
Article:
Wang, H, Ho, ESL and Komura, T (2015) An energy-driven motion planning method for two distant postures. IEEE Transactions on Visualization and Computer Graphics, 21 (1). 1. pp. 18-30. ISSN: 1077-2626

https://doi.org/10.1109/TVCG.2014.2327976

An Energy-driven Motion Planning Method for Two Distant Postures

He Wang, Edmond S. L. Ho, and Taku Komura

Abstract—In this paper, we present a local motion planning algorithm for character animation. We focus on motion planning between two distant postures where linear interpolation leads to penetrations. Our framework has two stages. The motion planning problem is first solved as a Boundary Value Problem (BVP) on an energy graph which encodes penetrations, motion smoothness and user control. Having established a mapping from the configuration space to the energy graph, a fast and robust local motion planning algorithm is introduced to solve the BVP to generate motions that could only previously be computed by global planning methods. In the second stage, a projection of the solution motion onto a constraint manifold is proposed for more user control. Our method can be integrated into current keyframing techniques. It also has potential applications in motion planning problems in robotics.

Index Terms—Character Animation, Motion Planning

1 Introduction

Synthesizing movements that involve close interactions, such as those in wrestling, dancing, carrying objects, and passing through constrained environments, is essential for applications such as computer animations, computer games, and robotics. Collisions and contacts between individual body parts and the body with objects in the environment are very common, which makes the path planning problem extremely difficult to solve.

One common practice in computer animation for synthesizing such movements is keyframing. Key postures are inserted manually between frames so that every two consecutive postures can be smoothly interpolated. If we see the generated motion as a function, this method finds the motion function passing through the initial and end posture in the configuration space. This assumes every two consecutive postures are close enough so that the motion function can be approximated well by a piece-wise linear function. However, due to the representation of postures based on joint angles, which does not take into account the spatial relationships between different body parts or the body and objects in the environment, many key frames are required to produce a penetration free, natural motion. As a result, only very experienced animators can produce a realistic motion that involves such close interactions.

- *H. Wang is now with Disnery Research LA. He was with the School of Informatics, University of Edinburgh, Edinburgh, EH8 9AB, UK. E-mail: He.Wang@ed.ac.uk see http://homepages.inf.ed.ac.uk/hwang3/index.html*
- *E. S. L. Ho is with the Department of Computer Science,Hong Kong Baptist University*
- *T. Komura is with the School of Informatics, University of Edinburgh.*

Alternatively, global motion planning approaches such as Rapidly-exploring Random Trees (RRT) [1] or Probabilistic Road Maps (PRM) [2] are possible solutions. These methods explore the configuration space in a random fashion to find possible solutions connecting postures. However, the computational expense is high due to the curse of dimensionality [3]. In addition, they become slow when there are narrow passages in the configuration space, which is usually the case when there are many body parts and objects in close proximity.

In this paper, we propose a method to solve this problem by composing an energy graph where distances between different body parts are taken into account. Smooth motions are calculated as geodesic lines between the start and end nodes on the energy graph. Based on this concept, we form the motion planning problem between two arbitrary postures of a human skeleton character as a Boundary Value Problem (BVP).

Besides the boundary conditions, the formulation of the BVP needs to consider three factors: (1) **Physical plausibility**, for instance, there should be no penetrations. (2) **Smoothness**. (3) **Incorporation of user control**. First, we construct an energy graph that meets these requirements. Next, we develop an iterative algorithm to compute the geodesic line between the two postures. Finally, we project the motion function onto a constraint manifold incorporating factors such as physical constraints and additional user control.

The structure of the rest of the paper is as follows. We first review closely related work in computer animation, robotics and computational geometry in Section 2. Then we give an overview of our method in Section 3, followed by detailed explanations of the methodology in Section 4, Section 5 and Section 6. Then, experimental results are presented and evalu-

ated in Section 7. Finally, limitations are discussed in Section 8 and the conclusion is drawn in Section 9.

2 Related Work

Character animation has been an active field in the past two decades. Although we are trying to target a classic problem, we take a new perspective by modeling the posture via interactions between different body parts. It narrows our review scope down to interaction simulation and path-planning for complex movements. We first review some works of simulating close interactions for character animation. Then we move on to path-planning methods for synthesizing movements of close interactions. Finally, we review the work in computational geometry that handles similar problems.

2.1 Interaction in animation

Character interactions have been attracting many researchers. Liu et al. [4] simulate the close dense interactions of two characters by repetitively updating the motion of each character with spacetime constraints. Lee and Lee [5] produce a boxing match by reinforcement learning. Treuille et al. [6] use reinforcement learning to simulate pedestrians avoiding each other. Shum et al. [7] use interaction patches to synthesize multi-character fighting. However, these interactions do not include movements, such as Yoga, where body parts are in proximity and penetrations cannot be avoided by simple methods.

Kallmann et al. [8] propose a Probabilistic Roadmap-based planning algorithm to synthesize reaching and grasping motion while assuming prior knowledge about the motion. Pan et al. [9] propose an efficient method to plan the motion in constrained environment by classifying body parts into groups with low-correlation. Their method assumes that some body parts can always be de-correlated, which is not always the case in motions such as Yoga and Contortion. Wang et al. [10], [11] suggest control methods to interact with deformable objects. They deal with a high number of degrees of freedom but the character control is too simple to handle impending penetrations.

Ho and Komura [12], [13] propose to use topological measures to encode complex interactions so that very close and continuous interactions can be analyzed and synthesized. However, their methods mainly encode tangling information which is only useful when tangles are present.

2.2 Path-planning of close interactions

Early work in path planning [14] employs potential fields to compute collision-free paths efficiently. Kuffner et al. [15] propose a fast distance determination method to detect self-collision of the body parts of humanoid robots using Voronoi-clip algorithm. Sugiura et al. [16] propose a method to avoid self-collision for humanoid robots by adding virtual forces between the body segments to maintain the minimum distances while reaching the target configuration.

Global path-planning such as Rapidly-exploring Random Trees (RRT) [1] or Probabilistic Road Maps (PRM) [2] are often used in both animation and robotics. Yamane et al. [17] simulate motions to move luggage from one place to another by combining IK and RRT. Hirano et al [18] and Berenson et al. [19] propose to use RRT to search the state space for grasping objects. Zhang et al. [20] propose an approach similar to [9] and blend the motion with MOCAP data to produce realistic human motion. Ho and Komura [21] use RRT to plan motions such as holding and piggybacking. Ladd and Kavraki [22] develop a variation of RRT to untangle knots. Their work is similar to ours in the way that they use an energy function to break the global path-planing problem into local problems that are manageable. There are two shortcomings for these methods: 1. They are usually computationally expensive. In order to search for collision-free paths, it requires sampling in a large space and evaluation for various factors. 2. Because the final path is built upon random search, the motion style is not consistent. Besides, post-processing is usually needed to ensure the smoothness. Therefore, these approaches are not ideal from the viewpoint of character motion synthesis and editing.

The most similar work to ours is [23], which uses linkage unfolding to plan the motions. However, their method is solely based on linkage folding/unfolding. The user does not have enough control over the motion synthesis process.

2.3 Linkage unfolding in computational geometry

Since characters are commonly represented by tree-structure skeletons, character motions can be seen as motions by 3D polygonal trees, that have been investigated in computational geometry. More specifically, a number of researchers have looked into the configuration space of linkages for folding/unfolding. One main question in this area is what types of linkages always have connected configuration spaces [24]. If the configuration is connected, it is called *unlocked*, and any two configurations can be interpolated. Otherwise it is called *locked*. In 3D, a locked open chain of five links is found by Cantarella and Johnson [25] (see Figure 1). In [26], Connelly et al. prove that there are no locked 2D polygonal linkages. Upon this discovery, Cantarella et al. [27] propose an energy-driven approach to unfold such linkages, and Iben et al. [28] propose an extended approach to interpolate arbitrary polygonal linkage configurations.

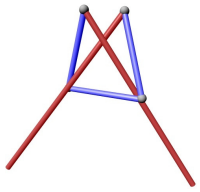


Fig. 1: A locked linkage of five links. This linkage cannot be unfolded into a flat one without penetrations or change of link lengths.

The research of linkage folding/unfolding greatly inspired our research. We show how we leverage their techniques to facilitate the motion generation for complex postures.

3 Overview

The whole framework of our system is shown in Figure 2. Two key postures and the user control signals are the inputs to our system. Then we generate the motion via a two-stage process. The first stage is a Boundary Value Problem solver, where we calculate the valid motion between the two postures. The second stage is a spatial-temporal solver which projects the resulting motion onto a constraint manifold given by the user.

