts and Victoria University
Email: zilla@computer.org

***Abstract*—Motion capture is commonly used in movies and games, and may become more widespread if anticipated virtual reality and augmented reality applications become popular. Unfortunately, body motion capture often suffers from significant noise resulting from intrinsic factors not present in other media such as images and videos. This paper adapts the non-local means (NLM) principle from image processing to noise reduction in motion capture data. We show that the NLM principle can be applied in this context by operating in the vector space obtained through a log/exponential map, thereby respecting the rotational nature of skeletal movement. The resulting algorithm is efficient, surprisingly effective, and requires no training or access to data other than the motion clip itself. Our results rival or outperform other techniques in a survey of standard denoising methods in signal processing.**

***Index Terms*—motion capture, noise, denoising, noise reduction, character animation**

## I. Introduction

Motion capture ("mocap") is widely employed in movies and games, and the Microsoft Kinect has made motion capture relatively commonplace. Future virtual reality and augmented reality online spaces may make motion capture ubiquitous.

Unfortunately existing motion capture systems produce results that are corrupted by noise to a greater or lesser extent. If this noise is not removed, the illusion of a virtual character presence is compromised. Human motion denoising is difficult because the body has a large number of degrees of freedom (often 100 or more, depending on the chosen model) with hierarchical rotational movement [1]. The 'manifold' of the movement is complex and non-Euclidean. Each of these factors is challenging in their own right; when coupled together, slight changes in the data can create unrealistic motion.

One cause of motion capture noise is noise at the sensor level, such as multipath effects in time-of-flight sensors, and inexact discrimination between the body or motion capture markers and the background. A more intrinsic source of noise arises due to the estimation of the skeleton pose. Here, it is necessary to estimate the position of body parts that are occluded, and any error in this estimate will result in a visible discontinuity or "pop" when the part becomes visible again. High-end motion capture systems reduce this problem through the use of many cameras (more than 100 may be used on a large motion capture stage). However the problem reappears in complex motion capture scenarios due to the occlusion caused by multiple actors in close proximity. In severe cases, such disocclusions can (depending on the skeleton solver) result a body part or whole character flipping orientation.

From Newton's second law, the velocity of the actor is the integral of applied forces including muscle forces applied by the actor. The actor's movement thus has a first-order smoothness almost everywhere. Traditional motion capture software exploited this by applying simple linear smoothing filters (Gaussian, Butterworth, etc.) to remove noise. Unfortunately, while the motion is smooth almost everywhere, it is not smooth everywhere. Foot steps and other impacts are the exception to smooth motion. It is crucial to retain these derivative discontinuities – failure to do so results in the tracked feet not being in precise contact with the floor, leading to the impression that a walking character is hovering, sliding, or slightly penetrating the floor. As eloquently argued in [2], no linear filter can simultaneously achieve the goals of fully removing high frequencies and preserving derivative discontinuities.

This has lead to more sophisticated nonlinear and data-driven methods. These are briefly surveyed in Section 2.

Our research aims to find a noise reduction solution that:

- preserves motion details with little blurring or skating,
- requires no external training data and relies only on the given animation,
- works at interactive computation speeds,
- reliably reduces noise for a variety of conditions.
- outperforms other denoising filters,
- is manifold-aware, whereby the procedure involves operations that respect the rotational nature of body movement.

To satisfy these goals, in this paper we introduce a new approach to denoising motion capture based applying the non-local-means principle in a manifold-aware fashion. Non-local means (NLM) is an image processing technique that performs a weighted average of similar but possibly non-adjacent image patches for detail-preserving denoising [3]. The principle assumes that similar image patches will have different noise, and thus a weighted average will reduce noise while preserving or enhancing detail. This detail preservation is important for motion capture—it will enforce spatio-temporal cohesion in

*authors contributed equally

the animation, preserving the 'texture' of the motion curves (i.e. high frequency motion details).

