Motion Control via Metric-Aligning Motion Matching

NAOKI AGATA, The University of Tokyo, Japan
TAKEO IGARASHI, The University of Tokyo, Japan

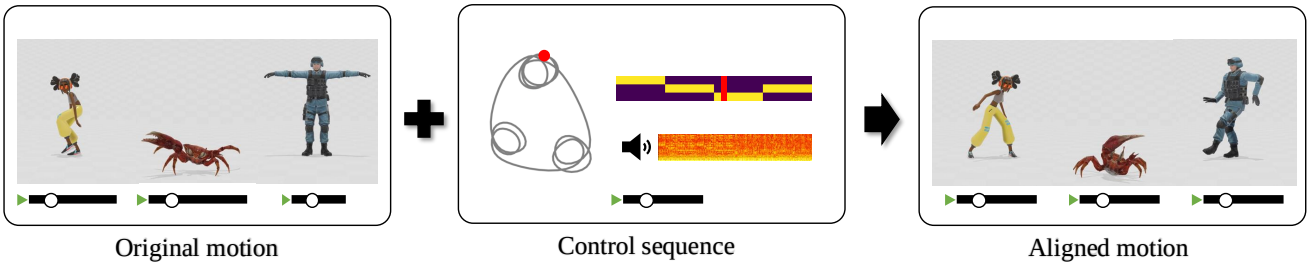


Fig. 1. Overview of our method. Our method aligns given original motion to given control sequence. Our method can take arbitrary control sequences, such as sketches, labels, audio, and motions. Our method solely relies on within-domain distance in original and control, without needing manual definition of mapping or training with annotated data.

We introduce a novel method for controlling a motion sequence using an arbitrary temporal control sequence using temporal alignment. Temporal alignment of motion has gained significant attention owing to its applications in motion control and retargeting. Traditional methods rely on either learned or hand-craft cross-domain mappings between frames in the original and control domains, which often require large, paired, or annotated datasets and time-consuming training. Our approach, named *Metric-Aligning Motion Matching*, achieves alignment by solely considering within-domain distances. It computes distances among patches in each domain and seeks a matching that optimally aligns the two within-domain distances. This framework allows for the alignment of a motion sequence to various types of control sequences, including sketches, labels, audio, and another motion sequence, all without the need for manually defined mappings or training with annotated data. We demonstrate the effectiveness of our approach through applications in efficient motion control, showcasing its potential in practical scenarios.

CCS Concepts: • **Computing methodologies** → **Motion processing**.

Additional Key Words and Phrases: character animation, motion control, motion editing interface, optimal transport

ACM Reference Format:
Naoki Agata and Takeo Igarashi. 2025. Motion Control via Metric-Aligning Motion Matching. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '25), August 10–14, 2025, Vancouver, BC, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3721238.3730665

Authors' addresses: Naoki Agata, The University of Tokyo, Bunkyo-ku, Tokyo, Japan, agata-naoki@is.s.u-tokyo.ac.jp; Takeo Igarashi, The University of Tokyo, Bunkyo-ku, Tokyo, Japan, takeo@acm.org.

SIGGRAPH Conference Papers '25, August 10–14, 2025, Vancouver, BC, Canada

ACM ISBN 979-8-4007-1540-2/2025/08
https://doi.org/10.1145/3721238.3730665

1 INTRODUCTION

The problem of aligning a motion sequence with a control sequence has gained increasing attention owing to the wide range of tasks it enables. Applications include, for instance, controlling motion based on a user's dynamic input, retargeting human movement to a character with a different skeletal structure, and synchronizing dance motion with audio structure. However, these tasks present significant difficulties, often demanding precise temporal correspondence and structural coherence between control sequences and motion sequences, even when the control and motion domains differ significantly in structure, dimension, duration, or complexity.

Early studies on motion-to-motion retargeting relied on handcrafted mapping functions and strategies [Choi and Ko 2000; Gleicher 1998; Lee and Shin 1999; Rhodin et al. 2015; Seol et al. 2013; Tak and Ko 2005; Yamane et al. 2010]. More recent approaches leverage large-scale learning-based frameworks to learn direct mappings between two motions [Li et al. 2023b; Lim et al. 2019; Villegas et al. 2018] or shared embeddings between two domains [Aberman et al. 2020; Lee et al. 2023; Li et al. 2024a]. While deep learning techniques compute mappings empirically, they often require extensive annotated datasets and prolonged training to ensure robust generalization. Similarly, audio-driven motion synthesis [Alexanderson et al. 2023; Li et al. 2024b; Tseng et al. 2023] depends heavily on large paired datasets and substantial computational resources. Recent efforts to reduce data demands, as seen in works like GANimator [Li et al. 2022], GenMM [Li et al. 2023a], and SinDDM [Raab et al. 2023], demonstrate that motions can be generated from limited data. However, these methods still lack robust, built-in alignment strategies for complex temporal inputs and often require domain-specific constraints or supplementary annotations to measure cross-domain correspondences.

In this study, we propose a motion alignment technique that relies solely on within-domain distances, which can be naturally defined without requiring hand-crafted or learned mapping functions (Fig. 1). Our approach first computes distances among patches within the original motion and control sequences separately. We then search

for couplings between patches in the control sequence and those in the original motion such that the within-domain distance structure is preserved as much as possible. These couplings are used to compute a weighted blend of original patches, resulting in an aligned motion. Technically, our method uses metric alignment techniques based on the fused semi-unbalanced Gromov-Wasserstein (FSUGW) optimization objective [Xu and Gould 2024], a modified version of the Gromov-Wasserstein optimal transport (GW OT) problem. GW OT has been successfully applied in various applications, such as identifying correspondences between shapes [Solomon et al. 2016]. By leveraging advancements in GW OT algorithms, our method achieves cross-domain coupling without explicitly computing cross-domain distances.

This framework enables the alignment of motion sequences to various types of control sequences, facilitating a wide range of applications. These include: (1) waveform-to-motion and sketch-to-motion, allowing users to control motion sequences through parameterized 1D waveform signals and 2D hand-drawn sketches, and (2) motion-by-numbers, where users specify only the temporal segmentation labels. Our framework is also applicable to existing tasks, such as: (3) motion-to-motion alignment, and (4) audio-conditioned motion control.

Our contributions are summarized as follows:

(1) We introduce *Metric-Aligning Motion Matching (MAMM)*, a unified FSUGW-based optimization framework that aligns motions with a diverse range of control sequences without requiring task-specific training or algorithm redesign.
(2) We demonstrate that our method achieves high-quality alignments between a given motion and control sequence within seconds, eliminating the need for large datasets, extensive annotations, or prolonged training times.
(3) We present novel motion editing techniques—waveform-to-motion, sketch-to-motion, and motion-by-numbers—enabled by MAMM, which, to our knowledge, are the first of their kind.
(4) We demonstrate the generalizability of the MAMM framework to established tasks, such as motion-to-motion and audio-to-motion alignment.

2 RELATED WORKS

2.1 Motion Editing

Motion editing techniques aim to modify existing motion data or create new motion that satisfies new spatiotemporal constraints, adheres to specific stylistic requirements, or adapts to different environmental contexts. Early foundational approaches introduced the concept of *spacetime constraints*, formulating motion editing as an optimization problem guided by physical laws [Gleicher 1997; Witkin and Kass 1988]. This line of research enabled animators to enforce physically plausible edits on motion sequences while maintaining user control over keyframed poses.

Subsequent works focused on lowering the barrier for nonexperts and improving workflow efficiency through sketch-based interfaces and other intuitive user controls [Choi et al. 2016; Garcia et al. 2019; Guay et al. 2015; Terra and Metoyer 2004; Thorne et al. 2004]. These methods allowed users to apply edits by sketching curves or gestures, which were then mapped to character poses and motion adjustments. Another approach involved creating poses by interpolating keyframes spatially [Igarashi et al. 2005]. Parallel efforts explored physically guided animation tools, where motion was fine-tuned by automatically inferring or imposing physical properties, resulting in more believable outputs [Hahn et al. 2012; Shapiro and Lee 2011].

More recent solutions integrate optimization frameworks directly into keyframe-based workflows, enabling artists to selectively refine motion segments without fully discarding manual control [Ciccone et al. 2019; Koyama and Goto 2018]. In addition, tangible or actuated puppets provide haptic feedback and a physical interface, further bridging digital and real-world motion editing [Jacobson et al. 2014; Yoshizaki et al. 2011]. User performance itself has also served as an intuitive control mechanism [Dontcheva et al. 2003; Lockwood and Singh 2012].

Compared to existing methods, a notable feature of our method is automatically computing the control-to-motion mapping from input sequences in a unified manner. This allows various input methods for motion control. By contrast, existing methods require developers to explicitly define the mapping from control signals to motion.

2.2 Motion Retargeting and Alignment

Motion retargeting transfers motion data from a source character to a target character, even when their skeletal structures or proportions differ. Early methods formulated this as an optimization problem, incorporating constraints such as low-level motion representations [Gleicher 1998], kinematic constraints [Lee and Shin 1999], end-effector trajectories [Choi and Ko 2000], or dynamic feasibility [Tak and Ko 2005]. However, these approaches struggled with significant skeletal differences. Later approaches introduced partial correspondences [Rhodin et al. 2015; Seol et al. 2013; Yamane et al. 2010] or learned mappings independent of skeletal structures [Rhodin et al. 2014] to handle differing topologies.

The advent of deep learning and large motion capture datasets led to data-driven methods [Aberman et al. 2020; Li et al. 2024a, 2023b; Lim et al. 2019; Villegas et al. 2018]. Early learning-based techniques leveraged cycle-consistency and adversarial losses [Villegas et al. 2018] or skeleton-aware components [Aberman et al. 2020]. ACE [Li et al. 2023b] advanced retargeting for characters with significant skeletal differences using pretrained motion priors. Learning correspondences between general dynamical systems has also been investigated [Kim et al. 2020].