3.1 Contributions

- We propose an energy graph representation for motion planning between distant postures as Boundary Value Problems.
- We introduce a fast and robust local planning scheme to solve BVPs which can produce motions that could only previously be achieved by global path-planning algorithms in the past.

4 Energy Graph

The joint angle representation does not encode penetrations for human postures. It would be easier if we can form this problem in a different space, denoted by EG, where constraints are encoded into the representation and path-planning can be performed locally. The space should meet several requirements:

1) Valid postures are aggregated together while invalid postures are marginalized. Here, the validity mainly refers to no penetration. The boundary between them should be clear so that we can plan the motion in the subspace where all postures are valid.
2) The global structure is simple so that the path-planning between two points in EG is straightforward.
3) The mapping between the configuration space C to EG should be injective and has at least first-order continuity, so that a continuous motion in C can be mapped continuously into EG.
4) The mapping can be easily calculated.

Finding an analytical form of EG is hard. But we can establish the mapping between C and EG as follows.

4.1 Repulsive Energy Function

We first define a repulsive energy function E: $C \rightarrow R^+$. A posture, $q \in C$, consists of global translation and orientation plus joint angles. The skeleton of every q consists of a set of rigid body segment, or *links*, which is denoted by $l = \{l_1, l_2, ..., l_n\}$. Each link, l_i, is a bone segment with a neighborhood l_i^{nb} consisting of all links that are directly connected to it. The environment is denoted by E_m. The energy function is defined as:

$$E(q, E_m) = E_{int}(q) + E_{ext}(q, E_m),$$
$$E_{int} = \sum_{l_i \in l} \sum_{\substack{l_j \in l, j \neq i, \\ l_j \notin l_i^{nb}}} \frac{1}{dist(l_i, l_j)^d}, \quad (1)$$
$$E_{ext} = \sum_{l_i \in l, O_k \in E_m} \frac{1}{dist(l_i, O_k)^2} \quad (2)$$

where $dist(x, y)$ is the shortest distance between x and y in 3D Euclidean space. We approximate the link geometries with cylinders. d is an integer which controls the fall-off rate between postures in C.

Intuitively, the further body parts are apart from each other, the lower the energy is. All the postures with energy smaller than $+\infty$ do not have penetrations. Further, when the energy is very low, the limbs are very stretched and apart from each other. So linear interpolation of postures with low energy is more likely to be successful. We first formally discuss some properties of the energy function, then explain how we construct EG.

4.2 Properties of the Energy Function

The energy E is composed of two terms: the internal (E_{int}) and external (E_{ext}) terms where E_{int} monitors the distances between different body links while E_{ext} monitors the distances between the links and the environment. There are several important properties of this mapping that meet our requirements mentioned in the beginning of this section:

- Computation efficiency. The energy function can be computed quickly which meets requirement (4).
- Continuity. The mapping is infinitely continuous. So locally a small change in the configuration space C corresponds to a small change in R^+. This satisfies requirement (3).
- Repulsiveness. For the internal energy term, when all the distances between every pair of disjunct body links are maximized, the energy

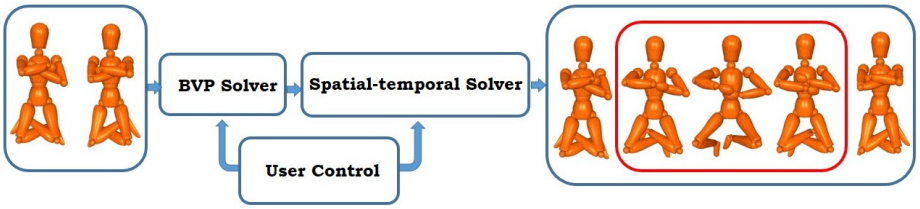


Fig. 2: System Overview. Our system takes two postures as input and generates the motion via two-stage processing. The frames circulated by the red frame are the generated frames.

reaches the lowest value. It is similar for the external energy. A typical low energy configuration is a fully-stretched body in an empty environment (Figure 3 d). This property helps with requirement (2) which will be explained in Section 5.

- Energy barrier. Given a non-penetrating initial configuration, the energy is finite and reaches $+\infty$ when penetrations happen. It means all the valid postures are aggregated on R^+ where $E < +\infty$. It draws a clear boundary between valid and invalid postures, thus meets requirement (1). In addition, any path that starts with a non-intersecting configuration and does not increase the energy will not cause penetrations.
- Controllable gradients. Note that we put a parameter, d, in Equation 1. We define $d \geq 1$, so Equation 1 is actually a family of functions. The higher d is, the larger the energy fall-off rate is. Its function will be explained in Section 5.

4.3 BVP and Energy Graph Construction

With these properties, we form a Boundary Value Problem for motion generation. C is a high-dimensional vector field, Equation 1 assigns a scalar from R^+ to every vector in C and gives us a new space EG. So we let $EG = C \times R^+$. Given two postures P_1, $P_2 \in C$ and their corresponding points $EG(P_1)$, $EG(P_2) \in EG$, we look for a function $f : t \to C$, $t \in [0, 1]$ bounded by $f(0) = P_1$ and $f(1) = P_2$. Directly solving such a BVP requires additional information about f, such as higher-order information similar to the classic Dirichlet Problem. In our problem, we would like to find a f so that the following energy is minimized:

$$E_f(f(t)) = \int_{t=0}^{1} \| E'(f(t)) \|^2 \, df \simeq \sum_{t=0}^{1} \| E'(f(t)) \|^2$$

$$subject\ to$$

$$f(0) = P_1 \quad f(1) = P_2 \ and \ E_f < +\infty \tag{3}$$

Such an f essentially corresponds to the geodesic path between $EG(P_1)$ and $EG(P_2)$ in EG. However, Equation 1 is only injective, not diffeomorphic. So EG is not necessarily a manifold.

In order to understand EG, we show the following observations. Given a projection, $EG(P)$, of a posture P, a key observation is that the energy is high when there are body links in proximity. According to Section 4.2, there are paths in EG starting from $EG(P)$ and decreasing the energy. In C, these paths correspond to expanding posture P into a stretched one P^e. The most stretched posture is associated with the lowest energy and this posture is a *canonical* posture into which other postures can be expanded. A typical canonical posture is shown in Figure 3(d). In addition, some postures with the similar energy values cannot be linearly interpolated. This reveals some information about the structure of EG.

Formally, Equation 1 is a Morse function which maps the manifold C to R^+. So EG can be represented by a Reeb graph [29] as follows:

$E : C \to R^+$ is a continuous function defined on C. Define an equivalent relation $(P_1, E(P_1)) \sim (P_2, E(P_2))$ which holds if and only if:

- $E(P_1) = E(P_2)$ and
- P_1 and P_2 are in the same connected component if there is a path $H \subset C$ passing through P_1 and P_2 so that $\forall P \in H, E(P) = E(P_1) = E(P_2)$.

The Reeb graph is the quotient space of the graph of E in $C \times R^+$ by the equivalent relation. We call this graph the *energy graph*, denoted by EG.

In [29], the height function is used as the Morse function for extracting the Reeb graph of a 2D manifold. Nodes in this graph represent the inner space of the mesh at a certain height. Edges are between nodes if they are directly connected. Branching happens when nodes at the same height are only connected through nodes at a different height level. For example, imagine given a T-pose character mesh, there is a branch at the pelvis extending into both of the thighs. This is because two nodes from two thighs respectively are only connected through the pelvis even if they are at the same height. Likewise, we use the energy function as the Morse function instead to extract the Reeb graph out of C. In this graph (Figure 3), postures that can be transformed to each other without changing the energy are grouped as nodes. Postures form branches when they are only connected via a node at a low energy level. In C, these postures have to be expanded first to meet at a lower energy level.

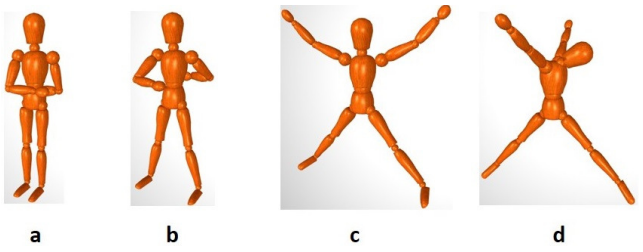


Fig. 3: Posture **a** is gradually expanded to **b**, **c** and then to **d** when decreasing the energy. Posture **d** is a canonical posture.

