A Distance-Based Approach for Human Posture Simulations

Antonio Mucherino[*], Douglas Gonçalves[†], Antonin Bernardin[‡], Ludovic Hoyet[§], Franck Multon[¶]

[*]IRISA, University of Rennes 1, Rennes, France.
antonio.mucherino@irisa.fr
[†]DM-CFM, Federal University of Santa Catarina, Florianópolis, Brazil.
douglas@mtm.ufsc.br
[‡]University of Limoges and INRIA Rennes, France.
antonin.bernardin@inria.fr
[§]INRIA Rennes, France.
ludovic.hoyet@inria.fr
[¶]IRISA, University of Rennes 2, Rennes, France.
franck.multon@irisa.fr

Abstract—**Human-like characters can be modeled by suitable skeletal structures, which basically consist in trees where edges represent bones and vertices are joints between two adjacent bones. Motion is then defined as variations of the joints' configuration (i.e., partial rotations) over time, which also influences joint positions. However, this representation does not allow to easily represent the relationship between joints that are not directly connected by a bone. This work is therefore based on the premise that variations of the relative distances between such joints are important to represent complex human motions. While the former representations are currently used in practice for playing and analyzing motions, the latter can help in modeling a new class of problems where the relationships in human motions need to be simulated. Our main interest in this work is in adapting previously captured human postures (one frame of a given motion) with the aim of satisfying a certain number of geometrical constraints, which turn out to be easily definable in terms of distances. We present a novel procedure for approximating the relative inter-joint distances for skeletal structures having arbitrary features and respecting a predefined posture. This set of inter-joint distances defines an instance of the Distance Geometry Problem (DGP), that we tackle with a non-monotone spectral gradient method.**

I. Introduction

IN COMPUTER animation, human characters are typically modeled by a skeletal structure, which is a weighted tree $S = (V, E_{skel}, b_{skel})$, where every vertex $v \in V$ represents a joint of the character, and where every edge connecting two joints u and v represents a bone. Weights are associated to the bones, that indicate their length δ_{uv} through the function:

$$b_{skel} : \{u, v\} \in E_{skel} \longrightarrow \delta_{uv} \in \mathbb{R}_+.$$

In this setup, a motion for a given character can be represented by the joints' orientations over time, which influence joint positions [4], [9]. While this representation is commonly used in domains such as video games or human motion analysis, it does not allow, however, to easily represent the relationship between joints that are not directly connected by a bone (e.g., how a wrist moves in relation to the hip of the character).

We focus our attention on a novel methodology to describe a character and its motion by relative inter-joint distances. In the current literature, the simulation of character motions is generally performed either by modifying existing captured motions, where character and/or motion features may be manipulated [8], or by simulating motions from physical equations [3]. By exploiting distance information, it is our aim to study novel methods that are able to distinguish the geometrical information regarding a character, and the motion associated to it. In this way, the simulation of a modified motion for a modified character becomes trivial to model, and solely based on distance constraints.

In first analysis, we focus in this paper on human *postures*, that is, on only one "frame" of a given human motion. The main problem that we address is the following. How to adapt a given posture, extracted from a certain skeletal structure, to another skeletal structure having arbitrary bone lengths? The methods proposed in this paper can potentially be integrated in tools for motion simulation, and applied to all frames (all postures) of a given motion.

A posture x for the skeletal structure S in the Euclidean space $\mathbb{R}^3$ is a mapping $x : V \longrightarrow \mathbb{R}^3$, so that every $x_v = x(v)$ provides the coordinates of the joint v in the posture. If an ordering is associated to the joints (some possible orderings can be deduced from the tree structure of S), then the posture can also be expressed in matrix form by $X = [x_1 x_2 \ldots x_n]^\top$, where $n = |V|$ and every x_v is a column vector. The matrix X has n rows and 3 columns (corresponding to the dimension of the Euclidean space). From a known posture X for S, the distance matrix $D = [D_{uv}]$ can be defined such that:

$$\forall \{u, v\} \in V \times V, \quad D_{uv} = ||x_u - x_v||, \tag{1}$$

where $||\cdot||$ is the Euclidean norm.