WalkTheDog [Li et al. 2024a] introduced motion alignment via a learned discretized phase manifold, enabling shared embeddings for characters with different topologies, such as humans and dogs. Their method uses frequency-scaled motion matching but requires extensive training and large datasets.

Our method simplifies this process, offering efficient motion alignment for small data pairs without requiring extensive training or large datasets, offering a practical solution for limited-data scenarios.

2.3 Audio-Driven Motion Synthesis

Audio-driven motion generation (e.g., dance, gesture synthesis) generates character motion from acoustic features, a long-standing topic in computer animation. For a comprehensive review, see [Nyatsanga et al. 2023; Zhu et al. 2024].

Early studies on dance synthesis primarily focused on synchronizing dance motions with musical beats, rhythms, and intensity using hand-crafted mappings [Ofli et al. 2008; Shiratori and Ikeuchi 2008; Shiratori et al. 2006] or employing example-based approaches and statistical models [Fan et al. 2012; Fukayama and Goto 2015; Ofli et al. 2012]. Recent advances in deep learning have enabled the generation of more expressive dance motions by leveraging sophisticated motion generation models [Bhattacharya et al. 2024; Lee et al. 2019; Li et al. 2021; Siyao et al. 2022], including diffusion models [Alexanderson et al. 2023; Li et al. 2024b; Tseng et al. 2023]. Other approaches combine learned choreo-musical embeddings with motion graphs [Au et al. 2022; Chen et al. 2021].

Parallel efforts in speech-driven gesture synthesis have similarly aimed to produce natural co-speech motion. Earlier approaches relied on extensive manual rules [Cassell et al. 2001; Kopp et al. 2003; Lee and Marsella 2006; Lhommet et al. 2015], while more recent methods use learning-based methods [Alexanderson et al. 2020; Ferstl and McDonnell 2018; Hasegawa et al. 2018; Takeuchi et al. 2017; Wu et al. 2021; Yazdian et al. 2021]. However, many of these methods depend heavily on large paired audio-motion datasets [Alexanderson et al. 2023; Bhattacharya et al. 2024; Chen et al. 2021; Fan et al. 2012; Lee et al. 2019; Li et al. 2021; Siyao et al. 2022] and are computationally demanding [Alexanderson et al. 2023; Li et al. 2024b; Tseng et al. 2023], limiting their adaptability to new characters or resource-limited environments. Our approach bypasses these challenges by employing fused semi-unbalanced Gromov-Wasserstein optimal transport, which reduces reliance on large datasets and hand-crafted features. This enables more efficient and flexible audio-to-motion synthesis suitable for a wide range of contexts.

2.4 Motion Synthesis from Small Data

Building on recent advances in image generation from a single or a few examples [Granot et al. 2022; Kulikov et al. 2023; Shaham et al. 2019], several studies have explored synthesizing motion using only a limited number of example sequences. For instance, GANimator [Li et al. 2022] employs generative adversarial networks to generate diverse motion patterns from a single instance. GenMM [Li et al. 2023a], in contrast, bypasses the training phase entirely, enabling rapid motion synthesis through a bidirectional similarity-based matching framework. SinDDM [Raab et al. 2023] further improves the quality of generated motion using diffusion models.

While these approaches demonstrate significant flexibility in conditioned motion synthesis tasks – such as inbetweening, trajectory control [Li et al. 2022, 2023a], and style transfer within the same skeletal structure [Raab et al. 2023] – they often lack inherent mechanisms to handle more complex motion control by sequences. Consequently, achieving desirable outcomes in such scenarios frequently requires additional annotations or manual interventions, which can increase both labor and computational costs.

2.5 Optimal Transport

Optimal transport (OT) has been widely applied across various domains, including machine learning, computer vision, and neuroscience [Montesuma et al. 2024]. For instance, Elnekave and Weiss [Elnekave and Weiss 2022] introduced methods for generating natural images by matching patch distributions. Advancements in Gromov-Wasserstein (GW) distance have enabled mappings between disparate domains, such as shape correspondence and image-to-color distribution matching [Solomon et al. 2016]. Recent developments in Unbalanced and Fused GW [Chizat et al. 2018; Sejourne et al. 2021] have further expanded OT applications to areas such as aligning individual brains [Thual et al. 2022] and unsupervised action segmentation [Xu and Gould 2024]. We incorporate the fused unbalanced GW OT problem solver [Xu and Gould 2024] as a core component of our framework, offering a unified and effective solution to sequence-conditioned motion control.

3 METHOD

3.1 Motion Representation

We represent a motion sequence as a series of time-indexed poses, as in a previous study [Li et al. 2023a], where each pose is described by *root displacement* V and *joint rotation* R in character space. At time t, the pose is given by $[V_t, R_t]$, where $V \in \mathbb{R}^{T\times 3}$ represents the changes in the root joint position compared to consecutive frames, and $R \in \mathbb{R}^{T\times J\times 3\times 3}$ describes the rotation of each joint in the character's root space in matrix representation. Here, J denotes the number of joints.

3.2 Metric-Aligning Motion Matching

Our framework, named *Metric-Aligning Motion Matching (MAMM)*, integrates *motion patch extraction and blending*, *FSUGW optimization*, which involves the minimization of *Gromov-Wasserstein (GW) loss* and *Wasserstein loss*, into a unified scheme.

We denote the original motion sequence as X and the control (input) sequence as Y, with X' representing the aligned motion sequence (output). The aligned sequence X' is optimized to retain similarity to the original sequence X while aligning with the control sequence Y. Both X' and X reside in the same domain, whereas X' and Y often belong to different domains, making it challenging to directly measure the distance between them (e.g., motion and music). Additionally, X and Y may have different lengths. Ultimately, X' and Y are optimized to have the same length and exhibit structural similarity.

Conceptually, the objective of the MAMM framework is to output an aligned motion such that the pairwise distance matrix of its frames closely resembles the distance matrix of the control sequence while retaining similarity to the original motion (see Fig. 2). As shown in the figure, the distance matrix of our aligned motion resembles that of the control sequence.

In the actual algorithm, we define the problem as the minimization of the fused semi-unbalanced Gromov-Wasserstein (FSUGW) loss L_{FSUGW}. Our formulation of L_{FSUGW}, as presented in Equations (1), (2), and (3), along with the projected mirror-descent algorithm employed as a component of our framework, follows the formulations in previous work [Xu and Gould 2024]. By minimizing this

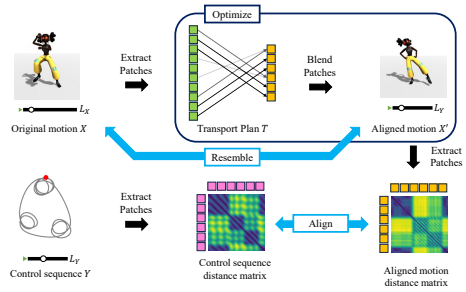


Fig. 2. Intuitive concept of MAMM framework. Our framework optimizes transport plan T and aligned motion sequence X', whose patch pairwise distance matrix is similar to that of control sequence Y, while resembling original motion X.

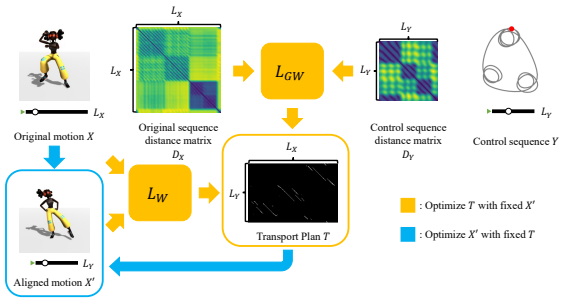


Fig. 3. Explanation of the fused semi-unbalanced Gromov-Wasserstein (FSUGW) objective and algorithm to minimize it. L_W constrains X' to resemble X via transport plan T and L_{GW} encourages T to be metric-aligning, which leads to structural similarity between X' and Y. We optimize FSUGW objective with alternating steps, where the first step optimizes T with X' fixed, and second step optimizes X' over fixed T.

objective via MAMM, X' is optimized as described. Our contribution lies in integrating coarse-to-fine optimization (Sec. 3.2.5) and the patch extraction and blending process into the optimization loop to ensure temporal consistency (Sec. 3.2.4). Fig. 3 provides an overview of the framework.

3.2.1 Motion Patch Extraction. To enable fine-grained optimization, we follow previous patch-based generation methods [Elnekave and Weiss 2022; Li et al. 2023a] by splitting the control sequence Y, original motion X, and aligned motion X' into overlapping temporal patches with a stride of one frame. These patches are labeled as $\tilde{Y}$, $\tilde{X}$, and $\tilde{X'}$, respectively. Note that $\tilde{Y}$ and $\tilde{X}$ remain fixed, while $\tilde{X'}$ is updated at every step of the optimization loop ($\tilde{X'} \leftarrow \text{ExtractPatches}(X')$). Measuring the FSUGW loss at the patch level ensures the capture of local temporal structures and ensures continuity.

Hereafter, we denote the length of the patch sequences as $L_X = |\tilde{X}|$ and $L_Y = |\tilde{Y}| = |\tilde{X'}|$.