A direct application of NLM to our problem is poorly motivated, since NLM involves a weighted average based on *Euclidean* distance, which is inappropriate for body motion. To address this issue, we embed the NLM principle in a vector space obtained by the log map of the body pose.

The resulting non-linear pose means (NLPM) algorithm retains the strengths of its image-based predecessor. We demonstrate the results on data provided by the Carnegie Mellon University motion capture database [4]. The results of our method outperform a number of standard alternative approaches. While we target an entertainment industry scenario where the motion is denoised following capture, but there is nothing in the method that prohibits an on-line solution – this would only require restricting the patch window (Section III-A) to be causal.

In the following sections, we will outline related work in image and motion denoising, explain the non-local pose means method, and compare the NLPM results to standard benchmarks in this research domain.

## II. Background and Related Work

For the purpose of our review, generic imaging denoising techniques may be divided into the categories of linear filters, discontinuity-preserving methods, sparse-coding approaches, data-driven methods, and others. Discontinuity-preserving methods include bilateral filters [5], total-variation denoising [6], anisotropic diffusion [7], and others. The methods above make generic assumptions that may not be appropriate for actual data. For example, linear filters assume the noise has particular spectral content, whereas most edge-preserving smoothing algorithms assume a piecewise constant signal. Methods such as [8], [9] assume that the motion can be sparsely represented in some (possibly overcomplete) basis, and thus shrink or zero the small coefficients with respect to this basis. These methods are simple, efficient, and reasonably effective. Whereas wavelet shrinkage makes use of a data-agnostic basis, data-driven methods such as KSVD [10] build a basis or otherwise invoke properties or statistics of the particular image or signal. Some data-driven methods [9] also invoke the sparse-coding principle.

In our categorisation, NLM can be considered as a data-driven method.

### A. Motion Denoising

While the literature on image denoising is vast, the research literature on motion denoising is also sizeable and too large to fully survey here.

Many researchers have applied the aforementioned image filters to motion capture, and this has provided common benchmarks for comparison in this research area. In motion denoising, researchers have come to prefer systems aware of the spatio-temporal motion characteristics, such as dynamical systems [11]. Linear dynamic systems (LDS) model observed measurements as noisy linear projections that evolve via a low-dimensional dynamic process—such as a Markov process, in which the state evolution of a system is dependent on a finite set of previous states [12], [13]. Temporal lag is a problem in dynamical systems, as a time delay will offset the data [14], [15]

Data-driven methods have been noted as the most successful approach to motion denoising [1], [15]. Lou and Chai represent the motion in terms of eigenvectors of a time-lapped covariance matrix, and use a robust norm to reconstruct smoothed motion without outliers. It denoises extremely well, but only when the database contains similar 'families' of motion [15]. Xiao et al. [15] adopt ideas from K-SVD and sparse coding, operating on overlapping segmented regions called poselets. Feng et al. [14] similarly operate on a poselet representation. These methods outperform the previous approaches, but rely on pre-cleaned motion input libraries to function properly.

*1) Denoising in Commercial Animation Software:* Commercial motion-capture editing software occasionally offers some forms of denoising. Blender, Maya, Motion Builder [16], and other major animation software titles often rely on simple signal processing and are seldom bundled with extensive denoising features. Motion Builder is designed specifically for motion editing and contains the most comprehensive collection of animation data tools: Gaussian, Butterworth, smoothing, resampling, and other rudimentary signal processing functions are available for minor edits. Blender 3D features a simple weighted moving means function filter for smoothing bumps in motion curves. This tool is very handy for eliminating basic noise. We will compare Blender's results to our own in the evaluation section.