Then solving Equation 3 can be considered as a problem of finding the shortest path in EG between $EG(P_1)$ and $EG(P_2)$.

5 Energy-based Interpolation for Boundary Value Problem

5.1 Energy-descent interpolation

In this energy graph, there are four canonical postures due to symmetry. One of them is shown in Figure 3(d). As all the joints are fully stretched in these postures, linearly interpolating any two of them will not result in penetrations. We define a set P_c to include all the canonical postures. If we can expand P_1 and P_2 to P_c^1 and P_c^2 where $P_c^1, P_c^2 \in P_c$, we consider that the planning is successful. We search for $f(t) = f_1(t_1) + f_2(t_2)$ with boundary conditions $f_1(0) = P_1$, $f_1(t_c) = P_c^1, f_2(t_c) = P_c^2$ and $f_2(1) = P_2$, where $t_1 \in [0, t_c]$, $t_2 \in [t_c, 1]$, $t_c \in [0, 1]$.

Finding f_1 and f_2 is equivalent to finding the geodesics between $EG(P_1)$ and $EG(P_c^1)$ as well as $EG(P_2)$ and $EG(P_c^2)$ on the graph. Because the path between canonical postures are fixed, E_f is lower-bounded by $E_f^{low} = E_f(f_1) + E_f(f_2)$. Note this is a relaxation of the original BVP and it helps to find a valid motion by a fast local algorithm explained below.

Given a posture, we compute the geodesic to $EG(P_c)$ by tracing the negative gradient flow of the energy iteratively:

$$D = -\alpha \bigtriangledown E \tag{4}$$

where D is the update direction, α is a small positive number and $\bigtriangledown E$ is the normalized energy gradient. In this way, we compute $f_1 = \{P_1^1, P_1^2, ..., P_1^{n-1}, P_c^1\}$ and $f_2 = \{P_c^2, P_2^{m-1}, P_2^{m-2}, ..., P_2^1\}$ respectively, where the superscript is the index of steps. On the graph, this corresponds to monotonically decreasing the energy until the postures become canonical. An example is shown in Figure 3. We emphasize that P_c is not fully connected. Theoretically, it can cause failure when two postures are expanded into two canonical postures that are not connected. However, experimentally we find linear interpolation between canonical postures will not cause penetrations thus can be used. More details are given in Section 5.2.2.

5.2 Relaxation by Linear Interpolation and Safe Zone

5.2.1 Leveraging Linear Interpolation

Although the motion can be generated by energy-descent planning as shown in Figure 3, sometimes the synthesized movements include unnecessary expansion, which is different from natural motions that usually take the shortest path. This is because the energy of a canonical posture, $E(P_{low})$ where $P_{low} \in P_c$ is too low and simply tracing the negative gradient flow is too strict. To address this problem, we employ two strategies: using linear interpolation and increasing the fall-off rate d in Equation 1.

Linear interpolation follows the shortest path in C thus should be applied when the synthesized motion does not cause any penetrations. In each iteration, we check if it will cause penetrations. If it does not, we only use linear interpolation. Otherwise, we project the linear interpolation direction to the null space of the energy gradient so it does not increase the energy. This requires to compute a *safe zone* which is explained later as well as a modification of Equation 4.

We employ the same strategy proposed in [28]. At every step, instead of changing both postures, we update the posture with the higher energy, P_h, towards the other posture, P_l. So D is first initialized by $\frac{P_l - P_h}{\|P_l - P_h\|}$, then Equation 4 is changed into:

$$D = \begin{cases} \alpha D & \text{if } D^T \cdot \bigtriangledown E \leq 0 \\ (I - \bigtriangledown E \cdot \bigtriangledown E^T) D & \text{otherwise} \end{cases} \tag{5}$$

where α is the same parameter as in Equation 4. Equation 5 essentially projects the linear interpolation direction, $\frac{P_l - P_h}{\|P_l - P_h\|}$, so that it does not increase the energy. In this way, unnecessary expansions can be mitigated.

Incorporating linear interpolation into the energy-descent approach results in actively finding a common posture P'_{low} such that $E(P'_{low}) > E(P_{low})$. In the energy graph, $EG(P'_{low})$ is a common parent node that is closer to $EG(P_1)$ and $EG(P_2)$ than $EG(P_c)$, thus reduces the total distance that the two bodies have to travel on the graph to meet each other. The higher $E(P'_{low})$ is, the less the two bodies have to travel, thus decrease E_f. An illustrative example is shown in Figure 4.

Although using a posture with a lower energy as a way point can reduce excessive expansion of the bodies, some unnecessary expansion still happens especially at the peripheral joints. This is because the energy function traces the energy between all pairs of body links and expands all parts almost equally. To tackle this issue, we tune the energy fall-off rate between postures by adjusting d in Equation 1. A

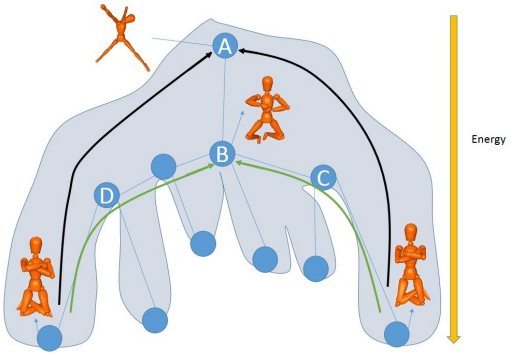


Fig. 4: An illustrative example of the energy graph. The two postures at the bottom are given. Without relaxation, they take the black paths to A. With relaxation, they meet at B by taking the green paths.

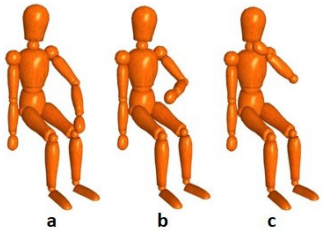


Fig. 5: Two postures (a and c) that can be linearly interpolated at high energy postures without penetrations.

high value makes the energy more dominated by the closest pair of body links, therefore they are expanded more. The user can change the style of the motion by adjusting the value of d. Experimental results of synthesized motions with different ds are shown in Section 7.2.

5.2.2 Safe Zone

The second strategy to relax the BVP is to identify a space in which postures can be linearly interpolated without penetrations. A simple example of such a space is P_c where only canonical postures are included. Actually, there are other postures that can be linearly interpolated but have high energies. A simple example is shown in Figure 5. A linear interpolation between Figure 5(a) and Figure 5(c) resulting in Figure 5(b). It shows for every $P \in C$, there is a neighborhood around P so that linear interpolation between P and any other posture in the neighborhood will not cause penetrations. We call these postures *safe postures* w.r.t. P. Every posture in P_c is a safe posture w.r.t any other postures in P_c. Further, the neighborhood of every posture together constitutes a space S, and we call S the *safe zone*. If we can calculate S, it will be helpful for interpolation.

Formally, we define a relation between safe postures as $P_i \sim P_j$ if P_i can be linearly interpolated to P_j in the joint angle space without penetrations. Then a subspace $S_p \subset C$ is *safe* with respect to posture P iff :

- $\forall P \in S_p, P \sim P_t$ where $\forall P_t, P_t \in S_p \setminus P$

The boundary of S_p is $\partial S_p \subset C$. So we have the safe zone defined as:

$$S_c = \cup S_i \tag{6}$$

with its boundary $\partial S_c = \cup \partial S_i$. Intuitively, there is a subspace of S_p in C around every posture P. For example, P_a, P_b and P_c in Figure 5 are in $S_a \cap S_b \cap S_c$ so any two of them can be linearly interpolated. Note the relation is reflective, symmetric but not transitive. So only postures within the same subspace of S can be linearly interpolated. Figure 2 shows two representative postures. In the output, every two consecutive postures are in the same subspace, but clearly the first and last posture are not.

Safe zone allows us to further shorten the motion path. Previously, linear interpolation is only applied when it does not increase the energy. Introducing S changes the BVP in the following way. Given two postures P_1 and P_2, they can be expanded into P_1^e and P_2^e where $P_1^e, P_2^e \in \partial S_i$. Here S_i can be P_c or any other subspace of S_c. The worst case is when $S_i = P_c$ which means P_1 and P_2 are fully expanded. However, this can happen way before they meet each other by the method explained in the last section. When it happens, linear interpolation between P_1^e and P_2^e further shortens the path. The solution thus is changed to: $f_1 = \{P_1^1, P_1^2, ..., P_1^{n-1}, P_1^e\}$, $f_2 = \{P_2^e, P_2^{m-1}, P_2^{m-2}, ..., P_2^e\}$ and $f = f_1 + LinearInterpolation(P_1^e, P_2^e) + f_2$.