3.2.2 Wasserstein Loss. The Wasserstein loss focuses on establishing direct correspondence between the two sets, X' and X. It is defined as follows:

$$L_W(\tilde{X'}, \tilde{X}, T) = \sum_{x'_i \in \tilde{X'}, x_k \in \tilde{X}} d_X(x'_i, x_k) T_{i,k}. \tag{1}$$

where $T \in \{T \in \mathbb{R}^{L_Y \times L_X} | T \geq 0, T\mathbf{1} = a, a = \mathbf{1}/L_Y\}$ denotes the transport plan matrix (coupling matrix) between $\tilde{X'}$ and $\tilde{X}$, and d_X is a normalized distance function in the domain of $\tilde{X}$ and $\tilde{X'}$. We use the L2-norm as the distance function, normalizing it by either the maximum or mean distance, depending on the task. By minimizing $\min_T L_W$ w.r.t X', we encourage each patch in X' to move closer to its corresponding patch in X via the coupling T, resulting in the aligned motion resembling the original motion.

3.2.3 Gromov-Wasserstein Loss. The Gromov-Wasserstein (GW) loss evaluates the alignment of internal structures between two datasets, specifically the control Y and the original X. The loss, $L_{GW}(\tilde{Y}, \tilde{X}, T)$, is expressed as follows:

$$L_{GW}(\tilde{Y}, \tilde{X}, T) = \sum_{\substack{y_i, y_j \in \tilde{Y} \\ x_k, x_l \in \tilde{X}}} |d_Y(y_i, y_j) - d_X(x_k, x_l)|^2 T_{i,k} T_{j,l}. \tag{2}$$

where d_Y is the normalized distance function in the domain of Y. It is important to note that we do not use the distance between $x \in \tilde{X}$ and $y \in \tilde{Y}$ at all.

Minimizing L_{GW} encourages the transport plan T to be metric-aligning, ultimately promoting structural alignment between X' and Y when combined with the Wasserstein loss described in Sec. 3.2.2.

3.2.4 Fused Semi-Unbalanced Gromov-Wasserstein (FSUGW) Block. We combine the Wasserstein loss and GW loss to define and optimize L_{FSUGW} as an instance of the entropic fused semi-unbalanced Gromov-Wasserstein problem [Xu and Gould 2024]:

$$\begin{aligned} X'^*, T^* &= \underset{X', T}{\text{argmin}} \quad L_{\text{FSUGW}}(\tilde{X'}, \tilde{X}, \tilde{Y}, T) \\ \text{where} \quad L_{\text{FSUGW}}(\tilde{X'}, \tilde{X}, \tilde{Y}, T) &= \alpha \cdot L_{\text{GW}}(\tilde{Y}, \tilde{X}, T) \\ &\quad + (1-\alpha) \cdot L_{\text{W}}(\tilde{X'}, \tilde{X}, T) \\ &\quad + \lambda \cdot D_{\text{KL}}(T^\top \mathbf{1} \,\|\, \boldsymbol{b}) - \epsilon \cdot H(T), \\ \text{subject to} \quad T\mathbf{1} &= \boldsymbol{a}, \quad T \geq 0. \end{aligned} \tag{3}$$

Here, we define $a = \mathbf{1}_{L_Y}/L_Y$, $b = \mathbf{1}_{L_X}/L_X$, where L_Y and L_X are the number of data (frames) in $\tilde{Y}$ (or $\tilde{X'}$) and $\tilde{X}$, respectively. The term $H(T) = -\sum_{i,j} T_{i,j} log(T_{i,j})$ represents the soft entropy constraint, which facilitates optimization via the projected mirror-descent algorithm [Solomon et al. 2016; Xu and Gould 2024]. The soft marginal constraint, defined as the D_{KL} term, encourages bi-directional fidelity of the distribution of the aligned motion X' to that of X. While soft constraints are imposed on one side, we enforce hard constraints on the other side of the marginals, $T\mathbf{1} = a$, ensuring that T satisfies the specified marginals. Intuitively, this constraint ensures that every part of the aligned motion X' corresponds to some part of the original motion X. The effects of the parameters α and λ will be discussed in Sec. 5.1.

Optimizing L_{FSUGW} w.r.t T and X' is similar to the (fused unbalanced) Gromov-Wasserstein Barycenter problem [Peyré et al. 2016; Thual et al. 2022], which is typically solved using a block coordinate

descent algorithm. Inspired by this, our proposed FSUGW block optimizes L_{FSUGW} by alternating the following steps:

(1) **Extract patches:** Update the patches $\tilde{X}'$ using the current value of X',
$$\tilde{X}' \leftarrow \text{ExtractPatches}(X')$$
(2) **Optimize T with fixed X':** Given X', we find T as a local minimum of L_{FSUGW} by applying the projected mirror-descent algorithm, identical to the version in [Xu and Gould 2024] except for the parameters. At each stage, T is initialized as $\mathbf{a}\mathbf{b}^T$, following previous practices.
(3) **Optimize X' with fixed T:** With T fixed, update X' by matching weighted by the transport plan T and blending the overlapping regions through averaging.
$$X' \leftarrow \text{BlendPatches}(T \cdot \tilde{X} \cdot L_Y).$$

The patch extraction and blending process in the optimization loop ensures that spatio-temporally adjacent patches in X are encouraged to remain adjacent in X'. If X' is formed by blending spatio-temporally distant patches, it may result in highly unnatural poses in the overlapping regions of those patches. However, such cases are naturally excluded during optimization, as they increase L_W, which the algorithm minimizes.

In practice, this algorithm typically converges after several iterations to produce visually appealing results across various applications, as demonstrated in Sec. 4. For these steps, we set $M = 20$ as the number of iterations.

3.2.5 Coarse-to-Fine Motion Alignment. Owing to the non-convex nature of GW problems [Solomon et al. 2016], proper initialization of the FSUGW block is crucial for guiding the optimization process toward high-quality aligned motion. To ensure effective initialization, we adopt a coarse-to-fine optimization strategy, as used in previous works [Elnekave and Weiss 2022; Li et al. 2022, 2023a]. This approach begins at a coarse level, with the initial transport plan T computed by minimizing only the L_{GW} term, and progressively refines the motion through upsampling until reaching the final stage:

(1) **Upsampling**: Upsample the output X'^{k-1} from the previous stage to serve as the initial guess for stage k.
(2) **FSUGW Optimization**: Apply the FSUGW block using Y^k, X^k to refine X'^k.

After the final stage $k = N$, X'^N achieves a high-quality alignment with the control sequence Y, while preserving the content of the original motion X. In our examples, the coarsest level is set to be 4 times shorter than the final length, and $N = 6$ stages are used to progressively upsample the sequence. The pseudocode for the entire MAMM algorithm is provided in Alg. 1.

ALGORITHM 1: MAMM (Metric-aligning motion matching)

Input: control sequence Y, original motion X, parameters $\alpha, \lambda, \epsilon$, number of stages N, number of iterations M
Output: aligned motion X'
// Coarse-to-fine Motion Alignement
for $k \leftarrow 0$ **to** N **do**
  $X' \leftarrow (k = 0)\,?\,\text{Initialize}() : \text{Upsample}(X')$;
  $X^k, Y^k \leftarrow \text{Downsample}(X, Y, k)$;
  // FSUGW Block
  for $m \leftarrow 0$ **to** M **do**
    $(\tilde{X}', \tilde{Y}, \tilde{X}) \leftarrow \text{ExtractPatches}(X', Y^k, X^k)$;
    $a \leftarrow \mathbf{1}/L_Y, \quad b \leftarrow \mathbf{1}/L_X, \quad T \leftarrow (m = 0)\,?\,\mathbf{a}\mathbf{b}^T : T$;
    // Optimize T via projected mirror descent
    $T \leftarrow \text{SolveFSUGW}(\tilde{X}', \tilde{X}, \tilde{Y}, T, a, b, \alpha, \lambda, \epsilon)$;
    // Optimize X' via matching
    $X' \leftarrow \text{BlendPatches}(T \cdot \tilde{X} \cdot L_Y)$;
  end
end
return X';

3.3 Handling Additional Space-Time Constraints

3.3.1 Soft Keyframes by Example Pairs. We introduce the concept of "soft keyframes" by adding *example patch pairs* into the FSUGW optimization. These example patch pairs represent sample-to-sample correspondences between the domains of $\tilde{Y}$ and $\tilde{X}$. The patch samples do not necessarily have to be part of the original or control sequence; the only requirement is that they belong to the same domain. We incorporate these example patch pairs into the optimization by applying the following substitution:

$$\tilde{X} \leftarrow \left[\tilde{X},\, X_{\text{example}}\right], \tilde{X}' \leftarrow \left[\tilde{X}', X_{\text{example}}\right], \tilde{Y} \leftarrow \left[\tilde{Y},\, Y_{\text{example}}\right],$$
$$a \leftarrow \left[a,\, \tfrac{w_{\text{example}} \cdot \mathbf{1}}{L_{\text{example}}}\right], b \leftarrow \left[b,\, \tfrac{w_{\text{example}} \cdot \mathbf{1}}{L_{\text{example}}}\right],$$

where X_{example} and Y_{example} denote the example patch pairs, w_{example} denotes the weight of the soft keyframe constraints, and L_{example} denotes the number of example patches. Additionally, we constrain the transport matrix T such that all masses assigned to the example patches in $\tilde{X}$ (or $\tilde{X}'$) are transported exclusively to their corresponding example patches, and vice versa. This ensures that these soft keyframes do not interfere with the overall distribution derived from the original motion X, which is reflected in the aligned motion X' through the soft marginal constraint $D_{\text{KL}}\left(T^\top \mathbf{1} \,\|\, \mathbf{b}\right)$. The difference between these soft keyframes and traditional (hard) keyframes, along with illustrative examples, is discussed in Sec. 5.

3.3.2 Other Constraints. Because a portion of our method relies on patch extraction and blending, it is compatible with various space-time control methods. In our implementation, we integrate *hard keyframing* and *infinite loop* constraints, as outlined in GenMM [Li et al. 2023a].