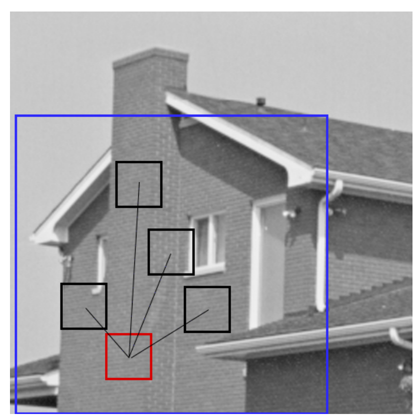


Fig. 1: An example of the non-local means algorithm. A noisy pixel *(centre of red square)* is denoised by comparing its local patch to other similar patches of pixels *(black squares)* inside the search radius *(blue square)*.

### B. Non-local Means

Whereas a typical image or signal filter computes a weighted average of surrounding pixels or samples, the NLM principle computes weighted average of pixels taken from similar but discontiguous neighbourhoods (Fig. 1), with weights determined by the similarity of the patches. The principle is that most small patches in an image are similar to other patches elsewhere in the image [3]. For example, a portion of a brick texture will be similar to other parts of the same texture, and a small portion of the edge of a leaf may closely resemble portions of the edges of other leaves in the image. By forming an average of these patches, the per-patch noise is averaged away while detail is often enhanced.

In more detail, given an image and and a specified search window $I$ of radius $r < min(width(I), height(I))$, for each pixel, form the patch $p$ of surrounding pixels and then create the dictionary of the $k$ most similar pixel patches $q_j, j \leq k$ within the search window. Then non-local means can be expressed as:

$$p_{\text{filtered}}(x,y) = \frac{1}{C(p)} \sum_{q \in I} q(x,y) f(p,q), \tag{1}$$

where $(x,y)$ are pixel coordinates and $C(p)$ is the sum of the (typically) Gaussian weights,

$$f(p,q) = e^{-\frac{\|q-p\|^2}{2\sigma^2}}. \tag{2}$$

Put simply, the output pixel is a normalised sum of the most similar pixel patches in a search window. When $p = q$, the Gaussian weight is equal to 1. To prevent a pixel from weighting itself too highly, we instead assign it the maximum weight of the other pixel patches [17].

*1) Related Research for NLM:* Later research [18] has categorised NLM as a semi-local filter, rather than a truly non-local one. Results are dependent on input image structure, but the best output is usually constrained within a smaller search window. As the training area tends toward non-locality by expanding to the size of the image, the MSE declines for many typical examples of pictures. This phenomenon is due to the large number of small weights contained within the oversized training window [18]. Too many weights, although insignificant, lead to the averaging of dissimilar patches.

Conversely, in periodic images where patterns repeat themselves at a larger scale, increasing the radius of the search window has a profound positive impact on mean-squared error (MSE). When we later apply this technique to motion capture, human motion can be both periodic and non-periodic, so it is necessary to choose a locality measure for case-by-case denoising.

Fig. 2 shows non-local means correcting a slightly corrupted image. There is no visible blurring, and all of the noise appears to be eliminated.

The high-dimensional nearest-neighbour search implicit in (1) is costly and has been accelerated with a number of schemes in recent research. Goossens et al. [19] criticise the algorithm for its $O(n^4)$ complexity, which is impractical in a large 2D image arrays. A later paper introduces an accelerated scheme [20]. Other research enhancements to NLM include FFT-inspired acceleration [21], integration with the Laplacian Pyramid [21], and a GPU-based implementation [22].

Goossens et al. also observe that NLM is the first iteration of a Jacobi algorithm, and offer improvements from this method.

Fig. 2: Non-local means denoising of an image. Denoised output image *(left)*, from the badly noised input image *(right)*.

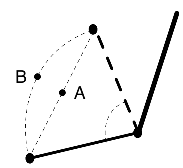
While they remark that NLM is not able to compete with the recent trends in sparse-coding methods, their NLM method is able to produce comparable results.

## III. NON-LOCAL POSE MEANS

This section details our contribution to motion capture denoising using the non-local means algorithm. We explain our method for a novel adaptation of the image processing technique to process n-dimensional rotation vectors.