Then the question comes down to how to recognize if two postures are in the same subspace of S. Accurately calculating S_c is hard. However, we can implicitly approximate S_c by a method based on tangle energy [13]. In [13], they use Gauss Linking Integral (GLI) to compute the tangle energy between two linkages. GLI is the average number of crossings when viewing the tangle from all directions divided by two. Here we consider the GLI value between two directed curves. Simple examples of two directed curves with different GLI values are given in Figure 6. When the GLI value is equal to 1, the two curves twist around one another once. If it is over 0.5 but smaller than 1, it means two curves are tangled and cannot be separated by a plane without cutting either one of them. If it is smaller than 0.5, it means two curves are untangled. Switching two curves only changes the sign of the value. GLI can be calculated by:

$$GLI(g_1, g_2) = \frac{1}{4\pi} \int_{g_1} \int_{g_2} \frac{\delta g_1 \times \delta g_2 \cdot (g_1 - g_2)}{\| g_1 - g_2 \|^3} \tag{7}$$

where g_1 and g_2 are two directed curves, $\times$ and $\cdot$ are the cross and dot operator, δg_1 and δg_2 are two infinitesimal segments of g_1 and g_2.

Approximating curves by linkages, we can calculate GLI values for each pair of links between two linkages. Figure 7 shows such an example. The GLI value

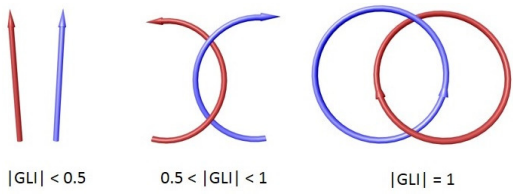


Fig. 6: Gauss Linking Integral between two linkages.

$$\begin{bmatrix} 0.4 & 0.1 & 0.0 \\ 0.1 & 0.4 & 0.1 \\ 0.0 & 0.1 & 0.4 \\ 0.0 & 0.0 & 0.1 \end{bmatrix}$$

Fig. 7: The GLI Matrix of two linkages.

between two links has an analytical solution proposed in [30]. By integration over the matrix, we get the total GLI value for the two linkages.

We use GLI due to the following reasons. Firstly, our tree structure skeleton can be easily decomposed into linkages for tangle energy calculation. Secondly, the encoding of spatial relations between linkages is continuous and can be computed quickly. It does not only give information about how close two individual links are, but also how one linkage as a whole is placed around another, thus can discover the potential penetrations between body limbs well.

Since directed curves can be approximated by directed linkages, we decompose the skeleton into 5 linkages shown in Figure 8. Given a posture P, we can calculate a GLI matrix M_p whose entries are GLI values between linkages. This GLI matrix is similar to the one in Figure 7 except that its rows and columns are the body linkages shown in Figure 8 and the diagonal only contains zeros. Given P_1 and P_2, we can calculate M_{P_1} and M_{P_2}. Take the GLI value between $LeftLeg$ and $RightLeg$ in Figure 9 for example. The absolute difference between posture a and b, and

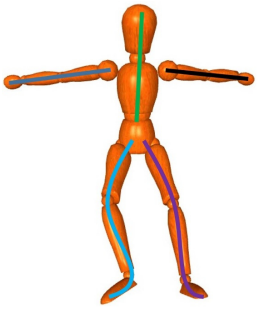


Fig. 8: The skeleton is decomposed into 5 linkages shown in five colors. They are $LeftArm$, $RightArm$, $Torso$, $LeftLeg$ and $RightLeg$

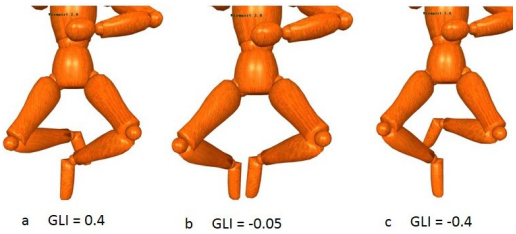


Fig. 9: GLI values between $LeftLeg$ and $RightLeg$

between b and c are small. So linear interpolation is less likely to cause penetrations. While the difference is high between posture a and c so linear interpolation will cause penetrations.

We define our safe zone energy barrier as follow:

$$E_{gli}(P_1, P_2) = maxAbs(M_{P_1} - M_{P_2}) \tag{8}$$

where $maxAbs(A)$ returns the largest absolute value in matrix A. We have a threshold which is empirically set to 0.3. If $E_{gli}(P_1, P_2) < 0.3$, then there is no impending penetrations between any linkages when we linearly interpolate the two postures. Note this is a conservative approximation of ∂S_c, but it works well in our experiments.

In each iteration, if both postures are safe w.r.t each other, they are linearly interpolated unless there are external obstacles. Overall, the whole interpolation algorithm is given in Algorithm 1.

6 Constrained Planning for User Control

Algorithm 1 is a very basic version of the energy-based planning. To further enhance our algorithm and allow more user control over generated motions, we establish a two-stage framework. In the first stage, we generate penetration-free motions with constraints. Then in the second stage, we enforce constraints by projecting the motion onto a new constraint manifold. The user control is implemented as positional constraints in this framework.

6.1 Constrained Planning

In the first stage, we incorporate constraints into the energy interpolation in the same way as [28]. Without loss of the generality, constraints can be linearized as:

$$K\Delta q = C \tag{9}$$

where K is the Jacobian matrix, $\Delta q = q_{i+1} - q_i$ in ith iteration and C is the deviation of the current value from the target one. Given a Δq, we project it onto the null space of the constraint:

$$\begin{aligned} \Delta q' &= \Delta q - K^T l \\ l &= (KK^T)^{-1}(K\Delta q + \gamma\sigma) \end{aligned} \tag{10}$$

where γ is set to 0.01 for all our experiments and $\sigma = K\triangle q - C$. However, this only ensures that $\Delta q'$

Algorithm 1 Energy-based Interpolation

Require: Two postures P_1 and P_2, the environment O, Parameters α.
Require: Motion sequence $M_1 = \emptyset$ and $M_2 = \emptyset$
repeat
  Calculate energies E_1 and E_2 for P_1 and P_2 by Equation 1.
  if $E_1 \geq E_2$ **then**
    $P_h \leftarrow P_1$, $P_l \leftarrow P_2$, $\bigtriangledown E \leftarrow \bigtriangledown E_1$
  else
    $P_h \leftarrow P_2$, $P_l \leftarrow P_1$, $\bigtriangledown E \leftarrow \bigtriangledown E_2$
  end if
  $D \leftarrow \frac{P_l - P_h}{\|P_l - P_h\|}$
  Calculate maximal GLI value difference E_{gli} for P_1 and P_2 by Equation 7.
  if $E_{gli} < 0.3$ and $O = \emptyset$ **then**
    $P_h \leftarrow P_h + \alpha D$
  else if $D^T \cdot \bigtriangledown E \leq 0$ **then**
    $D \leftarrow \alpha D$
  else
    $D \leftarrow (I - \bigtriangledown E \cdot \bigtriangledown E^T) D$
  end if
  $P_h \leftarrow P_h + \alpha D$
  if $E_1 \geq E_2$ **then**
    $P_1 \leftarrow P_h$
    add P_1 to M_1
  else
    $P_2 \leftarrow P_h$
    add P_2 to M_2
  end if
until P_1 is similar to P_2
return M_1 + $reverse(M_2)$

satisfies the constraints and sometimes it can increase the energy, $\Delta q\prime \cdot \bigtriangledown E > 0$. When this happens, we need to include an energy term into the constraints:

$$\begin{aligned} \Delta q'' &= \Delta q - K^T l_e \\ l_e &= (JJ^T)^{-1}(J\Delta q' + \gamma\epsilon) \\ J &= \begin{bmatrix} K \\ \bigtriangledown E \end{bmatrix}, \epsilon = \begin{bmatrix} \sigma \\ \omega/\gamma \end{bmatrix} \end{aligned} \tag{11}$$

where ω is set to -0.01 in all our experiments. After projection, $\Delta q''$ is in the null space of the constraints and it decreases the energy. J can be under-ranked, full-ranked or over-ranked. Essentially, this projection satisfies the constraints in a least-square-error sense thus it handles the constraints as soft constraints. The enhanced version of Algorithm 1 is shown in Algorithm 2. Various constraints used in our experiments are as follows.