Hard Keyframes. Hard keyframes are specified pairs of patches in X and Y that must be coupled. When imposing a hard keyframe constraint, we fix the specified segments of both X' and T at every step of the FSUGW optimization process, including during the projected mirror-descent routine. This ensures that the predetermined keyframes remain unchanged throughout the alignment process.

Infinite Looping. When imposing an infinite loop constraint, the last patch of the aligned motion sequence is enforced to match the first patch of the same sequence, ensuring a seamless loop. Following the practice in GenMM [Li et al. 2023a], we achieve this by blending the overlapping region between the end and start patches consistently.

Alignment results with these constraints can be observed in the supplemental video along with the demonstration of our sketch-to-motion applications.

4 APPLICATIONS

In this section, we demonstrate how the MAMM framework can be applied to various input control sequences. All use cases are realized by splitting each control sequence into patches, computing the distance matrix among the patches, and adjusting the optimization parameters; the underlying algorithm itself remains unchanged. All data used in the examples are sourced from Mixamo [Adobe 2025] or Truebones [Truebones 2025], which contain various motions and skeletons of humanoids and non-humanoids, respectively. The sequence lengths in our experiments range from approximately 50 to 600 frames at 30 fps. The patch size of the original motion was set to 11 frames, with a stride of one frame. Patch extraction from the control sequence used the same temporal duration and interval. Additionally, we used the AIST dance database [Tsuchida et al. 2019] for music data and the BEAT dataset [Liu et al. 2022] for speech data.

We implemented our applications on a personal computer equipped with an NVIDIA GeForce RTX 4080 GPU. Each generation typically finishes within 10 s but may take longer depending on the sequence length. The parameter values used in our applications are detailed in the supplemental materials. We also include a supplemental video to facilitate the visual evaluation of the alignment results.

4.1 Sketch-to-Motion (2D Curve-to-Motion)

In this application, the control sequence is a temporal series of two-dimensional data points provided by the user. To enhance the user experience, we have developed a dedicated user interface (UI) for sketch-based motion control. In this UI, the user draws a 2D curve on the left-hand canvas and observes the aligned animation on the right. The user can iteratively refine the animation by redrawing the curve, placing soft or hard keyframes on the canvas, or enabling infinite looping. The 2D points along the curve are resampled at constant time intervals as patches and then fed into the alignment algorithm. To reduce the impact of unintended noise in the input, we apply Gaussian smoothing before processing the curve in our framework.

The user does not need prior knowledge of which part of the canvas corresponds to specific frames of the animation. As such, the exact location of the curve does not matter. Instead, the system analyzes the overall shape and movement of the curve and maps it to the general structure of the motion. For example, a large circle is automatically matched with a large cyclic motion, while a small circle corresponds to a smaller motion. Note that we compute a mapping from a sampled point on the curve to a motion patch. This is not a continuous mapping from an arbitrary spatial position to a pose. Therefore, even if the curve passes through the same position (i.e., a crossing), the pose at that location on the curve can differ from the pose at the same spatial location at a different point in time. This implicit matching is effective in scenarios where exact poses are less critical and dynamics are more important. If precise mapping is needed, the user can specify it by adding soft or hard constraints. An example result is shown in Fig. 4 and Fig. 5. Additionally, we performed a user study to gather feedback on this tool, which is discussed in Sec. 6.

4.2 1D Waveform-to-Motion

Here, the control sequence is a temporal series of one-dimensional scalar values. We demonstrate how periodic motion patterns can be controlled using time-varying periodic waveforms. A low-frequency sine wave generates slow periodic motion, while a high-frequency sine wave produces fast periodic motion. Sharp corners in the waveform result in sharp turns in the motion. Additionally, we use a sine wave with an alternating center as input to achieve complex periodicity control. The results are provided in Fig. 6 and the supplemental videos.

4.3 Motion-by-Numbers

Drawing inspiration from successful image synthesis pipelines guided by segmentation labels [Isola et al. 2017], we align motion to a user-specified temporal sequence of segmentation labels, each represented as a one-hot class vector. With MAMM, the original motion sequence does not require any precomputed or annotated segmentation labels; instead, the algorithm automatically aligns each segment according to the input labels. The user does not need prior knowledge of which label corresponds to which motion. However, when necessary, the user can explicitly specify the mapping by providing soft or hard constraints. Results are shown in the center row of Fig. 9 and Fig. 10, as well as in the supplemental video.

4.4 Motion Control by Audio

As mentioned in Sec. 2.3, audio-driven motion control has been extensively studied in the motion generation community. We focus on a scenario where the available example sequences are relatively short (3–20 s in duration) and lack paired audio or auxiliary annotations. The audio input is represented as 40-dimensional MFCC features, extracted using the Librosa library [McFee et al. 2024]. Both speech and music data are used as audio controls, while short segments of gesture or dance motion serve as the original motion. As presented in Fig. 7, our method naturally synchronizes the resulting motion with style, beat, or volume changes in the audio, even when the original motion style differs from that in the audio. Notably, this approach does not require any training or specialized, hand-tuned optimization.

4.5 Motion-to-Motion Alignment

We applied MAMM to cross-skeletal motion-to-motion alignment, where one motion sequence is matched to another with a different skeletal structure [Li et al. 2024a]. An example is shown in Fig. 8. Leveraging its metric-alignment principle, our method effectively handles both periodic motions, such as walking, and non-periodic activities, such as combat moves, without requiring additional training or customized feature engineering.

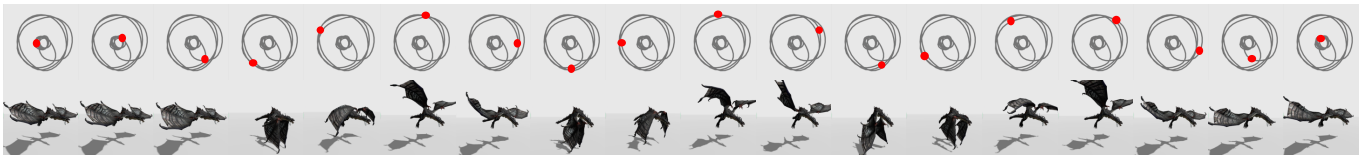


Fig. 4. Demonstration of character motion control using sketched curve. Character's movement follows abstract structure of the curve. Additional examples are available in supplementary video.

5 DISCUSSIONS

5.1 Parameter Effects

We explain the effect of the main parameters in our framework (α and λ). See Fig. 9, Fig. 10 or supplemental materials to see examples.

The parameter α balances the Wasserstein loss L_W and the Gromov-Wasserstein loss L_{GW}. The L_{GW} term maintains structural similarity between the control sequence and the aligned motion, while L_W enforces local, patchwise fidelity to the original motion. Increasing α amplifies the influence of the control input but may lead to unnatural outputs if set too high.

The parameter λ scales the soft marginal regularization term $D_{\mathrm{KL}}(T^\top \mathbf{1} \,\|\, \mathbf{b})$. Intuitively, a larger λ forces the aligned motion to more closely maintain the overall distribution of the original motion, potentially diminishing the influence of the control input if λ is set too high.

We recommend selecting relatively large values for α and small values for λ, such as $\alpha = 0.8$ and $\lambda = 0.05$. After evaluating the initial results, these parameters can be fine-tuned by decreasing α or increasing λ to enhance naturalness, or vice versa, depending on the desired outcome. Additionally, we have observed that setting $\epsilon = 1$ works well for most settings; however, adjustments may be needed based on the size or complexity of the data.

5.2 Coarse-to-Fine Motion Alignment

While adopting a coarse-to-fine optimization approach generally improves results, it becomes especially important when the distance matrix of the control sequence is more complex, such as in motion-to-motion alignment scenarios. In the supplementary video, we highlight failure cases that arise when using a one-stage (non-coarse-to-fine) optimization method.

5.3 Soft Keyframes and Hard Keyframes

As discussed in Sec. 3, our MAMM framework accommodates both *soft keyframing* and *hard keyframing*. In this section, we use the sketch-to-motion application to illustrate these keyframe concepts in a practical context.

Soft keyframes are defined as pairs of patch samples drawn from the control domain and the original motion domain. By contrast, hard keyframes pair a specific point in time (on the temporal axis) with a patch in the original motion domain. Consequently, in the *sketch-to-motion* setting, soft keyframes can be placed anywhere on the canvas, whereas hard keyframes must be associated with a specific time frame and positioned directly on the curve.

This distinction enables an interesting application of soft keyframes: they can act as "negative keyframes". As illustrated in Fig. 5, if the user's drawn curve is positioned far from keyframes representing the "hands-up horse rider" and "hands-down horse rider" poses, the algorithm instead generates a looped "side step" sequence that stays distant from both keyframe poses.

6 USER FEEDBACK

We invited five participants to test the sketch-to-motion application to obtain feedback. The participants were computer science students with no prior experience in character animation editing. During the user study session, we provided a 10-minute tutorial followed by 15 minutes of guided practice. Examples of sketches created by participants are shown in Fig. 11. Finally, the participants were tasked with arranging an animation to match a given video. All participants successfully recreated a motion sequence similar to that shown in the video.

We generally received positive feedback, such as "The application can be used even without experience in animation editing." and "I found changing animations with keyframes to be very convenient." However, some participants noted difficulties, stating "I found the relationship between sketches and animations to be a bit unclear." They also suggested improvements, including "Suggesting keyframe candidates could reduce editing time," and "A feature to completely remove unwanted keyframes (hard negative keyframes) would be useful." The animations created by the participants and detailed survey results are included in the supplementary materials.