### *A. Pre-processing Poses*

The NLM algorithm in (1), (2) involves two operations that are not suitable for body pose data. The Euclidean distance $\|q - p\|$ in (2) does not respect the rotational nature of body poses. However, it is monotonically related to the correct distance, and so can be used in the search for similar patches. The addition operation in the weighted sum (1) presents a more serious problem since it is used to compute the result. For example, a convex sum of two poses of a limb results in the limb changing length (point $\mathbf{A}$ in the adjacent figure) rather than showing rotational motion (point $\mathbf{B}$).




The quaternion representation of rotations commonly used in robotics, vision, and graphics does not solve this problem. The product of quaternions is a rotation, however the weighted sum of quaternions is not generally interpretable as a rotation. To handle rotations correctly, we use log quaternions [23]. The log quaternion represents rotations as vectors whose direction is the axis of rotation and magnitude is the rotation angle. Importantly, linear combinations of log-quaternions produce valid rotations. Following the application of the weighted blending analogous to (1), (2) but using a pose vector described below, the result is exponentiated. The result is in effect an exponential map for rotations, wherein the log-quaternion representation provides a vector space structure [24].

In general the exponential map is a diffeomorphism only in a neighbourhood of the identity. This is not a problem in NLPM because the poses that are averaged in (1) are chosen to be similar. On the other hand, calculation of the log quaternion can be unstable for extremely small rotations, which might arise with NLPM. [23] provides an alternate computation that addresses this issue. The subject of rotations is complex and there are other forms of exponential map and other approaches to averaging rotations. Our approach resembles a single step of the gradient descent in a Karcher (manifold) mean [25]. We found the log-quaternion approach to be simple and effective.

Much like the image-based NLM, we begin by preprocessing the motion capture poses into patches of pose vectors. The patch radius $r_p$ determines the number of included poses from the animation timeline, forming a patch of size $(2r_p + 1)$. The patch vector is expressed as:

$$\mathbf{P}(t) = [p(-r_p), \cdots, p(-1), p(0), p(1), \cdots, p(r_p)]^T$$

for pose $p(t)$ and patch $\mathbf{P}(t)$ at time $t$ in the mocap timeline. When we calculate weights for points beyond the edges of the data set, we optionally employ mirroring, Neumann (data boundary gradient is held constant), or Dirichlet ( values are assumed constant) boundary conditions [26]. The choice of boundary condition should depend on the nature of the motion bounds.

### B. The Non-local Pose Mean Filter

*1) Locality on the Manifold or Time Domain:* The choice of learning window determines the type of locality our algorithm covers. A direct implementation of non-local means would place the learning window bounds as the k-nearest pose frames: $p(t-k) < p(t) < p(t+k)$ for learning window $\mathbf{T}$ of radius $k$ frames.

*2) The Pose Mean:* The remainder of our algorithm follows the original NLM scheme. Instead of a weighted-average pixel, we will find the average pose determined by distance weights. This average pose represents the average collection of spherical joint rotations, i.e. the average pose on the animation manifold. To denoise a pose corresponding to the patch $\mathbf{P}(t)$, we calculate the Gaussian weights for all other patches $\mathbf{Q}(t)$:

$$f(p,q) = e^{-\frac{\|\mathbf{Q}-\mathbf{P}\|^2}{2\sigma^2}}. \tag{3}$$

The poses can be averaged via their weights,

$$\mathbf{P}_{\text{filtered}}(x,y) = \frac{1}{C(\mathbf{P})} \sum_{\mathbf{Q}\in\mathbf{T}} \mathbf{Q}(t) f(\mathbf{P},\mathbf{Q}), \tag{4}$$

once again taking care to set $\mathbf{P}(t)$'s weight to the maximum weight of its neighbours.

## IV. Results and Evaluation

Here we present the results of our method, and evaluate the NLPM noise reduction quality against some of the methods discussed in the related works section. We demonstrate the results on noisy data provided by the Carnegie Mellon University motion capture database [4].