6.1.1 Positional Constraints

Positional constraints for joints are formed as:

$$J_{pos}\Delta q = C_{tar} - C_{cur} \tag{12}$$

Algorithm 2 Constrained Planning

Require: two postures P_1 and P_2, the environment O, Parameters α.
Require: Motion sequence $M_1 = \emptyset$ and $M_2 = \emptyset$
repeat
  Calculate energies E_1 and E_2 for P_1 and P_2 by Equation 1.
  if $E_1 \geq E_2$ **then**
    $P_h \leftarrow P_1$, $P_l \leftarrow P_2$, $\bigtriangledown E \leftarrow \bigtriangledown E_1$
  else
    $P_h \leftarrow P_2$, $P_l \leftarrow P_1$, $\bigtriangledown E \leftarrow \bigtriangledown E_2$
  end if
  $D \leftarrow \frac{P_l - P_h}{\|P_l - P_h\|}$
  Evaluate constraints.
  if Constraints violated **then**
    Form constraints by Equation 9
    Solver for $\Delta q'$ by Equation 10
    if ${\Delta q'}^T \cdot \bigtriangledown E \leq 0$ **then**
      $D \leftarrow \Delta q'$
    else
      Project $\Delta q\prime$ and solver for $\Delta q''$ by Equation 11
      $D \leftarrow \Delta q''$
    end if
  else
    Calculate maximal GLI value difference E_{gli} for P_1 and P_2 by Equation 7.
    if $E_{gli} < 0.3$ and $O = \emptyset$ **then**
      $D \leftarrow \alpha D$
    else
      $D \leftarrow (I - \bigtriangledown E \cdot \bigtriangledown E^T) D$
    end if
  end if
  $P_h \leftarrow P_h + D$
  if $E_1 \geq E_2$ **then**
    $P_1 \leftarrow P_h$
    add P_1 to M_1
  else
    $P_2 \leftarrow P_h$
    add P_2 to M_2
  end if
until P_1 is similar to P_2
return M_1 + $reverse(M_2)$

where J_{pos} is the Jacobian matrix. C_{tar} and C_{cur} are the target and current positions of the joint of interest.

6.1.2 Joint Limits

Joint angles are bounded by $q_{min} \leq q \leq q_{max}$. When these limits are violated, the following constraint is formed:

$$J_{joint}\Delta q = C_{bound} - C_{cur} \tag{13}$$

where J_{joint} is the Jacobian. C_{bound} and C_{cur} are the bound value and current joint angle.

6.2 Constraints Enforcement

After Algorithm 2, constraints can be violated by small amounts. We build a second stage to enforce the constraints and incorporate additional user control. These constraints and user control signals form a constraint manifold E_{con}. We project the motion onto E_{con} to get a solution motion f by minimizing:

$$\underset{f}{\operatorname{argmin}} \parallel E_{con}(f) \parallel^2 \tag{14}$$

There are several concerns regarding this motion adaptation. Firstly, the original motion should be kept as much as possible. This is to prevent new penetrations as well as keeping the motion style. Secondly, user control must be enforced. Lastly, physical constraints should be considered too. We introduce Laplacian energy, momentum preservation and positional constraints into our iterative optimization. We first explain every single energy term then give details of the optimization.

6.2.1 Laplacian Energy

The motion projection can cause new penetrations or change the motion style. The first constraint we need to enforce is the Laplacian energy of the motion. Laplacian energy is widely used in many areas. For motion adaptation, Ho and Komura [31] used it for modeling the interactions between characters. In their work, the space between characters involved in close interactions is encoded by the tetrahedralization of joints of the characters. When morphing the motion, the Laplacian energy is maintained so the spatial relations are kept. Visually, it preserves the motion style and prevents penetrations when adapting the motion. We employ the same idea for modeling interactions between different body parts for a single character. A tetrahedralization of joint positions gives a volumetric mesh. Let $P_v(q) = \{p_1, p_2, ..., p_n\}$ be the vertices of the mesh. The Laplacian coordinates of each joint can be computed as:

$$L(p_i) = p_i - \sum_{j=1, j\in N_i}^{m} w_j p_j \tag{15}$$

where N_i is the set of vertices in p_i's *1-ring* neighborhood. w_j is the normalized weights that are inversely proportional to the distance between p_j and p_i.

The relation between the joint positions and the skeleton degrees of freedom can be linearized as $\Delta P_v = J_p \Delta q$. Let P_v^t, q^t and J_p^t be the vector of the vertex positions of the volumetric mesh, the vector of the skeleton Dofs and the Jacobian matrix at frame t respectively. We define a deformation energy of this volumetric mesh as:

$$E_{lap} = \sum_{t=1}^{n} \frac{1}{2} \parallel L(J_p^t \Delta q^t + P_v^t) - \Gamma^t \parallel^2 \tag{16}$$

where Γ^t is the Laplacian coordinates at frame t. Preserving this energy preserves the spatial relations between different body parts for each frame.

6.2.2 Momentum Preservation

For synthesizing highly dynamic motions, physically-based constraints should also be considered. Our framework can incorporate physically-based constraints, such as preserving the linear and angular momentum. As an example, we show how the angular momentum preservation can be done.

We compute the angular momentum of the Center of Mass (CoM), $P_{CoM}^t(q)$, with respect to the rotation axis. By linearization, we have $\Delta P_{CoM} = J_p^c \Delta q$. At frame t, we define the angular momentum as $A_{CoM}^t(P_{CoM}^t(q^t), P_{CoM}^{t+1}(q^{t+1}))$ and define a momentum deviation energy:

$$E_{at} = \sum_{t=2}^{n-1} \frac{1}{2} \parallel A_{CoM}^t - A_{CoM}^{t+1} \parallel^2$$
$$A_{CoM}^t = I^t \cdot w^t \tag{17}$$

where I^t is the inertia tensor and w^t is the angular velocity at frame t. Linear momentum can be preserved in a similar fashion:

$$E_{lm} = \sum_{t=2}^{n-1} \frac{1}{2} \parallel L_{CoM}^t - L_{CoM}^{t+1} \parallel^2$$
$$L_{CoM}^t = m \cdot v_{CoM}^t \tag{18}$$

where m is the mass and v_{CoM}^t is the velocity of the CoM at frame t.

6.2.3 Iterative Solver

Given the initial motion $m = \{P_1, P_2, ..., P_n\}$, joint limits and positional constraints are hard constraints formed as $H_i \Delta q_i - h_i = 0$ at frame i (see Equation 12 and Equation 13). So far, we have constructed a constraint manifold $E_{cons} = E_{lap} + E_{at} + E_{lm} + \sum_i (H_i \Delta q_i - h_i)$. We iteratively project m onto E_{cons}. In each iteration, projecting m onto E_{cons} is equivalent to solving the following quadratic programming problem:

$$\underset{\Delta q}{\operatorname{argmin}} \frac{1}{2} \Delta q^T M^T M \Delta q + w_{lap} E_{lap} + w_{am} E_{at}$$
$$+ w_{lm} E_{lm} + \sum_i \lambda_i (H_i \Delta q_i - h_i) \tag{19}$$

where M, w_{lap}, w_{am} and w_{lm} are a weight matrix and three real weights respectively. λ_i is the Lagrange multiplier. For the sake of simplicity, we omit the linear momentum below. Finding the stationary point is equivalent to solving the following linear system:

$$\begin{bmatrix} M^T M + J_{lap}^T J_{lap} + w_{am} J_{am}^T J_{am}, & H^T \\ H, & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta q \\ \lambda \end{bmatrix}$$
$$= \begin{bmatrix} J_{lap}\Gamma + J_{am}\Phi \\ h \end{bmatrix} \tag{20}$$

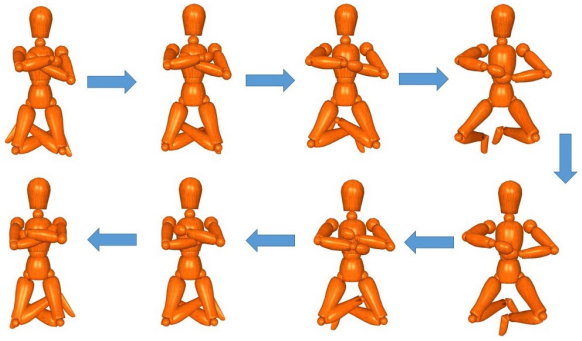


Fig. 10: Yoga posture interpolation. With d = 5 and safe zone

where M is identity matrix. λ is vector including all the multipliers. Γ includes all the Γ_t in Equation 16 and Φ is a vector of angular momentum differences $\{A^1_{CoM} - A^2_{CoM}, A^2_{CoM} - A^3_{CoM}, ..., A^{n-1}_{CoM} - A^n_{CoM}\}$. H is a block matrix stacking Jacobian matrices of Laplacian energy and angular momentum on its diagonal. J_{lap} and J_{am} are the block Jacobian matrices of E_{lap} and E_{am}.