Given a distance matrix $D = [\delta_{uv}]$, the Distance Geometry Problem (DGP) in dimension 3 asks whether a *realization* X in the three-dimensional space for D exists so that all distances

δ_{uv} are satisfied [6]. Notice that, the distance matrix D is said a *Euclidean Distance Matrix* (EDM[1]), in dimension 3, when the DGP admits at least one solution. In our context, a realization X corresponds to a posture for the corresponding skeletal structure S.

A common approach to the DGP is to reformulate it as an unconstrained global optimization problem, where a penalty function is utilized for measuring the violation of the distance constraints [6]. The terms of the penalty function, one for each constraint, measure the absolute difference between the computed value $||x_u - x_v||$ and the expected value δ_{uv}. One example of penalty function, that also takes into consideration the fact that priority levels π_{uv} can be associated to the distances, is given by

$$\sigma_w(X) = \frac{1}{2} \sum_{\{u,v\} \in E} \pi_{uv}(||x_u - x_v|| - \delta_{uv})^2. \qquad (2)$$

After the reformulation of a DGP as a global optimization problem, its solution consists in finding a global optimum for the chosen penalty function. When the matrix D is an EDM, the value of the objective function in the solution is supposed to be zero. Otherwise, this value is strictly positive because of the error introduced on some of the distances. In our particular DGP application, the magnitude of the constraint violations cannot be a priori estimated.

In this work, we propose a novel procedure for manipulating distance matrices D with the aim of imposing a predefined set of geometrical constraints to the postures X to be simulated. Our procedure is based on the idea of separating the geometrical information concerning the character (the tree S) from the posture X. The modified distance matrix $D = [\delta_{uv}]$, containing an approximated set of distances, is then *realized* by using an ad-hoc approach to the DGP.

It is important to remark that, after performing some modifications on distance matrix D obtained by applying (1), it is likely that this distance matrix is no longer an EDM, so that approximate posture X, in a least squares sense, need to be searched. This is equivalent to searching for the EDM that is the nearest to D [1]. In such a case, associating a priority level π_{uv} to every distance δ_{uv} becomes fundamental: we seek and select in fact approximate postures that are able to privilege higher priority distances.

The rest of this short paper is organized as follows. In Section II, we will present our original method for manipulating a distance matrix D, initially representing a posture for a given skeletal structure S, so that it represents the same posture for a skeletal structure having modified bone lengths. This distance matrix will be used for creating new DGP instances that we will solve by employing a spectral gradient method implementing a non-monotone line search. Some preliminary computational experiments on a predefined human walking posture are presented in Section III, and Section IV will briefly conclude the paper.

---

[1]In some articles, EDMs are distance matrices where the entries are squared distances. In this paper, the distance matrix D contains *non-squared* distances.

II. SKELETON-INDEPENDENT DISTANCE MATRICES

Let us suppose that the matrix X_1 of positions for the joints of the skeletal structure S_1 is known. X_1 is a possible realization of the skeletal structure S_1 which satisfies all bone-length constraints in the set $b_{skel}(E_{skel})$. The other inter-joint distances, that are not pre-defined in S_1, give a possible posture of the skeletal structure. Our main idea is to capture the distance information related to this posture, independently from the features of the skeletal structure (i.e. the bone lengths). This way, this distance-based information about the posture can be subsequently associated to a skeletal structure having different bone lengths.

Since X_1 is known for S_1, all inter-joint distances D_{uv} can be computed by applying equ. (1), so that an initial distance matrix D can be defined. Naturally, all bone-length constraints in $b_{skel}(E_{skel})$ are satisfied by X_1. The realization of this initial matrix D can be performed in polynomial computational time, and a high-quality approximation of the unique solution can be easily obtained [2]. Naturally, this unique solution corresponds to the original posture X_1 of S_1 (modulo rotations, translations and reflections).

