Project Report

Body Motion Generation

Team Name: P01
Shiwei Tey, Saurabh Chaudhari, Xingxuan Li, Yifan Zhang

1 Project Objective

Body motion generation aims to generate a simulation of human body motions for a target person's 3D body structure, based on a continuous motion sequence of a reference person's 3D body postures. Such technology has been widely used to enhance virtual reality and movie special effects to create characters that are able to mimic human actions. In this task, we assume that the source and target models have the same structure, i.e., the same number of joints. However, the length between two joints can be different (refer to Figure 1).

Since the bone lengths differ between the source and target models, naive approaches to directly map the reference points to the target body structure would not generate accurate action simulation. Considering an example where in the reference body is touching its heads with both arms, if the length of arms of the target body are significantly longer than that of the reference body then the arms will occupy the same space as head which is a clear violation. Therefore, the relative movements of each joint cannot be computed naively for the target. In this project, we aim to formally define this task and design an algorithm to solve the mentioned challenge.

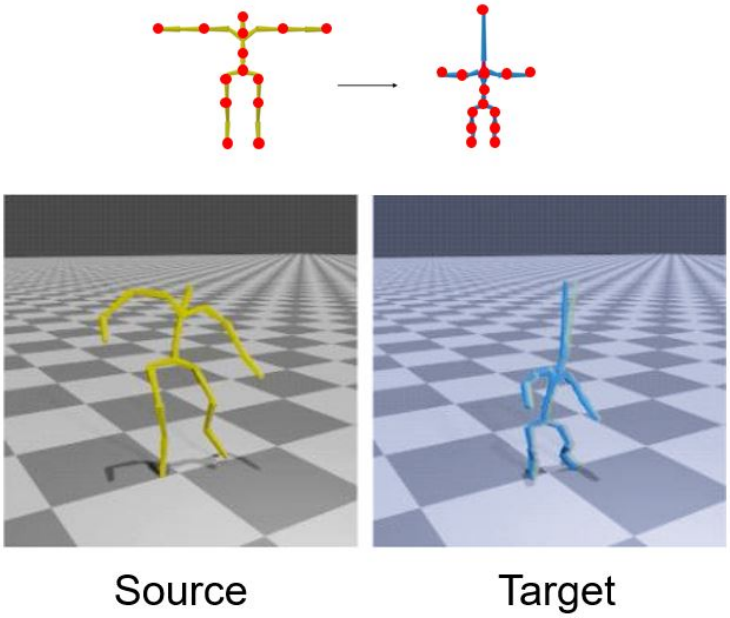

Figure 1: Different lengths of bones between the source and target models.

2 Entities and Relationships

In this task, we intend to generate a continuous motion sequence for a given target body structure, based on the motion of a reference body. Entities involved in this task consist of a reference body and a target body. We will have a prior understanding of both structures, including information pertaining to structural joints, and length/orientations of bones between pairs of joints.

2.1 Known and Unknown Entities

Here, we formulate the body structure as a graph structure, where the joints serve as the nodes and the bones serve as the edges.

- Known
  - Entities
    - Reference structure model
    - Target neutral structure model
  - Elements
    - Joints on structure
  - Attributes
    - Positions of joints in reference model
    - Positions of joints in target neutral model
    - Orientations of bones between joints of reference model
- Unknown
  - Entities
    - Resultant structure model
  - Attributes
    - Positions of joint points for resultant structure
    - Orientations of bones between joints for resultant structure

2.2 Relationships

- The posture of the resultant model should match that of the reference at a given time frame
- The body structure of the resultant model should match that of the target model during the whole motion
- In each movement, the orientations of bones between the resultant model and the reference model should be similar

3 Problem Definition

In body motion generation, the corresponding relationships between the source and target bodies are known in advance. We know that for each movement the orientations of corresponding bones between the resultant model and the reference model should match as closely as possible. Inspired by this, we define the optimization objective to enforce the orientations of bones between the resultant model and the reference model to be as similar as possible. Meanwhile, the target motion is also required to satisfy some other relationship constraints. We formally formulate the problem below.