We run the solver until constraints are satisfied within a threshold or the maximum iteration has been reached. This solver is similar to the spatial-temporal solver used in [4]. Although in our experiments we do not have timing constraints, incorporating them is straightforward.

7 Experimental Results

In this section, we first show motion planning between complex postures. Then we show how the motion generation is done for interactions with the environment. Next, we evaluate how the fall-off rate and safe zone affect the motion generation. Finally, we perform evaluations of different factors in our framework and comparisons with existing methods. The readers are referred to the supplementary video for details.

Our character has 63 degrees of freedom and only the first and last frame are given in each experiment unless specified otherwise.

7.1 Motion planning

Complex motions Two examples are given to show complex path-planning. The first one is generated from two yoga postures with both arms and legs entwined in opposite directions. The result is shown in Figure 10.

The second example is a synthesized motion of a contortion in Figure 11. Contortion is an unusual form of physical display which involves the dramatic bending and flexing of the human body. In this example, our system is fed with a common contortion posture (the last frame in Figure 11). The character starts lying face-down and ends up in a complete

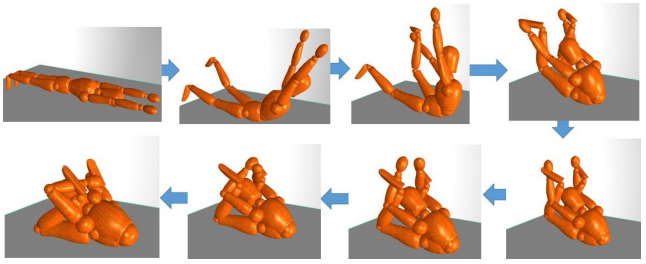


Fig. 11: Contortion motion. With d = 5 and safe zone

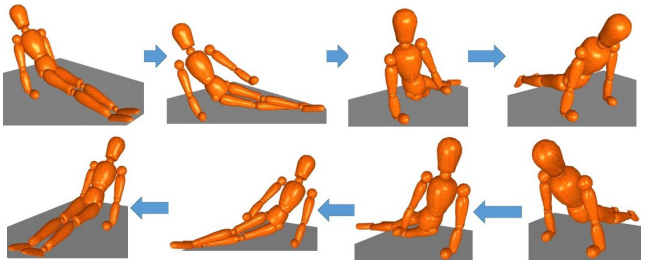


Fig. 12: Thomas flare. With d = 5 and safe zone

folded configuration. The final configuration is very sophisticated due to the 7 tangles made by body linkages. It is difficult for methods proposed in [12], [13] to deal with such a situation. Because they model and interpolate each tangle independently, they can suffer from local minima with a group of entwined tangles.

Movements involving interaction with the environment To show how our framework can easily incorporate user control, we show how a Thomas flare can be synthesized from two pairs of postures, shown as the first and last posture in each row of Figure 12. It is done in two phases. We first apply Algorithm 2 on every pair of postures. To make it look realistic, we add a floor plane under the character. We also fix the left hand when planning between posture 1 and posture 4 in the first row of Figure 12, and the right hand for posture 1 and posture 4 in second row of Figure 12. In both scenario, the non-supporting hand is automatically raised to let the legs pass.

In the second phase of motion synthesis, we enforce Laplacian energy and angular momentum constraints as well as positional constraints for hands then optimize the motion by Equation 19. Since the last postures of the two motions are the same, the two motions can be easily concatenated into a whole motion.

7.2 Evaluations and Comparisons

Fall-off rate control and safe zone Firstly, we use the Leg Cross example here to compare our method with the one in [23]. We show how the planning works with a fixed fall-off rate ($d = 2$ in Equation 1). The input consists of two postures with legs crossed

in opposite directions, shown in Figure 13 (The first and last frame). During planning, only joint limits are considered. The generated motion is shown in Figure 13(a).

Although the motion can be planned successfully, the arms are expanded excessively. This is caused by the reason mentioned in Section 5.2.1. In this example, the two legs tend to get closer initially because the solution follows the direction of the linear interpolation. The arms also have to move away from the body so that the total energy decreases. This problem is mitigated by increasing d as well as adding the safe zone interpolation, as shown in Figure 13(b). We show another series of Yoga motions synthesized with different d values and safe zone in the supplementary video.

Performance Our algorithm is implemented on a machine with a Intel 2.40GHz i7-2760QM CPU, 8GB Ram and Windows 8 64-bit. We use boost 1.47 [32], UMFPack [33] and GotoBlas [34] for interpolation and optimization. Table 1 gives detailed performances of different motion planning.

	Arm Fold	Leg Cross	Yoga	Contortion	Thomas Flare
Stage 1	11.415	41.874	85.108	224.923	23.726
Stage 1*	10.722	39.671	81.258	138.106	n/a
Stage 2	n/a	n/a	n/a	n/a	142.45

TABLE 1: All units are in seconds. Stage 1 and Stage 1* are the first stage without and with safe zone detection respectively. *n/a* means not available because we do not need to do it.

Table 1 shows how the safe zone detection significantly improves the performance, especially if there is a large safe zone detected during the motion planning stage (see Contortion in Table 1).

The current performance bottle neck is the stiffness of the energy function. Although each iteration can be computed quickly, computing the whole motion requires many iterations. This is because small steps are used for complex postures, similar to the problem of advancing a stiff ODE system. Adaptive step sizes might mitigate the problem. Another possibility is to use a backward Euler scheme but this involves solving another linear equation in each iteration.

Next, we show detailed timing information of different energy term evaluations in each iteration at stage 2 when planning a Thomas flare (half cycle). We show the results of applying optimization with and without Laplacian constraints. We set $w_{lap} = 0.02$, $w_{am} = 0.2$. And the maximum iteration is 30.

As shown in Table 2, the evaluation of Laplacian energy takes quite a big fraction of the computational time. However, since there is no dependence between frames for Laplacians, this computation can be boosted by parallelization. It is the same for joint limits and positional constraints. In addition, Laplacian energy is crucial in preserving motion validity and style during optimization.

	Lap	JL	Pos	AM	Solving
1	1.34	0.013	0.048	0.653	2.278
2	0	0.013	0.048	0.653	0.217

TABLE 2: Evaluation timing for different energy terms. All units are in seconds. 1: With Laplacian energy. 2: Without Laplacian energy. Lap: Laplacian energy. JL: Joint limits. Pos: Positional constraints. AM: Angular momentum.

Comparison We compare our method with RRT and PRM variants. Due to the stochasticity of RRT and PRM variants, we do 10 runs for each experiment and set it as a failure if no results are computed within 5 minutes. For implementations, we use the Open Motion Planning Library (OMPL) [35].

To cope with the irregular movements, we apply the smoothing method in [17] to all motions except ours. This method iteratively smooths the trajectory by connecting two frames using linear interpolation. If successful, it replaces the original trajectory with linear interpolation, then starts the next iteration. The algorithm runs until no further linear interpolation is possible. We chose a set of values (e.g. interpolation resolution and self-collision) conservatively for all experiments in order to produce movements without any self-collisions and pass-throughs. A boxplot that compare the performance are shown in Figure 14.

Our method outperforms the other methods. For the first three experiments, sometimes the best results of RRTConnect, PRM and PRM* are still comparable to ours but worse in general. Although they can find a solution much faster than our method for simple scenarios, the smoothing process takes a long time. In addition, our method has a higher success rate (especially the contortion example where all other methods failed). We also compare the quality of the motions synthesized out of similar amount of computational time. Although the computational time is comparable and the success rates are similar, the quality of the movements by other methods are far worse than those by ours. The readers are referred to the supplementary video for details.