We propose to extract the information about the posture of S_1 represented by X_1 with the following procedure. Our procedure is based on the idea to compute all shortest paths $P_{uv} = \{p_1, \ldots, p_k\}$ between pairs of distinct joints, where $p_1 = u$, $p_k = v$ and, for every $i = 1, \ldots, k-1$, we have $\{p_i, p_{i+1}\} \in E_{skel}$. The term "shortest" makes reference to the number of edges that need to the crossed by the path to walk from the vertex u to the vertex v of the skeletal structure S; it is not related neither to the distances δ, not to priority levels π. We refer to the sum of the distance values δ over a path P_{uv} as the *weight* τ_{uv} of the shortest path P_{uv}, computed as:

$$\tau_{uv} = \sum_{i=1}^{(|P_{uv}|-1)} \delta(p_i, p_{i+1}).$$

We will use the superscript "(1)" for indicating that shortest paths P_{uv} and weights τ_{uv} are related to the skeletal structure S_1; the superscript "(2)" will be employed below when referring to S_2.

Once all shortest paths $P_{uv}^{(1)}$ over the skeletal structure S_1 are computed, we normalize the computed distances D_{uv} with the weights $\tau_{uv}^{(1)}$ of the corresponding shortest path. In other words, instead of considering the distances computed by equ. (1), we apply the following formula:

$$\forall \{u,v\} \in V \times V, \quad D_{uv} = ||x_u - x_v||/\tau_{uv}^{(1)},$$

where x_v is the position of the generic vertex v in the posture X_1. Notice that all distances related to bone lengths are equal to 1 after the normalization, and that distances close to 0 indicate an inter-joint contact, while non-bone distances close to 1 are related to completely extended configurations. We point out that the idea to normalize relative distances is not completely new, and that it was partially exploited for example in [5] in a morphology-independent representation of
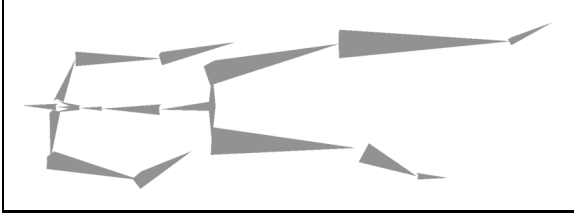
Fig. 1. The posture X_1 of the original skeletal structure S_1.

TABLE I
SOME DETAILS ABOUT THE CONSIDERED SKELETAL STRUCTURE S, WITH THE ORIGINAL BONE DISTANCES (DISTANCE OF EVERY JOINT FROM ITS PARENT, IN PARENTHESIS).

Joint label	Joint name	distance to parent	
H	Hips	/	
C	Chest	0.160	(H)
CA	Chest2	0.179	(C)
CB	Chest3	0.069	(CA)
N	Neck	0.069	(CB)
HD	Head	0.106	(N)
RC	RightCollar	0.072	(CB)
RS	RightShoulder	0.158	(RC)
RE	RightElbow	0.273	(RS)
RW	RightWrist	0.258	(RE)
LC	LeftCollar	0.072	(CB)
LS	LeftShoulder	0.158	(LC)
LE	LeftElbow	0.273	(LS)
LW	LeftWrist	0.258	(LE)
RH	RightHip	0.108	(H)
RK	RightKnee	0.435	(RH)
RA	RightAnkle	0.453	(RN)
RT	RightToe	0.125	(RA)
LH	LeftHip	0.108	(H)
LK	LeftKnee	0.435	(LH)
LA	LeftAnkle	0.453	(LK)
LT	LeftToe	0.125	(LA)

the motions, which is however not solely based on distance information.