- Inputs

  - Reference motion: The smooth reference motion consists of a sequence of body postures, $(F_0, F_1, ..., F_N)$, where each F_i denotes a body posture at the time i, and N denotes the total posture number. Moreover, F_0 is the reference neutral model.
  - Reference model: In each movement, the reference model F_i is a tree structure, consisting of a set of the body joints as nodes and bones as edges. For each F_i, we denote the joint node as v_i^j and denote the bone edge as b_i^l, where $j \in [n]$ denotes the index of the joint and $l \in [n-1]$ denotes the index of the bone. The coordinates for each joint w.r.t. the root node is given by $\mathbf{p}(v_i^j) \in \mathbb{R}^3$ (for simplicity, we use $\mathbf{p}_i^j$ here onwards). For example, in Figure 2, the relative position of the joint v^1 can be obtained by treating its root node v^0 as the local origin. Each node v_i^j also has a global position attribute $\mathbf{q}(v_i^j) \in \mathbb{R}^3$ (for simplicity, we use $\mathbf{q}_i^j$ here onwards).

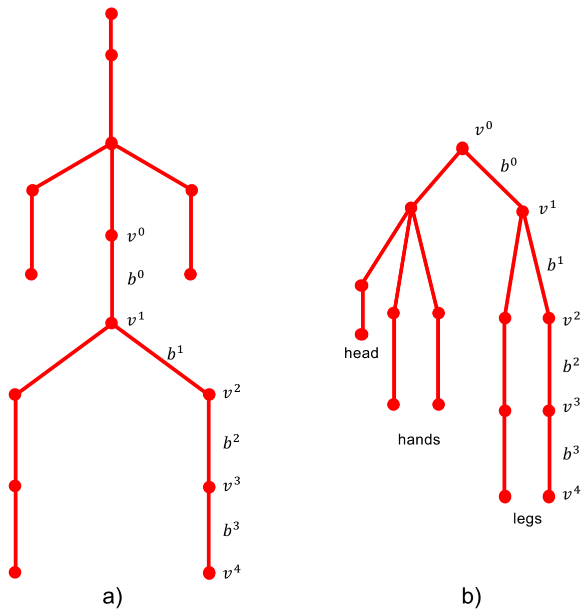

Figure 2: a) Human body bones and nodes b) Human body into a tree

  - Target neutral model: The target neutral model T_0 is a tree structure with the indexing similar to the reference structure. For T_0, we denote the joint node by $\hat{v}_0^j$ and bone edge by $\hat{b}_0^l$. Each node $\hat{v}_0^j$ has a global coordinates position attribute denoted by $\mathbf{q}(\hat{v}_0^j) \in \mathbb{R}^3$ (for simplicity, we use $\hat{\mathbf{q}}_0^j$ here onwards) and a relative position w.r.t. the root node denoted by $\mathbf{p}(\hat{v}_0^j) \in \mathbb{R}^3$ (for simplicity, we use $\hat{\mathbf{p}}_0^j$ here onwards).
  - Special motion constraints: Constraints set C for special postures like clapping action where both hands touch each other.

- Outputs

  - Posture mapping function $f{:}(T_0; F_0; F_i) \to T_i$. Given both neutral positions of reference and target model, as well as reference posture at time i, the function outputs the target posture at time i. Specifically, for one joint in target model, $\hat{\mathbf{q}}_i^j = \prod_{k=0}^{j} \hat{\mathbf{A}}_i^k \hat{\mathbf{p}}_0^j$, where $\hat{\mathbf{A}}_i^k$ is a transformation matrix for the k-th bone at time i on the target model, which contains both rotational as well as spatial information. Here, we transform the local and global coordinates of each point to their homogeneous variants. Moreover,