8 Discussions and Limitations

Relation to Previous Work Wang and Komura [23] also use a repulsive energy function to do motion planning between distant postures. Our energy function is an augmented version of theirs. The augmentation involves adding a new parameter to control the energy fall-off rate and a new energy term to model the energy between the character and the environment. We also include the global translation and orientation so that more complex motions with positional constraints can be synthesized. These changes

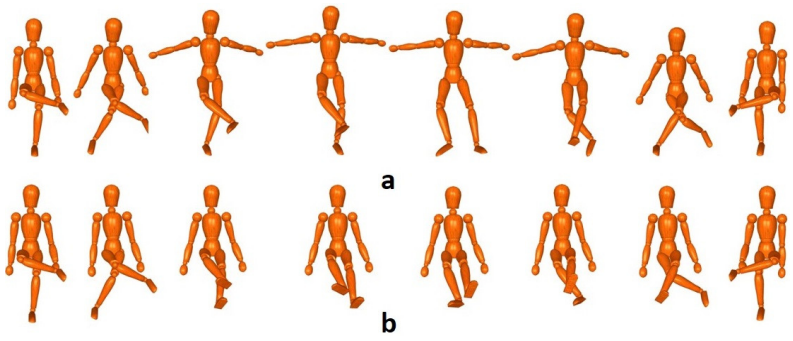


Fig. 13: Leg cross interpolation. The first and last postures are given. d=2, no safe zone for a. In b, d = 5 and safe zone is computed.

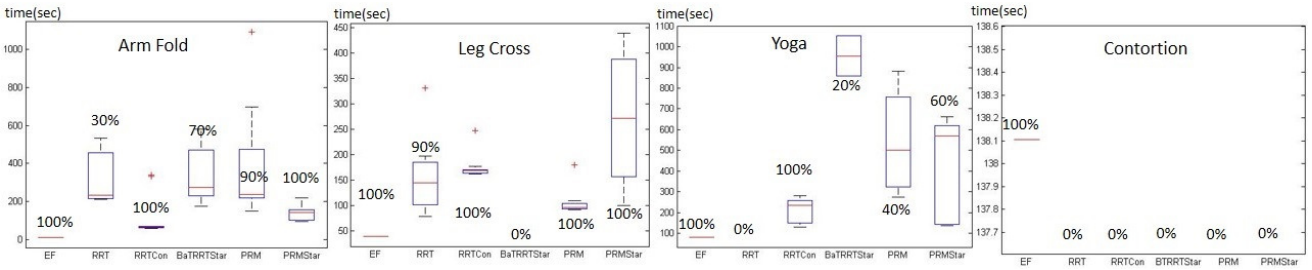


Fig. 14: In each figure from the left to the right: Our method. RRT [1]. RRT Connect [1]. Ball Tree RRT* [36], [37]. PRM [2]. PRM* [37]. Success rates are shown as percentages for each method. "+"s are outliers.

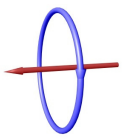


Fig. 15: An example where multi-linkages falling into local minimum. The red straight line goes through the center of the blue ring and is perpendicular to the plane spanned by the blue ring.

not only enable us to synthesize more complex motions but also control the motion style.

Simplicity of Human Skeleton Our method is mainly designed for human skeleton, because it cannot form complex or dead-locked knots due to the simplicity of its topology and geometry. The success of our method heavily depends on the expandability of the skeleton. For linkages in general, this is not the case. Firstly, dead locks are disconnected components in the configuration space. No method can jump from one component to another without penetrations, or changes of the geometry or topology of the structure (e.g. shortening links or breaking connections). Secondly, the energy of multiple linkages in general can easily fall into local minima. A simple example is shown in Figure 15. No matter how we move the bar w.r.t. the ring, the energy does not decrease. Thirdly, even if there is no dead lock and the linkage in interest is a single tree, there is still no guarantee it can be expanded thus cannot be handled by our method; it is proven that 3D linkage expansion is not guaranteed in general [38], [39].

Possible Local Minima As mentioned before, due to the simplicity of the human skeleton, we do not observe failures even for very complex postures in our experiments. However, there are certain parameter settings that can cause local minima.

In practice, when d in Equation 1 is very large, local minima can happen. This is because when d is very large, the energy is solely dominated by the closest pair of body links. We use a simple heuristic to detect possible local minimum. We use a small energy window and monitor the postures during motion planning. The energy window starts from the higher energy value of the two initial postures. In each iteration, we examine if the lower energy value of the two postures falls out of this window. If it does, we slide the window to contain it. If two postures stay in the same energy window for a certain number of iterations (50 in our experiment), we regard it as a local minimum. Nevertheless, d is set between 2 and 5 in all our experiments and no local minima were observed.

Subspace of the Solution Space Our method explores only a subspace of the valid motion space. This is due to the constraint that the energy cannot be increased. In Figure 5, linear interpolation brings the left hand into the air (Figure 5 (b)). Imagine the user wants the left hand to slide over the surface of the body, see Figure 16. It is a valid motion between the

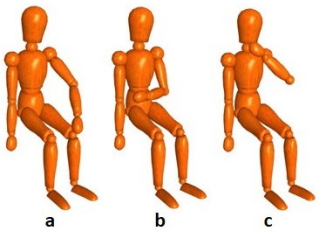


Fig. 16: A modified version of Figure 5, with a different intermediate posture of very high energy.

two postures at a very high energy level (Figure 16 (b)) and linear interpolation cannot generate this motion. This motion is not explored in our framework. Such a case can be handled by introducing proper control (adding extra constraints) in the second stage.

Limited Interaction with the Environment Although we allow the user to add environmental objects, we assume there are only simple interactions between the body and the environment. Firstly, the planning needs to be done in a relatively open area. Highly confined spaces can cause local minima. Secondly the contact should be invariant or subject to small changes. A complex motion sequence where contacts change greatly cannot be directly synthesized by our method. However, it is easy for the user to divide the whole sequence into contact invariant segments, similar to what we did in the Thomas flare demo.

Motion Naturalness The movements synthesized by our system can sometimes appear unnatural due to the monotonic velocity profile and the simultaneous movements of the limbs. Humans tend to move body parts sequentially for some motions such as dressing, and also the velocity profile follows a bell-shape trajectory [17]. Due to the simplicity of our approach, such effects are not considered. However, those effects can be easily added by passing the movements through an intermediate layer without increasing the complexity of the system.

9 Conclusion

In this paper, we present a framework to solve motion planning between two postures as a Boundary Value Problem. We assume that the solution is a bounded continuous function on an energy graph. We propose an fast constrained bidirectional motion planning method. To further incorporate user control, we project the motion function to a constraint manifold.

10 Future Work

We use geometric primitives such as cylinders and planes for approximating the body and the environment. Because our energy function is calculated by the shortest distance between geometries, the current approximation facilitates the computation. In the future, we would like to explore methods that handle more complicated geometries.

In addition, we plan to incorporate more complex, close interactions between the character and objects or the environment, such as grabbing objects or getting into a car. This requires updating the energy representation and designing a better motion planning algorithm. One possible solution is to expand the environment together with the character in a similar fashion as discussed in this paper.

Another good direction is to compare our generated motion with motion capture data and motion made by animators. By learning features and relations between these three groups of motions, we can learn a model of natural motion on the energy graph and help animators to synthesize motions biased towards natural motions.

Also, another direction to explore is planning the optimal sequence of movements by different body parts. Currently, the motion planning is done for all the body parts at the same time. In many situations, not all body parts move together. A proper sequencing may help to make the motion more realistic. For instance, in the yoga example in Figure 10, we can first interpolate the legs then the arms.

Finally, although local minima exist for free folding/unfolding of 3D trees in general, we intend to look into humanoid skeletons for theoretical proof of the applicability of our method. We would like also to apply our work to robotics for controlling humanoids or robotic arms for complex path-planning problems.

Acknowledgments

The authors thank the anonymous reviewers. This work is partially supported by grants from EPSRC (EP/H012338/1), EU FP7 TOMSY, HKBU Science Faculty Research Grants (FRG1/12-13/055 and FRG2/12-13/078) and the Hong Kong RGC GRF (No. 210813).

References

[1] S. M. LaValle and J. J. Kuffner, "Rapidly-exploring random trees: Progress and prospects," *Robotics: The Algorithmic Perspective. 4th Int'l Workshop on the Algorithmic Foundations of Robotics*, 2001.

[2] L. Kavraki, P. Svestka, J. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," Stanford University, Stanford, CA, USA, Tech. Rep., 1994.

[3] R. E. Bellman, *Dynamic Programming*. Courier Dover Publications, Mar. 2003.

[4] C. K. Liu, A. Hertzmann, and Z. Popovic, "Composition of complex optimal multi-character motions," *ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, pp. 215–222, 2006.

[5] J. Lee and K. H. Lee, "Precomputing avatar behavior from human motion data," *Proceedings of 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, p. 79–87, 2004.

[6] A. Treuille, Y. Lee, and Z. Popovic', "Near-optimal character animation with continuous control," *ACM Transactions on Graphics*, vol. 26, no. 3, pp. 7:1–7:7, 2007.