In order to impose the posture in X_1 to another skeletal structure S_2, we apply first of all the formula for computing a new distance matrix:

$$\forall \{u,v\} \in V \times V, \quad \delta_{uv} = \tau_{uv}^{(2)} \cdot D_{uv},$$

where $\tau_{uv}^{(2)}$ is the weight of the shortest path $P_{uv}^{(2)}$ over S_2. This formula makes it possible to reconstruct correctly the bone lengths of S_2 while modifying accordingly the original distances in X_1, for them to be adapted to the new bone lengths of S_2.

Intuitively, distances between joints that are close in the skeletal structure (i.e. corresponding to shortest paths P_{uv} over fewer bones) can be approximated better than others (for example the distance between a hand and a foot is more difficult to approximate). For this reason, every computed distance δ_{uv} is coupled with the priority level π_{uv}, that is based on the cardinality $|P_{uv}|$ of the corresponding shortest path:

$$\pi_{uv} = (|P_{\max}| - |P_{uv}| + 2)/|P_{\max}|,$$

where $P_{\max}$ is the longest shortest path that can be defined over the two skeletal structures S_1 and S_2. Notice that, in correspondence with the bone lengths, the priority π_{uv} is maximal and equal to 1; the smallest possible priority value is given by $2/|P_{\max}|$. This distance information, together with the associated priority levels, defines a DGP instance for the simulation of human postures.

III. COMPUTATIONAL EXPERIMENTS

In our computational experiments, we consider a human walking posture extracted from a walking motion. This section shows the results obtained by constructing an approximated distance matrix by the method detailed in Section II, and by looking for the corresponding posture by implementing the spectral gradient method with non-monotone line search described in [7]. All codes were written in Matlab 2016b and the experiments were carried out on an Intel Core 2 Duo @ 2.4 GHz with 2GB RAM, running Mac OS X.

A graphical representation of the original skeletal structure is displayed in Fig. 1, together with the initial posture from which we extract the distance information. Table I shows the list of short labels for all joints forming the skeletal structure, together with the original distances from the corresponding joint parents (bone lengths).