$\hat{\mathbf{A}}_i^k$ can be expressed as $\hat{\mathbf{A}}_i^k = \begin{bmatrix} \hat{\mathbf{R}}_i^k & \hat{\mathbf{t}}^k \\ 0^T & 1 \end{bmatrix}$, where $\hat{\mathbf{R}}_i^k$ is the rotation of the k-th bone at time i and $\hat{\mathbf{t}}^k$ is the spatial transform of the bone to make sure that two neighbouring bones are connected. Similarly for reference model, $\mathbf{q}_i^j = \prod_{k=0}^{j} \mathbf{A}_i^k \mathbf{p}_0^j$, where $\mathbf{A}_i^k = \begin{bmatrix} \mathbf{R}_i^k & \mathbf{t}^k \\ 0^T & 1 \end{bmatrix}$. Moreover, when $k > 0$, both $\hat{\mathbf{t}}^k$ and $\mathbf{t}^k$ are fixed for all postures within the same model and can be calculated easily on the neutral postures, respectively. For example, $\hat{\mathbf{q}}_i^2$ is based on the transformations of previous parent joints. Thus, $\hat{\mathbf{q}}_i^2 = \begin{bmatrix} \hat{\mathbf{R}}_i^2 & \hat{\mathbf{t}}^2 \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{R}}_i^1 & \hat{\mathbf{t}}^1 \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{R}}_i^0 & \hat{\mathbf{t}}^0 \\ 0^T & 1 \end{bmatrix} \hat{\mathbf{p}}_0^2$, where $\begin{bmatrix} \hat{\mathbf{R}}_i^0 & \hat{\mathbf{t}}^0 \\ 0^T & 1 \end{bmatrix}$ is the transformation of the local origin (root node) with respect to global origin. Note that, when $k > 0$, $\hat{\mathbf{t}}^k$ and $\mathbf{t}^k$ are not necessarily the same for the same posture because reference model and target model have different body structures. However, $\hat{\mathbf{t}}^0$ and $\mathbf{t}^0$ are the same because they are the transformation of root node with respect to the global origin.

  - Resultant target motion $T_i = f(T_0; F_0, F_i)$, for all $i \in [N]$. For T_i, all notations $\hat{v}_i^j, \hat{\mathbf{q}}_i^j, \hat{\mathbf{p}}_i^j$) are defined in the same way to the target neutral posture.

- Optimization Objectives
  - Optimize $\hat{\mathbf{R}}_i^k$ by minimizing $\sum_i \sum_k (||\hat{\mathbf{R}}_i^k - \mathbf{R}_i^k||^2)$, where values of $\mathbf{R}_i^k$ were obtained from F_i. This would provide us the transformation matrix $\hat{\mathbf{A}}_i^k$ for every k to generate our target posture $T_i = f(T_0; F_0, F_i)$.

- Constraints
  - Special motion constraint C. For simplicity, this report only considers the clapping action at the frame i_c as an example, where two hands should touch each other. Specifically, considering j_1 and j_2 as the indexes of left and right hands respectively. The constraint can be specified as $C_{i_c} = \{(\|\hat{\mathbf{q}}_i^{j_1} - \hat{\mathbf{q}}^{j_{i_c}}\|^2 + \|\hat{\mathbf{q}}_i^{j_2} - \hat{\mathbf{q}}^{j_{i_c}}\|^2)\}$. This constraint would ensure that the hands touch together at the position coordinate given by $\hat{\mathbf{q}}^{j_{i_c}}$. For interpretation purpose, we can consider the notation j_{i_c} (i.e., , the superscript of $\hat{\mathbf{q}}^{j_{i_c}}$) as a simple auxiliary node. Attribute value of $\hat{\mathbf{q}}^{j_{i_c}}$ is derived from the coordinate of the point where the left and right hands of reference model touched each other.
  - All $\mathbf{R}_i^k$ should be rotation matrices (orthogonal with the determinant of 1)

- Analysis of Problem Definition
  - Objective function: we try to minimize the orientation differences between each corresponding bone in the target and reference models. This would ensure that the overall posture of the target body is as close as possible to the reference body.
  - Posture mapping function $f(T_0; F_0, F_i) \rightarrow T_i$: In the function f, all $\hat{\mathbf{R}}_i^k$ and $\hat{\mathbf{t}}_i^0$ are initialized by fitting F_0 and F_i. Moreover, all $\hat{t}_i^k (k > 0)$ are obtained through the target neutral posture. Then, based on all $\hat{\mathbf{t}}_i^k$ and $\hat{\mathbf{R}}_i^k$,we can compute T_i based on T_0.
  - Constraint: The complete disregard to the positional awareness and relying only on orientation matching might result in some trivial posture generation which are similar in principle but not similar considering the semantics of the posture. For example, different lengths of arms would not exactly replicate a salute action even if the orientations are same. This is mitigated by the special motion constraint. This would mean that now due to this added constraint, the semantics meaning of the posture is achieved such that the orientation difference is minimum.