[7] H. P. H. Shum, T. Komura, M. Shiraishi, and S. Yamazaki, "Interaction patches for multi-character animation," *ACM Transactions on Graphics*, vol. 27, no. 5, 2008.

[8] M. Kallmann, A. Aubel, T. Abaci, and D. Thalmann, "Planning collision-free reaching motions for interactive object manipulation and grasping," in *ACM SIGGRAPH 2008 Classes*, ser. SIGGRAPH '08. New York, NY, USA: ACM, 2008, p. 58:1–58:11. [Online]. Available: http://doi.acm.org/10.1145/1401132.1401209

[9] J. Pan, L. Zhang, M. C. Lin, and D. Manocha, "A hybrid approach for simulating human motion in constrained environments," *Computer Animation and Virtual Worlds*, vol. 21, no. 3-4, p. 137–149, 2010. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1002/cav.365/abstract

[10] H. Wang and T. Komura, "Manipulation of flexible objects by geodesic control," *Computer Graphics Forum*, vol. 31, no. 2, 2012.

[11] H. Wang, K. A. Sidorov, P. Sandilands, and T. Komura, "Harmonic parameterization by electrostatics," *ACM Trans. Graph.*, vol. 32, no. 5, p. 155:1–155:12, Oct. 2013. [Online]. Available: http://doi.acm.org/10.1145/2503177

[12] E. S. L. Ho and T. Komura, "Character motion synthesis by topology coordinates," *Computer Graphics Forum (Proceedings of Eurographics 2009)*, vol. 28, no. 2, 2009.

[13] E. Ho and T. Komura, "Indexing and retrieving motions of characters in close contact," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 3, p. 481–492, 2009.

[14] Y. Hwang and N. Ahuja, "A potential field approach to path planning," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 1, pp. 23–32, 1992.

[15] J. Kuffner, K. Nishiwaki, S. Kagami, Y. Kuniyoshi, M. Inaba, and H. Inoue, "Self-collision detection and prevention for humanoid robots," in *IEEE International Conference on Robotics and Automation, 2002. Proceedings. ICRA '02*, vol. 3, 2002, pp. 2265–2270.

[16] H. Sugiura, M. Gienger, H. Janssen, and C. Goerick, "Real-time collision avoidance with whole body motion control for humanoid robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007. IROS 2007*, 2007, pp. 2053–2058.

[17] K. Yamane, J. Kuffner, and J. K. Hodgins, "Synthesizing animations of human manipulation tasks," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 532–539, 2004.

[18] Y. Hirano, K. Kitahama, and S. Yoshizawa, "Image-based object recognition and dexterous Hand/Arm motion planning u sing RRTs for grasping in cluttered scene," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.

[19] D. Berenso, R. Diankovage, K. Nishiwaki, S. K. agami, and J. Kuffner, "Grasp planning in complex scenes," *IEEE/RAS Humanoids*, 2007.

[20] L. Zhang, J. Pan, and D. Manocha, "Motion planning of human-like robots using constrained coordination," in *9th IEEE-RAS International Conference on Humanoid Robots, 2009. Humanoids 2009*, 2009, pp. 188–195.

[21] E. S. L. Ho and T. Komura, "Planning tangling motions for humanoids," in *Proceedings of Humanoids 2007*, 2007.

[22] A. M. Ladd and L. E. Kavraki, "Motion planning for knot untangling," *Int. J. of Robotics Research*, vol. 23, p. 797–808, 2002.

[23] H. Wang and T. Komura, "Energy-based pose unfolding and interpolation for 3D articulated characters," in *Motion in Games*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, vol. 7060, p. 110–119. [Online]. Available: http://www.springerlink.com/content/41613nh014697048/

[24] E. D. Demaine and J. O'Rourke, *Geometric Folding Algorithms: Linkages, Origami, Polyhedra*, reprint ed. New York, NY, USA: Cambridge University Press, 2008.

[25] J. H. Cantarella and H. Johnston, "Nontrivial embeddings of polygonal intervals and unknots in 3 -space," *Journal of Knot Theory and Its Ramifications*, vol. 7, no. 8, pp. 1027–1039, 1998.

[26] R. Connelly, E. D. Demaine, and G. Rote, "Straightening polygonal arcs and convexifying polygonal cycles," in *Discrete and Computational Geometry*, 2000, p. 432–442.

[27] J. H. Cantarella, E. D. Demaine, H. N. Iben, and J. F. O'Brien, "An energy-driven approach to linkage unfolding," in *Proceedings of the 20th Annual Symposium on Computational Geometry*, Brooklyn, New York, Jun. 2004. [Online]. Available: http://graphics.cs.berkeley.edu/papers/Cantarella-AED-2004-06/

[28] H. N. Iben, J. F. O'Brien, and E. D. Demaine, "Refolding planar polygons," *Discrete and Computational Geometry*, vol. 41, no. 3, p. 444–460, Apr. 2009. [Online]. Available: http://graphics.cs.berkeley.edu/papers/Iben-RPP-2009-04/

[29] Y. Shinagawa, T. L. Kunii, and Y. L. Kergosien, "Surface coding based on morse theory," *IEEE Computer Graphics and Applications*, vol. 11, no. 5, p. 66–78, Sep. 1991.

[30] M. Levitt, "Protein folding by restrained energy minimization and molecular dynamics," *J. Mol. Biol*, vol. 170, p. 723–764, 1983.

[31] E. S. L. Ho, T. Komura, and C.-L. Tai, "Spatial relationship preserving character motion adaptation," *ACM Transactions on Graphics*, vol. 29, no. 4, p. 1–8, 2010.

[32] Boost, "BOOST c++ libraries," 2000. [Online]. Available: http://www.boost.org

[33] T. A. Davis, "Algorithm 832: UMFPACK, an unsymmetric-pattern multifrontal method," *ACM Transactions on Mathematical Software*, vol. 30, no. 2 (Jun.), pp. 196–199, 2004.

[34] K. Goto and R. Van De Geijn, "High-performance implementation of the level-3 BLAS," *ACM Transactions on Mathematical Software*, vol. 35, no. 1, p. 1–14, 2008.

[35] I. A. Sucan, M. Moll, and L. E. Kavraki, "The open motion planning library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, p. 72–82, Dec. 2012, http://ompl.kavrakilab.org.

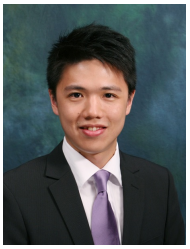
[36] A. Perez, S. Karaman, A. Shkolnik, E. Frazzoli, S. Teller, and M. Walter, "Asymptotically-optimal path planning for manipulation using incremental sampling-based algorithms," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 4307–4313.

[37] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *arXiv:1105.1186 [cs]*, May 2011. [Online]. Available: http://arxiv.org/abs/1105.1186

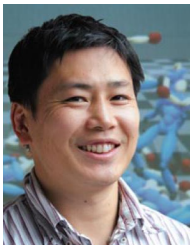
[38] E. D. Demaine and J. O'Rourke, "A survey of folding and unfolding in computational geometry," in *Combinatorial and Computational Geometry*, ser. Mathematical Sciences Research Institute Publications. Cambridge University Press, Aug. 2005, vol. 52, pp. 167–211.

[39] H. Alt, C. Knauer, G. Rote, and S. Whitesides, "The complexity of (un)folding," in *Proceedings of the nineteenth annual symposium on Computational geometry*. San Diego, California, USA: ACM, 2003, pp. 164–170. [Online]. Available: http://portal.acm.org/citation.cfm?id=777792.777818

He Wang is a post-doctoral Research Associate at the Institute of Perception, Action and Behavior, School of Informatics , University of Edinburgh. He received his PhD from Edinburgh University, UK and BEng. from Zhejiang University, China. His research has focused on motion planning and control involving deformable objects, scene analysis, data-driven animation and physical simulation. Recently he has been focusing on leveraging machine learning for solving problems in graphics and robotics.



Edmond. S.L. Ho received the Ph.D. degree from the University of Edinburgh, Edinburgh, Scotland, in 2011. He is currently a Research Assistant Professor with the Department of Computer Science, Hong Kong Baptist University. His current research interests include character animation, robotics, and human activity understanding.



Taku Komura is a Reader (Associate Professor) at the Institute of Perception, Action and Behavior, School of Informatics , University of Edinburgh. As head of the Computer Animation and Robotics group his research has focused on data-driven character animation, physically-based character animation, crowd simulation, cloth animation, anatomy-based modeling and robotics. Recently, his main research interests have been in indexing and animating complex close interactions, which includes character-character interactions and character-object interactions.