7 LIMITATIONS AND FUTURE WORKS

Real-time control. In our formulation, the user needed to provide a complete control sequence before computing the alignment, making real-time interaction impossible. Future work could focus on developing fast approximation techniques for MAMM to enable real-time user control.

Ambiguity to Rigid Transformation and Symmetry. Since the L_{GW} term relied on a metric, it was invariant to rigid transformations and symmetries in the patch space. For example, the method could not distinguish between stepping to the right and stepping to the left in symmetrical step motions, which could result in outputs differing from users' expectations. However, tools such as keyframes were provided to address and control this issue.

Pairwise Distance Scaling. A key challenge lay in optimally designing and normalizing the distance functions, denoted by d_X and d_Y. Although these distance functions were normalized using the max or mean value in our experiments, we also attempted to learn a scalar scaling factor s for $d_{\tilde{Y}}$ within the optimization loop by setting its value to the global minimum of L_{GW} with T fixed. However,

this approach did not yield consistent improvements. Future work should investigate more robust matrix-scaling strategies, particularly for larger datasets that exhibit multiple clusters.

Scalability Concerns. MAMM's computational complexity increased both spatially and temporally with larger input sequences. The largest example of T we tested involved motion with 594 frames (19.8 s) and control audio with 1441 MFCC features (48.04 s), which took 7 s to compute. However, as modern motion datasets sometimes contain over 100 thousands of frames, storing and operating on such high-dimensional distance matrices could become prohibitive. Developing more scalable approaches—such as improved hierarchical techniques or approximations to FSUGW—constitutes an important direction for future research.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their insightful feedback. We would like to thank Takaaki Shiratori and Peizhuo Li for the valuable feedback and helpful discussions on this project. This work was supported by Japan Science and Technology Agency (JST) as part of Adopting Sustainable Partnerships for Innovative Research Ecosystem (ASPIRE), Grant Number JPMJAP2401.

REFERENCES

Kfir Aberman, Peizhuo Li, Dani Lischinski, Olga Sorkine-Hornung, Daniel Cohen-Or, and Baoquan Chen. 2020. Skeleton-aware networks for deep motion retargeting. *ACM Trans. Graph.* 39, 4 (Aug. 2020), 62:62:1–62:62:14. https://doi.org/10.1145/3386569.3392462

Adobe. 2025. Mixamo. https://www.mixamo.com/#/

Simon Alexanderson, Gustav Eje Henter, Taras Kucherenko, and Jonas Beskow. 2020. Style-Controllable Speech-Driven Gesture Synthesis Using Normalising Flows. (2020). https://doi.org/10.1111/cgf.13946 Publisher: The Eurographics Association and John Wiley & Sons Ltd..

Simon Alexanderson, Rajmund Nagy, Jonas Beskow, and Gustav Eje Henter. 2023. Listen, Denoise, Action! Audio-Driven Motion Synthesis with Diffusion Models. *ACM Trans. Graph.* 42, 4 (July 2023), 44:1–44:20. https://doi.org/10.1145/3592458

Ho Yin Au, Jie Chen, Junkun Jiang, and Yike Guo. 2022. ChoreoGraph: Music-conditioned Automatic Dance Choreography over a Style and Tempo Consistent Dynamic Graph. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22)*. Association for Computing Machinery, New York, NY, USA, 3917–3925. https://doi.org/10.1145/3503161.3547797

Aneesh Bhattacharya, Manas Paranjape, Uttaran Bhattacharya, and Aniket Bera. 2024. DanceAnyWay: Synthesizing Beat-Guided 3D Dances with Randomized Temporal Contrastive Learning. *Proceedings of the AAAI Conference on Artificial Intelligence* 38, 2 (March 2024), 783–791. https://doi.org/10.1609/aaai.v38i2.27836 Number: 2.

Justine Cassell, Hannes Högni Vilhjálmsson, and Timothy Bickmore. 2001. BEAT: the Behavior Expression Animation Toolkit. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques (SIGGRAPH '01)*. Association for Computing Machinery, New York, NY, USA, 477–486. https://doi.org/10.1145/383259.383315

Kang Chen, Zhipeng Tan, Jin Lei, Song-Hai Zhang, Yuan-Chen Guo, Weidong Zhang, and Shi-Min Hu. 2021. ChoreoMaster: choreography-oriented music-driven dance synthesis. *ACM Trans. Graph.* 40, 4 (July 2021), 145:1–145:13. https://doi.org/10.1145/3450626.3459932

Lénaïc Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. 2018. Scaling algorithms for unbalanced optimal transport problems. *Math. Comp.* 87, 314 (Nov. 2018), 2563–2609. https://doi.org/10.1090/mcom/3303

Byungkuk Choi, Roger Blanco i Ribera, J. P. Lewis, Yeongho Seol, Seokpyo Hong, Haegwang Eom, Sunjin Jung, and Junyong Noh. 2016. SketchiMo: sketch-based motion editing for articulated characters. *ACM Trans. Graph.* 35, 4 (July 2016), 146:1–146:12. https://doi.org/10.1145/2897824.2925970

Kwang-Jin Choi and Hyeong-Seok Ko. 2000. Online motion retargetting. *The Journal of Visualization and Computer Animation* 11, 5 (2000), 223–235. https://doi.org/10.1002/1099-1778(200012)11:5<223::AID-VIS236>3.0.CO;2-5 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/1099-1778%28200012%2911%3A5%3C223%3A%3AAID-VIS236%3E3.0.CO%3B2-5.

Loïc Ciccone, Cengiz Öztireli, and Robert W. Sumner. 2019. Tangent-space optimization for interactive animation control. *ACM Trans. Graph.* 38, 4 (July 2019), 101:1–101:10. https://doi.org/10.1145/3306346.3322938

Mira Dontcheva, Gary Yngve, and Zoran Popović. 2003. Layered acting for character animation. In *ACM SIGGRAPH 2003 Papers (SIGGRAPH '03)*. Association for Computing Machinery, New York, NY, USA, 409–416. https://doi.org/10.1145/1201775.882285

Ariel Elnekave and Yair Weiss. 2022. Generating Natural Images with Direct Patch Distributions Matching. In *Computer Vision – ECCV 2022*, Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (Eds.). Springer Nature Switzerland, Cham, 544–560. https://doi.org/10.1007/978-3-031-19790-1_33

Rukun Fan, Songhua Xu, and Weidong Geng. 2012. Example-based automatic music-driven conventional dance motion synthesis. *IEEE transactions on visualization and computer graphics* 18, 3 (March 2012), 501–515. https://doi.org/10.1109/TVCG.2011.73

Ylva Ferstl and Rachel McDonnell. 2018. Investigating the use of recurrent motion modelling for speech gesture generation. In *Proceedings of the 18th International Conference on Intelligent Virtual Agents (IVA '18)*. Association for Computing Machinery, New York, NY, USA, 93–98. https://doi.org/10.1145/3267851.3267898

Satoru Fukayama and Masataka Goto. 2015. Music content driven automated choreography with beat-wise motion connectivity constraints. *Proceedings of SMC* (2015), 177–183.

Maxime Garcia, Remi Ronfard, and Marie-Paule Cani. 2019. Spatial Motion Doodles: Sketching Animation in VR Using Hand Gestures and Laban Motion Analysis. In *Proceedings of the 12th ACM SIGGRAPH Conference on Motion, Interaction and Games (MIG '19)*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/3359566.3360061

Michael Gleicher. 1997. Motion editing with spacetime constraints. In *Proceedings of the 1997 symposium on Interactive 3D graphics (I3D '97)*. Association for Computing Machinery, New York, NY, USA, 139–ff. https://doi.org/10.1145/253284.253321

Michael Gleicher. 1998. Retargetting motion to new characters. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques (SIGGRAPH '98)*. Association for Computing Machinery, New York, NY, USA, 33–42. https://doi.org/10.1145/280814.280820

Niv Granot, Ben Feinstein, Assaf Shocher, Shai Bagon, and Michal Irani. 2022. Drop the GAN: In Defense of Patches Nearest Neighbors As Single Image Generative Models. 13460–13469. https://openaccess.thecvf.com/content/CVPR2022/html/Granot_Drop_the_GAN_In_Defense_of_Patches_Nearest_Neighbors_As_CVPR_2022_paper.html

Martin Guay, Rémi Ronfard, Michael Gleicher, and Marie-Paule Cani. 2015. Space-time sketching of character animation. *ACM Trans. Graph.* 34, 4 (July 2015), 118:1–118:10. https://doi.org/10.1145/2766893

Fabian Hahn, Sebastian Martin, Bernhard Thomaszewski, Robert Sumner, Stelian Coros, and Markus Gross. 2012. Rig-space physics. *ACM Trans. Graph.* 31, 4 (July 2012), 72:1–72:8. https://doi.org/10.1145/2185520.2185568

Dai Hasegawa, Naoshi Kaneko, Shinichi Shirakawa, Hiroshi Sakuta, and Kazuhiko Sumi. 2018. Evaluation of Speech-to-Gesture Generation Using Bi-Directional LSTM Network. In *Proceedings of the 18th International Conference on Intelligent Virtual Agents (IVA '18)*. Association for Computing Machinery, New York, NY, USA, 79–86. https://doi.org/10.1145/3267851.3267878

T. Igarashi, T. Moscovich, and J. F. Hughes. 2005. Spatial keyframing for performance-driven animation. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation (SCA '05)*. Association for Computing Machinery, New York, NY, USA, 107–115. https://doi.org/10.1145/1073368.1073383

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-To-Image Translation With Conditional Adversarial Networks. 1125–1134. https://openaccess.thecvf.com/content_cvpr_2017/html/Isola_Image-To-Image_Translation_With_CVPR_2017_paper.html