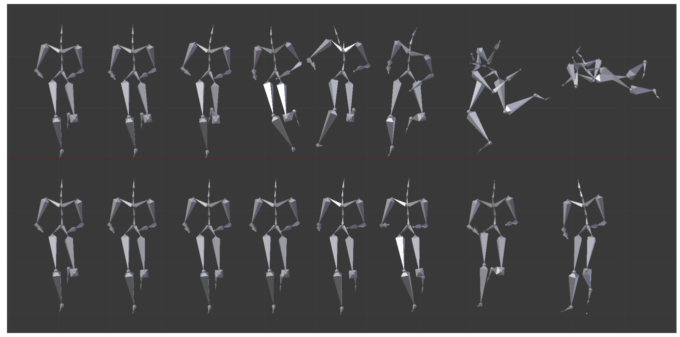


Fig. 3: Non-local means applied to noisy motion capture data. Top: Original data, with increasing degrees of Gaussian noise from left to right (ranging from $1\sigma$ to $100\sigma$). Bottom: The recovered motion for each respective animation, after applying non-local means.

### A. Non-local Pose Means

Fig. 3 highlights our method's noise reduction capabilities for light to drastic jitters in a skeleton. The top row of animated rigs feature a range of Gaussian additive noise, added individually to each joint in the animated skeletal rig. Non-local means can recover up to $\sigma = 1$ of normally-distributed rotational noise while retaining the original motion. Beyond this, even in the most extreme case of noise degradation the original motion can be identified, although some distortion is inevitable.

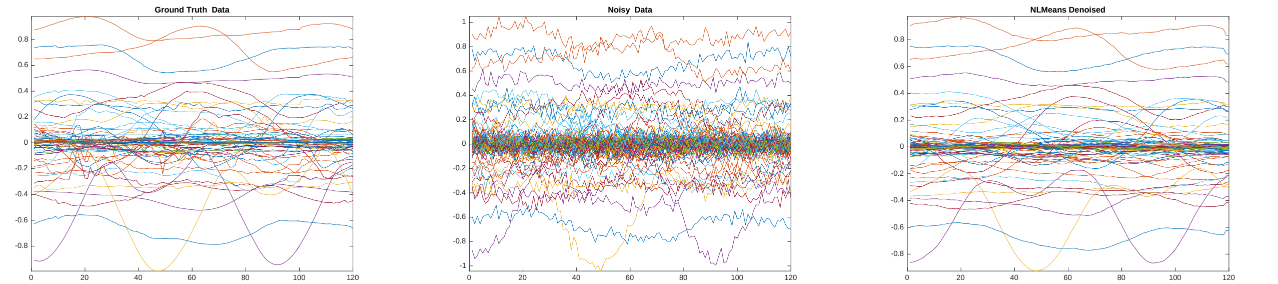
A strength of this algorithms is the ability to retain the physics and timing constraints of the original motion. In both low and high noise conditions, the foot contact remains solid. This is due to the NLPM's spatio-temporal gradient preservation, which preserves 'hard edges' in the motion curves.

### B. NLPM Benchmark Comparisons

Figure 4 shows the ground truth data, the noisy motion data, and the NLPM denoised output. The output looks much like the original, with a single sharp bend for all curves on the far right side of the graph. This is due to a poor choice of boundary condition, where mirroring causes abnormal weight distributions. Generally, the Dirichlet condition (i.e. assuming the gradient remains constant) works best.

In Figure 5, a single motion channel is isolated from these dense figures, and we compare the NLPM result to ground truth and noise signals. We can note some loss of precision, and a small degree of temporal shifting (e.g. Fig. 5 at frame 90).