4 Algorithm

In this section, we propose a new algorithm to generate the target motion from the reference motion. Before that, we first repeat several important notations given the time i:

- $\mathbf{p}_i^j, \mathbf{q}_i^j$ denote the local and global coordinates for the joint j on the reference structure
- $\mathbf{A}_i^k = \begin{bmatrix} \mathbf{R}_i^k & \mathbf{t}^k \\ 0^T & 1 \end{bmatrix}$ is the transformation matrix for the bone k on the reference structure
- $\hat{\mathbf{p}}_i^j, \hat{\mathbf{q}}_i^j, \hat{\mathbf{A}}_i^k = \begin{bmatrix} \hat{\mathbf{R}}_i^k & \hat{\mathbf{t}}^k \\ 0^T & 1 \end{bmatrix}$ are the corresponding notations for the target structure

To solve all $\hat{\mathbf{A}}_i^k$ (i.e., , $\hat{\mathbf{R}}_i^k$ and $\hat{\mathbf{t}}^k$) for target posture generation, our proposed algorithm consists of the following two steps.

1. Good initialization for $\hat{\mathbf{R}}_i^k$ and $\hat{\mathbf{t}}^0$ by computing reference transformations based on the reference neutral posture F_0 and the reference posture F_i (Section 4.1)
2. Compute $\hat{\mathbf{t}}^k$ and refine $\hat{\mathbf{R}}_i^k$ based on the target neutral posture T_0 and the reference posture F_i (Section 4.2)

4.1 Initialization of Transformation Matrices

Since the target posture should as close as possible to the reference posture, we initialize all $\hat{\mathbf{R}}_i^k$ and $\hat{\mathbf{t}}^0$ for the target posture by setting them to $\mathbf{R}_i^k$ and $\mathbf{t}^0$ of the reference posture. To achieve this, we propose the following algorithm to compute all $\mathbf{R}_i^k$ and $\mathbf{t}^k$. As shown in Figure 2, the reference and target structures have a total 5 branches, which are the head, left-right arms and left-right root. Therefore, the proposed algorithm will be run for every branches and every time frame independently. Note that, $\mathbf{R}$ is defined as a 3D rotational matrix where $\mathbf{R} = f(\theta_x, \theta_y, \theta_z)$. That is, in our algorithm, **optimizing $\mathbf{R}$ refers to finding and updating the value of the rotational angles.**

Algorithm 1 Reference Transformation Matrices

Input: - Reference posture F_i: joints $\{v_i^j\}_{j=1}^n$ with local/global positions $\{\mathbf{p}_i^j\}_{j=1}^n$ and $\{\mathbf{q}_i^j\}_{j=1}^n$;
- Reference neutral model F_0: joints $\{v_0^j\}_{j=1}^n$ with positions $\{\mathbf{p}_0^j\}_{j=1}^n$ and $\{\mathbf{q}_0^j\}_{j=1}^n$;

Output: Reference transformation matrices $\mathbf{A}_i^k = \begin{bmatrix} \mathbf{R}_i^k & \mathbf{t}^k \\ 0^T & 1 \end{bmatrix}$.

```
Compute t^k(k > 0) on the neutral model F_0;  // then t^k(k>0) are fixed;
for each branch in the body tree structure do
    // for convenience, we reindex the matrices by j in the branch
    for each j do
        if j = 0 then
            Obtain R_i^0, t^0 by solving the linear equation q_i^1 = A_i^0 p_i^1 ;
        else if j is not the leaf node in the branch then
            Obtain R_i^j by solving linear equations q_i^{j+1} = ∏_{l=0}^{j} A_i^l p_i^{j+1}.
        end if
    end for
end for
```