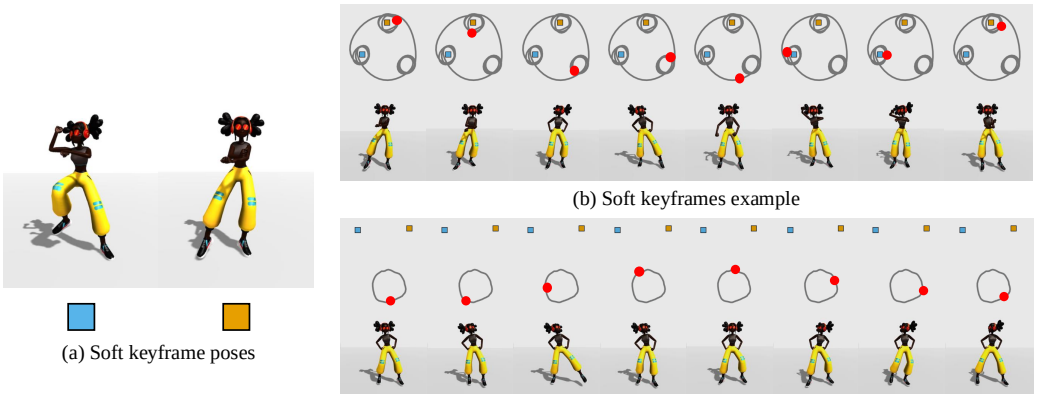
Alec Jacobson, Daniele Panozzo, Oliver Glauser, Cédric Pradalier, Otmar Hilliges, and Olga Sorkine-Hornung. 2014. Tangible and modular input device for character articulation. *ACM Trans. Graph.* 33, 4 (July 2014), 82:1–82:12. https://doi.org/10.1145/2601097.2601112

Nam Hee Kim, Zhaoming Xie, and Michiel Panne. 2020. Learning to Correspond Dynamical Systems. In *Proceedings of the 2nd Conference on Learning for Dynamics and Control*. PMLR, 105–117. https://proceedings.mlr.press/v120/kim20a.html ISSN: 2640-3498.

Stefan Kopp, Bernhard Jung, Nadine Pfeiffer-Leßmann, and Ipke Wachsmuth. 2003. Max - A Multimodal Assistant in Virtual Reality Construction. *Künstliche Intell.* 17 (2003), 11–. https://api.semanticscholar.org/CorpusID:14966714

Yuki Koyama and Masataka Goto. 2018. OptiMo: Optimization-Guided Motion Editing for Keyframe Character Animation. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3173574.3173735

Vladimir Kulikov, Shahar Yadin, Matan Kleiner, and Tomer Michaeli. 2023. SinDDM: A Single Image Denoising Diffusion Model. In *Proceedings of the 40th International Conference on Machine Learning*. PMLR, 17920–17930. https://proceedings.mlr.press/v202/kulikov23a.html ISSN: 2640-3498.



(a) Soft keyframe poses

(b) Soft keyframes example

(c) Soft keyframes as “negative keyframes” example

Fig. 5. Example of controlling aligned motion with soft keyframes. (a) Users can specify keyframe poses, such as "hands-up horse rider" and "hands-down horse rider," using our interface. For simplicity, users select poses from original sequence, although our method does not impose strict constraints on keyframe selection. (b) When motion curve approaches keyframe, algorithm ensures that corresponding poses in aligned motion closely match keyframe poses. (c) Conversely, when curve is distant from keyframe, algorithm selects alternative poses, such as "side-step" poses, which differ from keyframe poses. This illustrates how soft keyframes can influence motion in both positive and negative contexts.

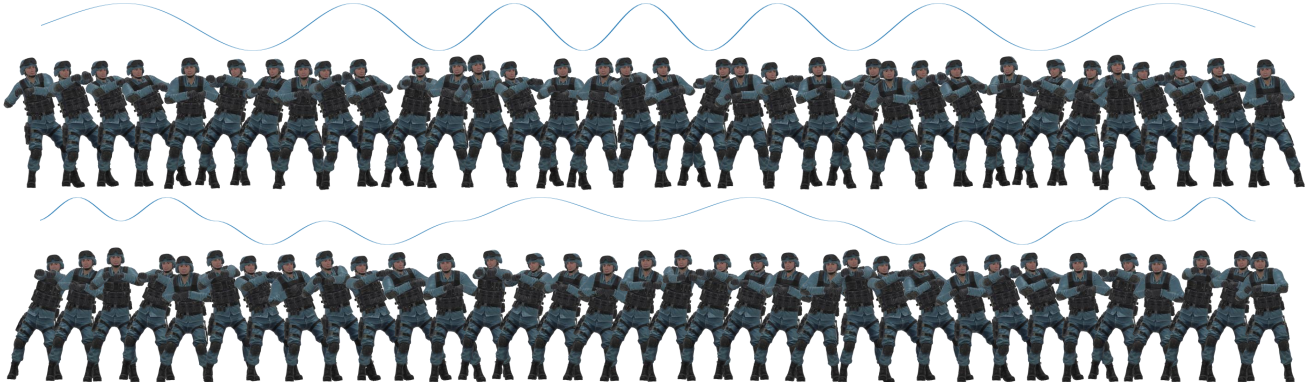
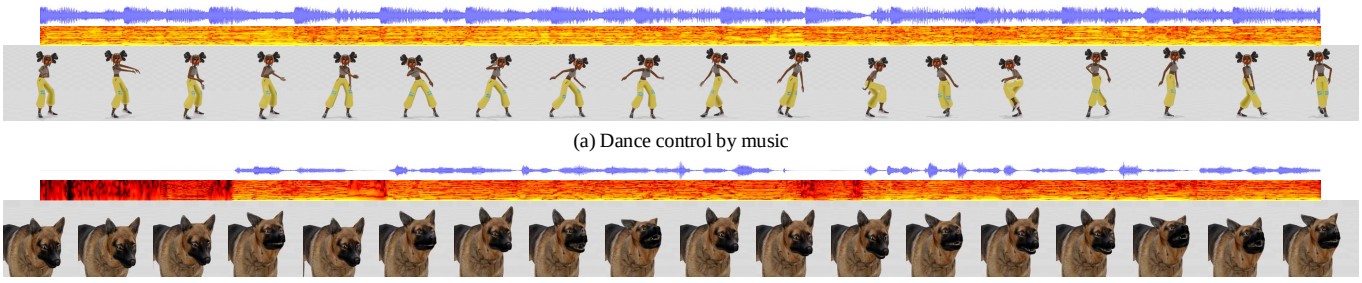


Fig. 6. Examples of controlling motion by waveforms. We used (1) frequency-varying sine wave and (2) center-alternating sine wave to control the dance motion’s frequency and phase. For more examples, please refer to the attached video.



(a) Dance control by music

(b) Gesture control by speech

Fig. 7. Examples of motion controlled by audio data. We tested two scenarios: (a) dance movements controlled by music and (b) gestures (barking) controlled by speech. Here audio data is represented in both waveforms and MFCCs, but we only used MFCCs as input. Synchronization between motion and intensity, beat, or style of audio was observed. For more details, please refer to supplemental video.

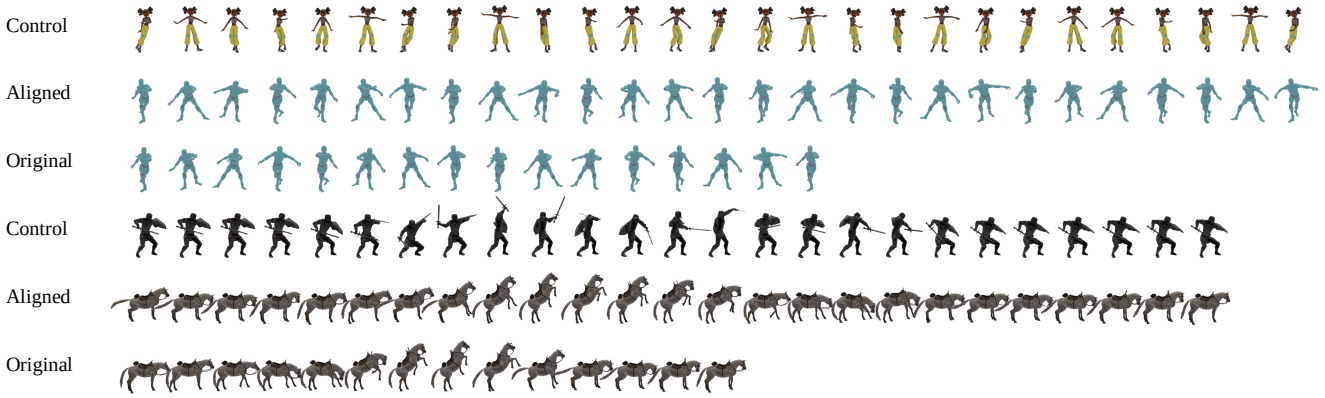


Fig. 8. Examples of motion-to-motion alignment, where original motion is synchronized with control motion. First example aligns stepping motion to step with different skeletal structure and style, matching both frequency and phase. Second example aligns nonperiodic combat sequence involving human and horse, demonstrating synchronized timing of combat actions.