Table I compares the peak signal to noise ratios (PSNRs) of the common motion denoising techniques that we discovered in the literature review. We add normally-distributed noise to the quaternion poses of the CMU library motion capture data and compare the mean-squared error of the denoised output to the original ground truth. For each denoising method, care was taken to find the optimal input parameters for the smoothest result, without sacrificing motion detail. We

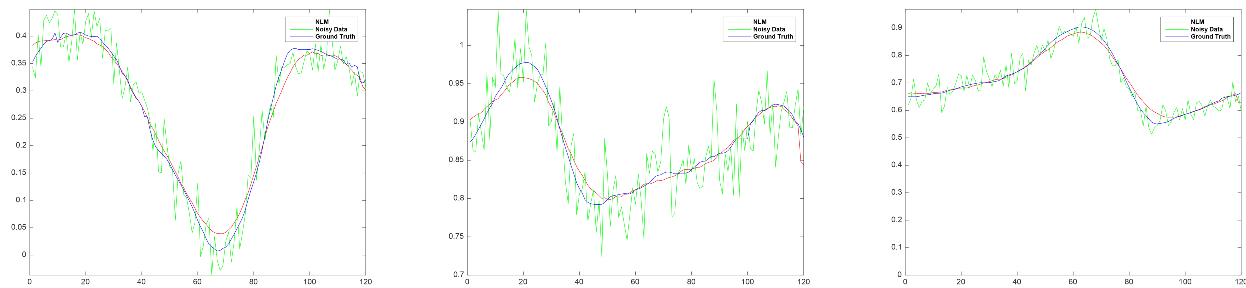



(a) Ground Truth curves for all rotations in the motion graph

(b) Added Gaussian noise to motion curves (rotational $\sigma = .4$) for all rotations in the motion graph

(c) Non-local pose means solution for all rotations in the motion graph, using noisy curves from (a) as input

Fig. 4: Results Comparison For All Motion Channels




(a) NLPM vs. Ground Truth vs. Noisy data for a single channel of joint rotation data.

(b) NLPM vs. Ground Truth vs. Noisy data for a single channel of joint rotation data.

(c) NLPM vs. Ground Truth vs. Noisy data for a single channel of joint rotation data.

Fig. 5: Results Comparison for Single Motion Channel

PSNR, when compared to ground truth

| | $\sigma$ | $15\sigma$ | $30\sigma$ | $50\sigma$ | $75\sigma$ | Impulse |
|---|---|---|---|---|---|---|
| Gaussian LPF | 73.89 | 73.87 | 73.76 | 73.33 | 72.02 | 73.61 |
| Kalman | 80.46 | 80.42 | 78.50 | 73.83 | – | 80.01 |
| Blender WMM | 93.39 | **92.53** | 84.99 | 84.40 | 78.99 | **88.61** |
| Wavelet | **93.76** | 67.01 | – | – | – | 82.59 |
| Butterworth | 82.21 | 82.14 | 81.69 | 80.30 | 77.21 | 81.39 |
| **Our method** | 85.58 | 85.52 | **85.29** | **84.68** | **82.46** | 85.18 |

TABLE I: PSNR Noise Elimination Result Comparison. The optimal result in each column has been emboldened or excluded where smooth data was not possible. $\sigma = .008$, added in quaternion log-space.

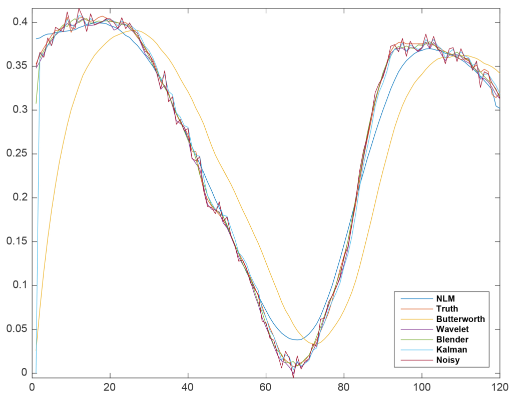
used Matlab signal processing libraries for Gaussian, wavelet thresholding, and Butterworth denoising. The python Kalman filter implementation was borrowed from Bishop and Welch [27], where it was optimised specifically for motion denoising research.