- Line 1: In reference neutral posture, given the local positions of joints $\mathbf{p}_0^0, \mathbf{p}_0^1, ..., \mathbf{p}_0^j$ and $\mathbf{t}^0 = 0$, unknown $\mathbf{t}^k (k > 0)$ can be computed by $\mathbf{t}^k = \mathbf{p}_0^k$ (here, for convenience, we represent the index of each joint by its corresponding matrix index k)

- Line 6: When $j = 0$, we have the following equation:

$$\mathbf{q}_i^0 = \begin{bmatrix} \mathbf{R}_i^0 & \mathbf{t}_i^0 \\ 0^T & 1 \end{bmatrix} \mathbf{p}_0^0.$$

  Obtain $\mathbf{R}_i^0, \mathbf{t}_0$ from $\mathbf{A}_i^0$ by solving linear equations with the close-form solution.

- Line 8: When $j > 0$, we have:

$$\mathbf{q}_i^j = \prod_{l=0}^{j} \mathbf{A}_i^l \mathbf{p}_i^j,$$

$$(\mathbf{A}_i^{j-1})^{-1} (\mathbf{A}_i^{j-2})^{-1} ... (\mathbf{A}_i^0)^{-1} \mathbf{q}_i^j = \mathbf{A}_i^j \mathbf{p}_i^j.$$

  Note that all $\mathbf{t}^k$ and $\mathbf{A}_i^0 ... \mathbf{A}_i^{j-2} \mathbf{A}_i^{j-1}$ are fixed and known, so $\mathbf{R}_i^j$ can be obtained by solving $\mathbf{A}_i^j$ from the above linear equations.

4.2 Refinement of Transformation Matrices

By now, we have initialize all $\hat{\mathbf{R}}_i^k$ and $\hat{\mathbf{t}}_i^0$ for the target posture based on the reference posture. Then, we compute and refine all $\hat{\mathbf{A}}_i^k$ for generating the target posture.

Algorithm 2 Refinement of Target Transformation Matrices

Input: - Target neutral model T_0: joints $\{\hat{v}_0^j\}_{j=1}^n$ with positions $\{\hat{\mathbf{p}}_0^j\}_{j=1}^n$ and $\{\hat{\mathbf{q}}_0^j\}_{j=1}^n$;
- Special motion constraint point $\hat{\mathbf{q}}^{j_{i_c}}$;
- Reference transformation matrices $\mathbf{A}_i^k = \begin{bmatrix} \mathbf{R}_i^k & \mathbf{t}^k \\ 0^T & 1 \end{bmatrix}$;

Output: Target transformation matrices $\hat{\mathbf{A}}_i^k$.

1: Compute $\hat{\mathbf{t}}^k (k > 0)$ on the target neutral model T_0; // then $\hat{\mathbf{t}}^k (k > 0)$ are fixed;
2: Initialize $\hat{\mathbf{t}}^0 = \mathbf{t}^0$ and $\hat{\mathbf{R}}_i^k = \mathbf{R}_i^k$ for all k, and then $\hat{\mathbf{A}}_i^k = \begin{bmatrix} \hat{\mathbf{R}}_i^k & \hat{\mathbf{t}}^k \\ 0^T & 1 \end{bmatrix}$;
3: **repeat**
4: Compute the target posture T_i based on T_0 and all $\hat{\mathbf{A}}_i^k$;
// T_i contains all joints $\{\hat{v}_i^j\}_{j=1}^n$ and their global coordinate $\{\hat{q}_i^j\}_{j=1}^n$
5: Refine all $\hat{\mathbf{R}}_i^k$ by minimizing $\sum_k (||\hat{\mathbf{R}}_i^k - \mathbf{R}_i^k||^2) + \lambda C_{i_c}$ by gradient descent.
// $C_{i_c} = (\|\hat{\mathbf{q}}_i^{j_1} - \hat{\mathbf{q}}^{j_{i_c}}\|^2 + \|\hat{\mathbf{q}}_i^{j_2} - \hat{\mathbf{q}}^{j_{i_c}}\|^2)$ is the clapping constraint at the frame i_c
// Using Lagrange methods to merge the constraint into the objective
6: **until** convergence