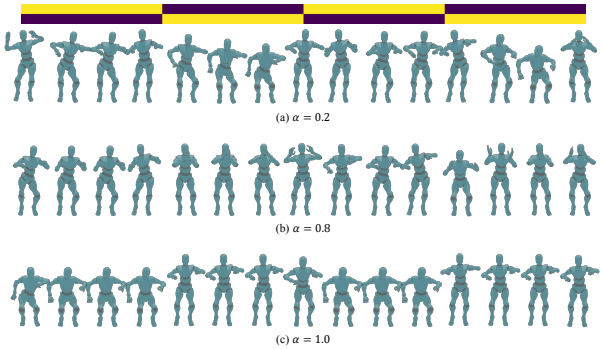


Fig. 9. Aligned motions for various α values with λ and ϵ fixed to 0.05 and 1.0, respectively. Small α (e.g., 0.2) results in natural motion but occasionally ignores control inputs, such as style changes within the same segmentation label. Conversely, α value of 1.0 leads to aligned motions with segments that exhibit no movement.

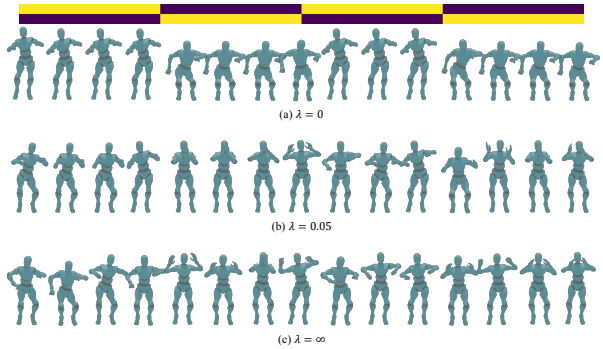


Fig. 10. Aligned motions for various λ values with α fixed at 0.8. When λ is small (e.g., 0), aligned motion includes segments where poses remain unchanged, deviating from the overall dynamics of the original motion. By contrast, large λ (e.g., ∞) ensures that distribution of aligned motion patches closely matches original motion, often at the expense of adhering to control sequence.

Hsin-Ying Lee, Xiaodong Yang, Ming-Yu Liu, Ting-Chun Wang, Yu-Ding Lu, Ming-Hsuan Yang, and Jan Kautz. 2019. Dancing to Music. In *Advances in Neural Information Processing Systems*, Vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2019/hash/7ca57a9f85a19a6e4b9a248c1daca185-Abstract.html

Jina Lee and Stacy Marsella. 2006. Nonverbal behavior generator for embodied conversational agents. In *Proceedings of the 6th international conference on Intelligent Virtual Agents (IVA'06)*. Springer-Verlag, Berlin, Heidelberg, 243–255. https://doi.org/10.1007/11821830_20

Jehee Lee and Sung Yong Shin. 1999. A hierarchical approach to interactive motion editing for human-like figures. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques (SIGGRAPH '99)*. ACM Press/Addison-Wesley Publishing Co., USA, 39–48. https://doi.org/10.1145/311535.311539

Sunmin Lee, Taeho Kang, Jungnam Park, Jehee Lee, and Jungdam Won. 2023. SAME: Skeleton-Agnostic Motion Embedding for Character Animation. In *SIGGRAPH Asia 2023 Conference Papers (SA '23)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3610548.3618206

Margot Lhommet, Yuyu Xu, and Stacy Marsella. 2015. Cerebella: Automatic Generation of Nonverbal Behavior for Virtual Humans. *Proceedings of the AAAI Conference on Artificial Intelligence* 29, 1 (March 2015). https://doi.org/10.1609/aaai.v29i1.9778 Number: 1.

Peizhuo Li, Kfir Aberman, Zihan Zhang, Rana Hanocka, and Olga Sorkine-Hornung. 2022. GANimator: neural motion synthesis from a single sequence. *ACM Trans. Graph.* 41, 4 (July 2022), 138:1–138:12. https://doi.org/10.1145/3528223.3530157

Peizhuo Li, Sebastian Starke, Yuting Ye, and Olga Sorkine-Hornung. 2024a. WalkTheDog: Cross-Morphology Motion Alignment via Phase Manifolds. In *ACM SIGGRAPH 2024 Conference Papers (SIGGRAPH '24)*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/3641519.3657508

Ruilong Li, Shan Yang, David A. Ross, and Angjoo Kanazawa. 2021. AI Choreographer: Music Conditioned 3D Dance Generation With AIST++. 13401–13412. https://openaccess.thecvf.com/content/ICCV2021/html/Li_AI_Choreographer_Music_Conditioned_3D_Dance_Generation_With_AIST_ICCV_2021_paper.html

Ronghui Li, YuXiang Zhang, Yachao Zhang, Hongwen Zhang, Jie Guo, Yan Zhang, Yebin Liu, and Xiu Li. 2024b. Lodge: A Coarse to Fine Diffusion Network for Long Dance Generation Guided by the Characteristic Dance Primitives. 1524–1534. https://openaccess.thecvf.com/content/CVPR2024/html/Li_Lodge_A_Coarse_to_Fine_Diffusion_Network_for_Long_Dance_CVPR_2024_paper.html

Tianyu Li, Jungdam Won, Alexander Clegg, Jeonghwan Kim, Akshara Rai, and Sehoon Ha. 2023b. ACE: Adversarial Correspondence Embedding for Cross Morphology Motion Retargeting from Human to Nonhuman Characters. In *SIGGRAPH Asia 2023 Conference Papers (SA '23)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3610548.3618255

Weiyu Li, Xuelin Chen, Peizhuo Li, Olga Sorkine-Hornung, and Baoquan Chen. 2023a. Example-based Motion Synthesis via Generative Motion Matching. *ACM Trans.*

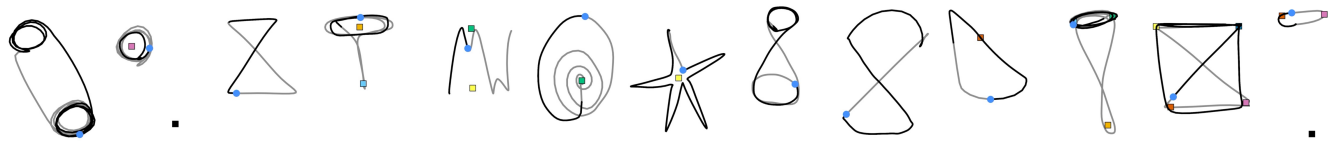


Fig. 11. Examples of sketches created by user study participants. Colored squares represent keyframes. For the animated results of each curve, please refer to the supplementary materials.

Graph. 42, 4 (July 2023), 94:1–94:12. https://doi.org/10.1145/3592395

Jongin Lim, H. Chang, and J. Choi. 2019. PMnet: Learning of Disentangled Pose and Movement for Unsupervised Motion Retargeting. https://www.semanticscholar.org/paper/PMnet%3A-Learning-of-Disentangled-Pose-and-Movement-Lim-Chang/cc0afd01b2b30570c1cd86b05e5ab8d5b8590145

Haiyang Liu, Zihao Zhu, Naoya Iwamoto, Yichen Peng, Zhengqing Li, You Zhou, Elif Bozkurt, and Bo Zheng. 2022. BEAT: A Large-Scale Semantic and Emotional Multi-Modal Dataset for Conversational Gestures Synthesis. *arXiv preprint arXiv:2203.05297* (2022).

Noah Lockwood and Karan Singh. 2012. *Finger Walking: Motion Editing with Contact-Based Hand Performance.* The Eurographics Association. https://doi.org/10.2312/SCA/SCA12/043-052 ISSN: 1727-5288.

Brian McFee, Matt McVicar, Daniel Faronbi, Iran Roman, Matan Gover, Stefan Balke, Scott Seyfarth, Ayoub Malek, Colin Raffel, Vincent Lostanlen, Benjamin van Niekirk, Dana Lee, Frank Cwitkowitz, Frank Zalkow, Oriol Nieto, Dan Ellis, Jack Mason, Kyungyun Lee, Bea Steers, Emily Halvachs, Carl Thomé, Fabian Robert-Stöter, Rachel Bittner, Ziyao Wei, Adam Weiss, Eric Battenberg, Keunwoo Choi, Ryuichi Yamamoto, C. J. Carr, Alex Metsai, Stefan Sullivan, Pius Friesch, Asmitha Krishnakumar, Shunsuke Hidaka, Steve Kowalik, Fabian Keller, Dan Mazur, Alexandre Chabot-Leclerc, Curtis Hawthorne, Chandrashekhar Ramaprasad, Myungchul Keum, Juanita Gomez, Will Monroe, Viktor Andreevitch Morozov, Kian Eliasi, nullmightybofo, Paul Biberstein, N. Dorukhan Sergin, Romain Hennequin, Rimvydas Naktinis, beantowel, Taewoon Kim, Jon Petter Åsen, Joon Lim, Alex Malins, Darío Hereñú, Stef van der Struijk, Lorenz Nickel, Jackie Wu, Zhen Wang, Tim Gates, Matt Vollrath, Andy Sarroff, Xiao-Ming, Alastair Porter, Seth Kranzler, Voodoohop, Mattia Di Gangi, Helmi Jinoz, Connor Guerrero, Abduttayyeb Mazhar, toddrme2178, Zvi Baratz, Anton Kostin, Xinlu Zhuang, Cash TingHin Lo, Pavel Campr, Eric Semeniuc, Monsij Biswal, Shayenne Moura, Paul Brossier, Hojin Lee, and Waldir Pimenta. 2024. librosa/librosa: 0.10.2.post1. https://doi.org/10.5281/zenodo.11192913

Eduardo Fernandes Montesuma, Fred Maurice Ngolè Mboula, and Antoine Souloumiac. 2024. Recent Advances in Optimal Transport for Machine Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024), 1–20. https://doi.org/10.1109/TPAMI.2024.3489030 Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

S. Nyatsanga, T. Kucherenko, C. Ahuja, G. E. Henter, and M. Neff. 2023. A Comprehensive Review of Data-Driven Co-Speech Gesture Generation. *Computer Graphics Forum* 42, 2 (2023), 569–596. https://doi.org/10.1111/cgf.14776 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14776.