We denoised a running character from the CMU library and applied NLPM with a learning window of 21 poses, a patch size of 11 poses, and a Gaussian width ($\sigma = .008$). In every noise category, our method produced superior denoised-signals that most resembled the ground truth data. As noise degradation consumes the original data, PSNR slowly and steadily declines in our method. In the most extreme noise, the animated output appears smooth, but the physical actions are visibly different. At $50\sigma$, smoothed trembles appear for the more prominent bones. In the other noise conditions, it is difficult to visibly spot any difference between the denoised animations.

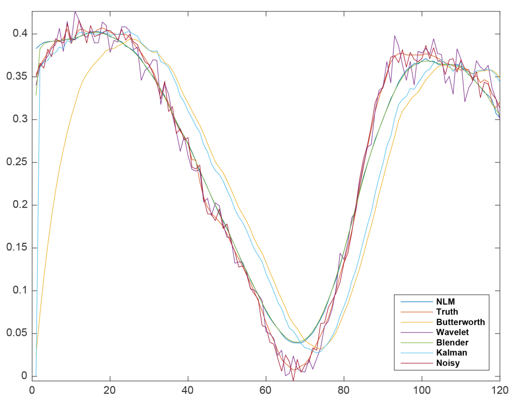
Blender's weighted moving means algorithm performs equally well as or better than NLPM for small levels of noise degradation, but worsens as noise increases. Our input parameters were optimised for medium to high-levels of noise; with further experimentation, NLPM may be able to match Blender's performance in the lower noise conditions. Regardless, the strong performance of NLPM against professional-grade software is indicative of our method's limited but promising success.

Figures 6b and 6b depict an overlaid comparison of the best denoising methods, for two different noise conditions. Between these two graphs, NLPM shows the best consistency and fewest artefacts. Blender's method declines in quality, as it derails from the ground truth between the two graphs. The minimum of NLPM does not reach the same minimum as the ground truth in this curve; this is an example of too many dissimilar poses watering down the result with their accumulated weights. A better non-local manifold parameterization may correct this, but we must leave this optimisation for future work.

Fig. 6: Results




(a) An overlaid comparison of all denoising methods from Table I, for the light noise condition ($15\sigma$).




(b) An overlaid comparison of all denoising methods from Table I, for the medium noise condition ($30\sigma$).

## V. Summary

We have examined image and signal filters for reducing noise in motion capture. Our application of NLM rests on preprocessing pose vectors from a manifold to a vector space using the exponential map. We then apply non-local averages, weighted by pose distance, to denoise the data.

NLPM produces high-quality denoised animation curves without compromising detail. Our method consistently outperforms the published approaches listed in Table 1. Surprisingly, Blender's weighted moving means algorithm outperformed NLPM for small amounts of noise, however our method outperforms Blender for moderate and large amounts of noise. Please see the associated videos for additional results.

## References


[1] H. Lou and J. Chai, "Example-based human motion denoising," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 5, pp. 870–879, 2010.

[2] A. Blake and A. Zisserman, *Visual Reconstruction*. Cambridge, MA, USA: MIT Press, 1987.

[3] A. Buades and B. Coll, "A Non-local Algorithm for Image Denoising," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, no. 0, pp. 60–65, 2005.

[4] "CMU motion capture library database," Carnegie Mellon University, [Online.] Available: mocap.cs.cmu.edu. Accessed 7- January- 2016.

[5] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proceedings of the Sixth International Conference on Computer Vision*, ser. ICCV '98. Washington, DC, USA: IEEE Computer Society, 1998, pp. 839–. [Online]. Available: http://dl.acm.org/citation.cfm?id=938978.939190

[6] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D*, vol. 60, no. 1-4, pp. 259–268, Nov. 1992. [Online]. Available: http://dx.doi.org/10.1016/0167-2789(92)90242-F

[7] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 12, no. 7, pp. 629–639, 1990.