- Line 1: In target neutral posture, given the local positions of joints $\hat{\mathbf{p}}_0^0, \hat{\mathbf{p}}_0^1, ..., \hat{\mathbf{p}}_0^j$ and $\hat{\mathbf{t}}_0^0 = 0$, unknown $\hat{\mathbf{t}}^k (k > 0)$ can be computed by $\hat{\mathbf{t}}^k = \hat{\mathbf{p}}_0^k$ (here, for convenience, we represent the index of each joint by its corresponding matrix index k).

- Line 2: Since the target posture should as close as possible to the reference posture, we initialize all $\hat{\mathbf{R}}_i^k$ and $\hat{\mathbf{t}}^0$ for the target posture by setting them to $\mathbf{R}_i^k$ and $\mathbf{t}^0$ of the reference posture.

- Line 5: C_{i_c} is the clapping constraint. To make the algorithm more self-contained, we have the following clarifications.
  - If there isn't any constraint at time i, then $C_i = 0$. That is, the target posture copies the exact rotational transformation $\mathbf{R}_i^k$ from the reference posture. Therefore, the objective function to make the orientations of bones the same is satisfied.
  - If there is a constraint at time i_c, we regard the constraint as a soft constraint and use Lagrange method to introduce it into the objective. C_{i_c} is a function that contains $\hat{\mathbf{R}}_i^k$.
  - The penalty for the constraint λ is a hyper parameter to adjust such that if the constraint is supposed to be strict, the value of λ should be high. Such a problem can be solved by gradient descent.

4.3 Overall Algorithm

The overall algorithm can now be summarized as follows:

Algorithm 3 Body Motion Generation

Input: - Reference motion $\{F_i\}_{i=0}^N$ with joint nodes $\{v_i^j\}_{j=1}^n$ and edges $\{b_i^k\}_{k=1}^{n-1}$;
- Target neutral model T_0;

1: **for** each i **do**
2: Compute the reference transformations $\mathbf{A}_i^k$ based on Algorithm 1;
3: Initialize part target transformations $\hat{\mathbf{A}}_i^k$ based on $\mathbf{A}_i^k$;
4: Compute and refine target transformations $\hat{\mathbf{A}}_i^k$ based on Algorithm 2;
5: Compute the target motion T_i based on T_0 and all $\hat{\mathbf{A}}_i^k$;
6: **end for**
7: **Output**: Target motion $\forall i : T_i$.

The proposed algorithm has the following properties.

- Correctness
  - The algorithm satisfies the optimization objective by enforcing the orientations of each target bones to be similar to the reference bones.
  - The specific actions are treated as a soft constraint by using the Lagrange method to match the desire action as much as possible. The special motion constraints can be better satisfied by increasing λ. Due to the difference in structural length between reference and target, are hard constraint might not lead to an optimal solution.
  - Rigid body is satisfied using similarity transformation ($det(R) = 1$, achieved by learning angles)
- Convergence
  - Initialization of $\hat{\mathbf{R}}_i^k = \mathbf{R}_i^k$ and $\hat{\mathbf{t}}^0 = \mathbf{t}^0$ are suitable since the target posture should be as similar as possible to the reference. That is, each bone in the target posture is similar in orientation to each bone in the reference posture under some special motion constraints.
  - Initialization of $\hat{\mathbf{t}}^k$ based on the target neutral posture is appropriate, since the target structures should be rigidly connected.
  - If there is no any constraint, then the objective is minimal. If there is a constraint, then we iterate the algorithm such that the minimal objective value can be obtained by gradient descent.

- Hence, the target motion should converge to at least a suboptimal approximation of the reference motion.

- Strengths and Drawbacks
  - Strengths: This algorithm is able to solve the optimization function while satisfying the special actions, and handle different scales of source and target bodies.
  - Drawbacks: We treated each special motion constraints as soft constraints, which may make the target motion fail to do a perfect action if the structural differences are significant.