Ferda Ofli, Yasemin Demir, Yücel Yemez, Engin Erzin, A. Murat Tekalp, Koray Balcı, İdil Kızoğlu, Lale Akarun, Cristian Canton-Ferrer, Joëlle Tilmanne, Elif Bozkurt, and A. Tanju Erdem. 2008. An audio-driven dancing avatar. *Journal on Multimodal User Interfaces* 2, 2 (Sept. 2008), 93–103. https://doi.org/10.1007/s12193-008-0009-x

Ferda Ofli, Engin Erzin, Yücel Yemez, and A. Murat Tekalp. 2012. Learn2Dance: Learning Statistical Music-to-Dance Mappings for Choreography Synthesis. *IEEE Transactions on Multimedia* 14, 3 (June 2012), 747–759. https://doi.org/10.1109/TMM.2011.2181492 Conference Name: IEEE Transactions on Multimedia.

Gabriel Peyré, Marco Cuturi, and Justin Solomon. 2016. Gromov-Wasserstein Averaging of Kernel and Distance Matrices. In *Proceedings of The 33rd International Conference on Machine Learning*. PMLR, 2664–2672. https://proceedings.mlr.press/v48/peyre16.html ISSN: 1938-7228.

Sigal Raab, Inbal Leibovitch, Guy Tevet, Moab Arar, Amit Haim Bermano, and Daniel Cohen-Or. 2023. Single Motion Diffusion. https://openreview.net/forum?id=DrhZneqz4n

Helge Rhodin, James Tompkin, Kwang In Kim, Kiran Varanasi, Hans-Peter Seidel, and Christian Theobalt. 2014. Interactive motion mapping for real-time character control. *Computer Graphics Forum* 33, 2 (2014), 273–282. https://doi.org/10.1111/cgf.12325 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12325.

Helge Rhodin, James Tompkin, Kwang In Kim, Edilson de Aguiar, Hanspeter Pfister, Hans-Peter Seidel, and Christian Theobalt. 2015. Generalizing wave gestures from sparse examples for real-time character control. *ACM Trans. Graph.* 34, 6 (Nov. 2015), 181:1–181:12. https://doi.org/10.1145/2816795.2818082

Thibault Sejourne, Francois-Xavier Vialard, and Gabriel Peyré. 2021. The Unbalanced Gromov Wasserstein Distance: Conic Formulation and Relaxation. In *Advances in Neural Information Processing Systems*, Vol. 34. Curran Associates, Inc., 8766–8779. https://proceedings.neurips.cc/paper/2021/hash/4990974d150d0de5e6e15a1454fe6b0f-Abstract.html

Yeongho Seol, Carol O'Sullivan, and Jehee Lee. 2013. Creature features: online motion puppetry for non-human characters. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '13)*. Association for Computing Machinery, New York, NY, USA, 213–221. https://doi.org/10.1145/2485895.2485903

Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. 2019. SinGAN: Learning a Generative Model From a Single Natural Image. 4570–4580. https://openaccess.thecvf.com/content_ICCV_2019/html/Shaham_SinGAN_Learning_a_Generative_Model_From_a_Single_Natural_Image_ICCV_2019_paper.html

Ari Shapiro and Sung-Hee Lee. 2011. Practical Character Physics for Animators. *IEEE Computer Graphics and Applications* 31, 4 (July 2011), 45–55. https://doi.org/10.1109/MCG.2010.22 Conference Name: IEEE Computer Graphics and Applications.

Takaaki Shiratori and Katsushi Ikeuchi. 2008. Synthesis of Dance Performance Based on Analyses of Human Motion and Music. *Information and Media Technologies* 3, 4 (2008), 834–847. https://doi.org/10.11185/imt.3.834

Takaaki Shiratori, Atsushi Nakazawa, and Katsushi Ikeuchi. 2006. Dancing-to-Music Character Animation. *Computer Graphics Forum* 25, 3 (2006), 449–458. https://doi.org/10.1111/j.1467-8659.2006.00964.x _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2006.00964.x.

Li Siyao, Weijiang Yu, Tianpei Gu, Chunze Lin, Quan Wang, Chen Qian, Chen Change Loy, and Ziwei Liu. 2022. Bailando: 3D Dance Generation by Actor-Critic GPT With Choreographic Memory. 11050–11059. https://openaccess.thecvf.com/content/CVPR2022/html/Siyao_Bailando_3D_Dance_Generation_by_Actor-Critic_GPT_With_Choreographic_Memory_CVPR_2022_paper.html

Justin Solomon, Gabriel Peyré, Vladimir G. Kim, and Suvrit Sra. 2016. Entropic metric alignment for correspondence problems. *ACM Trans. Graph.* 35, 4 (July 2016), 72:1–72:13. https://doi.org/10.1145/2897824.2925903

Seyoon Tak and Hyeong-Seok Ko. 2005. A physically-based motion retargeting filter. *ACM Trans. Graph.* 24, 1 (Jan. 2005), 98–117. https://doi.org/10.1145/1037957.1037963

Kenta Takeuchi, Dai Hasegawa, Shinichi Shirakawa, Naoshi Kaneko, Hiroshi Sakuta, and Kazuhiko Sumi. 2017. Speech-to-Gesture Generation: A Challenge in Deep Learning Approach with Bi-Directional LSTM. In *Proceedings of the 5th International Conference on Human Agent Interaction (HAI '17)*. Association for Computing Machinery, New York, NY, USA, 365–369. https://doi.org/10.1145/3125739.3132594

S. C. L. Terra and R. A. Metoyer. 2004. Performance timing for keyframe animation. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation (SCA '04)*. Eurographics Association, Goslar, DEU, 253–258. https://doi.org/10.1145/1028523.1028556

Matthew Thorne, David Burke, and Michiel van de Panne. 2004. Motion doodles: an interface for sketching character motion. *ACM Trans. Graph.* 23, 3 (Aug. 2004), 424–431. https://doi.org/10.1145/1015706.1015740

Alexis Thual, Quang Huy Tran, Tatiana Zemskova, Nicolas Courty, Rémi Flamary, Stanislas Dehaene, and Bertrand Thirion. 2022. Aligning individual brains with fused unbalanced Gromov Wasserstein. *Advances in Neural Information Processing Systems* 35 (Dec. 2022), 21792–21804. https://proceedings.neurips.cc/paper_files/paper/2022/hash/8906cac4ca58dcaf17e97a0486ad57ca-Abstract-Conference.html

Truebones. 2025. Truebones ZOO. https://truebones.gumroad.com/l/skZMC

Jonathan Tseng, Rodrigo Castellon, and Karen Liu. 2023. EDGE: Editable Dance Generation From Music. 448–458. https://openaccess.thecvf.com/content/CVPR2023/html/Tseng_EDGE_Editable_Dance_Generation_From_Music_CVPR_2023_paper.html

Shuhei Tsuchida, Satoru Fukayama, Masahiro Hamasaki, and Masataka Goto. 2019. AIST Dance Video Database: Multi-genre, Multi-dancer, and Multi-camera Database for Dance Information Processing. In *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019*. Delft, Netherlands.

Ruben Villegas, Jimei Yang, Duygu Ceylan, and Honglak Lee. 2018. Neural Kinematic Networks for Unsupervised Motion Retargetting. 8639–8648. https://openaccess.thecvf.com/content_cvpr_2018/html/Villegas_Neural_Kinematic_Networks_CVPR_2018_paper.html

Andrew Witkin and Michael Kass. 1988. Spacetime constraints. *SIGGRAPH Comput. Graph.* 22, 4 (June 1988), 159–168. https://doi.org/10.1145/378456.378507

Bowen Wu, Chaoran Liu, Carlos T. Ishi, and Hiroshi Ishiguro. 2021. Probabilistic Human-like Gesture Synthesis from Speech using GRU-based WGAN. In *Companion Publication of the 2021 International Conference on Multimodal Interaction (ICMI '21 Companion)*. Association for Computing Machinery, New York, NY, USA, 194–201. https://doi.org/10.1145/3461615.3485407

Ming Xu and Stephen Gould. 2024. Temporally Consistent Unbalanced Optimal Transport for Unsupervised Action Segmentation. 14618–14627. https://openaccess.thecvf.com/content/CVPR2024/html/Xu_Temporally_Consistent_Unbalanced_Optimal_Transport_for_Unsupervised_Action_Segmentation_CVPR_2024_paper.html

Katsu Yamane, Yuka Ariki, and Jessica Hodgins. 2010. Animating non-humanoid characters with human motion data. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '10)*. Eurographics Association, Goslar, DEU, 169–178.

Payam Jome Yazdian, Mo Chen, and Angelica Lim. 2021. Gesture2Vec: Clustering Gestures using Representation Learning Methods for Co-speech Gesture Generation. (Oct. 2021). https://openreview.net/forum?id=0Kj5mhn6sw

Wataru Yoshizaki, Yuta Sugiura, Albert C. Chiou, Sunao Hashimoto, Masahiko Inami, Takeo Igarashi, Yoshiaki Akazawa, Katsuaki Kawachi, Satoshi Kagami, and Masaaki Mochimaru. 2011. An actuated physical puppet as an input device for controlling a digital manikin. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. Association for Computing Machinery, New York, NY, USA, 637–646. https://doi.org/10.1145/1978942.1979034

Wentao Zhu, Xiaoxuan Ma, Dongwoo Ro, Hai Ci, Jinlu Zhang, Jiaxin Shi, Feng Gao, Qi Tian, and Yizhou Wang. 2024. Human Motion Generation: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46, 04 (April 2024), 2430–2449. https://doi.org/10.1109/TPAMI.2023.3330935 Publisher: IEEE Computer Society.