[8] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Trans. Inf. Theor.*, vol. 41, no. 3, pp. 613–627, May 1995. [Online]. Available: http://dx.doi.org/10.1109/18.382009

[9] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *Trans. Img. Proc.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007. [Online]. Available: http://dx.doi.org/10.1109/TIP.2007.901238

[10] M. Aharon, M. Elad, and A. Bruckstein, "Svdd: An algorithm for designing overcomplete dictionaries for sparse representation," *Trans. Sig. Proc.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006. [Online]. Available: http://dx.doi.org/10.1109/TSP.2006.881199

[11] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Gaussian process dynamical models for human motion," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 2, pp. 283–298, 2008.

[12] L. Li, J. Mccann, N. Pollard, C. Faloutsos, L. Li, J. Mccann, N. Pollard, and C. Faloutsos, "Bolero: A principled technique for including bone length constraints in motion capture occlusion filling," in *In Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2010.

[13] L. Li, J. Mccann, N. Pollard, and C. Faloutsos, "Dynammo: Mining and summarization of coevolving sequences with missing values."

[14] Y. Feng, M. Ji, J. Xiao, X. Yang, J. J. Zhang, Y. Zhuang, and X. Li, "Mining Spatial-Temporal Patterns and Structural Sparsity for Human Motion Data Denoising," *IEEE Transactions on Cybernetics*, vol. 45, no. 12, pp. 2693–2706, 2014.

[15] J. Xiao, Y. Feng, M. Ji, X. Yang, J. J. Zhang, and Y. Zhuang, "Sparse motion bases selection for human motion denoising," *Signal Processing*, 2015.

[16] Autodesk, Inc., "Autodesk motionbuilder," version 2016. [Online]. Available: https://www.autodesk.com/products/motionbuilder

[17] D. Raghuvanshi, S. Hasan, and M. Agrawal, "Analysing Image Denoising using Non Local Means Algorithm," *International Journal of Computer Applications*, vol. 56, no. 13, pp. 7–11, 2012.

[18] J. Salmon, "On two parameters for denoising with non-local means," *Signal Processing Letters, IEEE*, vol. 17, no. 3, pp. 269–272, 2010.

[19] B. Goossens, H. Luong, A. Pizurica, and W. Philips, "An improved non-local denoising algorithm," *2008 International Workshop on Local and Non-Local Approximation in Image Processing*, 2008.

[20] A. Dauwe, B. Goossens, H. Luong, and W. Philips, "A fast non-local image denoising algorithm," *Proceedings of SPIE - The International Society for Optical Engineering*, 2008.

[21] Y.-L. Liu, J. Wang, X. Chen, Y.-W. Guo, and Q.-S. Peng, "A robust and fast non-local means algorithm for image denoising," *Journal of Computer Science and Technology*, vol. 23, no. 2, pp. 270–279, 2008.

[22] K. Huang, D. Zhang, and K. Wang, "Non-local means denoising algorithm accelerated by gpu," in *Sixth International Symposium on Multispectral Image Processing and Pattern Recognition*. International Society for Optics and Photonics, 2009, pp. 749 711–749 711.

[23] F. S. Grassia, "Practical parameterization of rotations using the exponential map," *Journal of graphics tools*, vol. 3, no. 3, pp. 29–48, 1998.

[24] K. Anjyo and H. Ochiai, *Mathematical Basics of Motion and Deformation in Computer Graphics*, 2nd ed. Morgan & Claypool, 2017.

[25] X. Pennec, "Probabilities and Statistics on Riemannian Manifolds : A Geometric approach," INRIA, Tech. Rep. RR-5093, Jan. 2004. [Online]. Available: https://hal.inria.fr/inria-00071490

[26] S. B. Gueye, K. Talla, and C. Mbow, "Solution of 1d poisson equation with neumann-dirichlet and dirichlet-neumann boundary conditions, using the finite difference method," *Journal of Electromagnetic Analysis and Applications*, vol. 6, no. 10, p. 309, 2014.

[27] G. Welch and G. Bishop, "An introduction to the kalman filter," 1995.