Table II shows some preliminary experiments, where the parameters of the non-monotone spectral gradient method, are the same of the experiments presented in [7]. The distance matrix is generated by applying the procedure detailed in Section II. As a starting point X_0 in the spectral gradient method, we considered the original posture with the original bone lengths. All experiments took less than 1 second of CPU time.
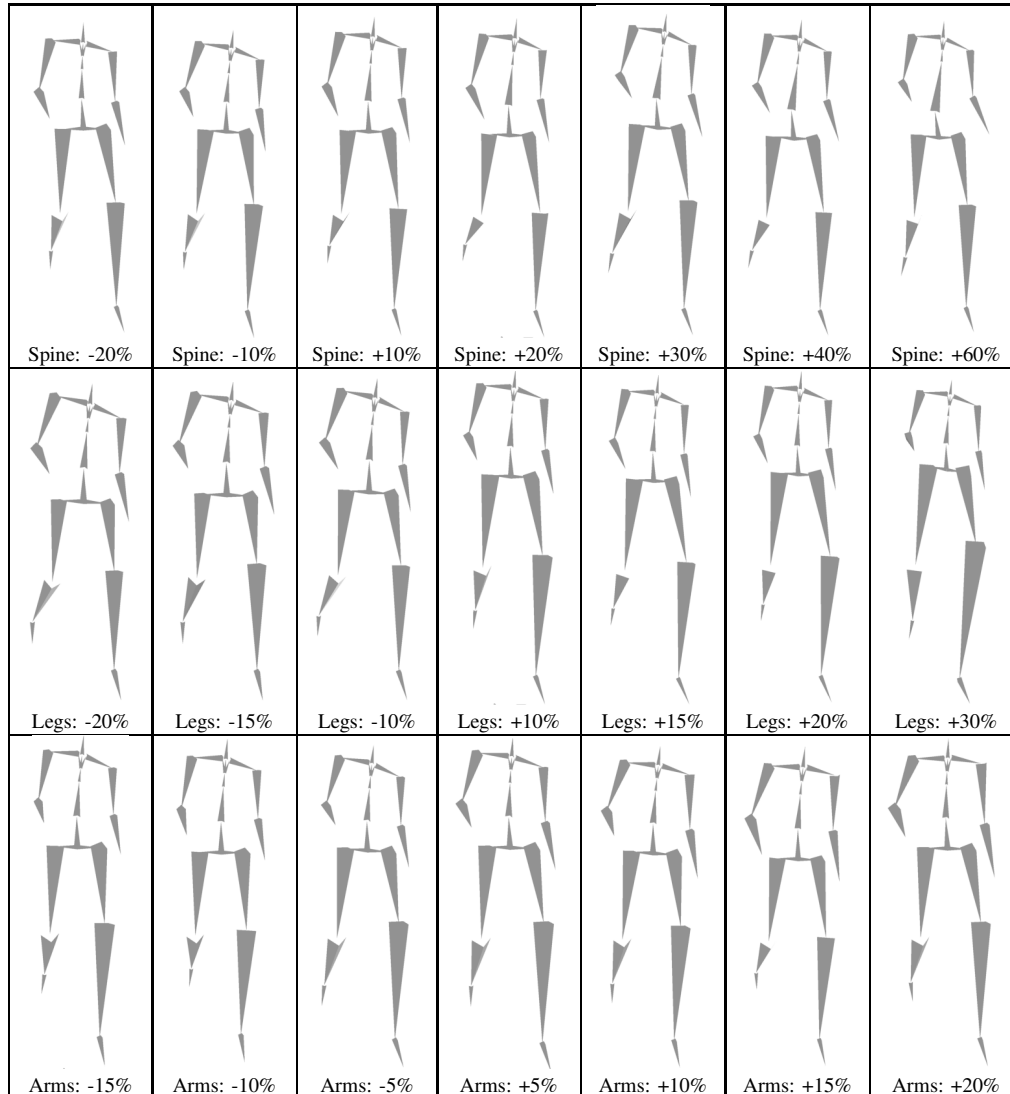
As the table shows, the method is able to reproduce quite well the original posture in the new skeletal structures. It is possible to remark, however, that when very important changes in the bone lengths are imposed, the obtained posture can show some slight deformations w.r.t. the original one (see for example the posture obtained when imposing a 60% longer spine). In order to avoid such undesired effects, we are currently working on a more advance method for the definition of the approximated distance matrices and of the associated priority levels.

IV. CONCLUSIONS

We have presented a novel approach, solely based on distance information, for simulating a given human posture of a skeletal structure having arbitrary bone lengths. Once an approximated distance matrix is defined, the problem to be solved is a classical one in the context of distance geometry, where an EDM near to the approximated matrix needs to be identified, by revealing in this way the posture for the new skeletal structure. Our computational experiments show that the presented methodology is promising when the corresponding distance geometry problem is solved by using the non-monotone spectral gradient method proposed in [7] in the context of the *dynamical* distance geometry. Future works will mostly be aimed at improving and at performing a

TABLE II
A SET OF EXPERIMENTS WHERE AN APPROXIMATED DISTANCE MATRIX IS GENERATED BY IMPLEMENTING THE PROPOSED METHOD, AND WHERE A
POSTURE IS OBTAINED BY A SPECTRAL GRADIENT METHOD WITH NON-MONOTONE LINE SEARCH.

Spine: -20%	Spine: -10%	Spine: +10%	Spine: +20%	Spine: +30%	Spine: +40%	Spine: +60%
Legs: -20%	Legs: -15%	Legs: -10%	Legs: +10%	Legs: +15%	Legs: +20%	Legs: +30%
Arms: -15%	Arms: -10%	Arms: -5%	Arms: +5%	Arms: +10%	Arms: +15%	Arms: +20%

theoretical validation of the procedure detailed in Section II, as well as at extending the entire methodology to the simulation of motions.

ACKNOWLEDGMENTS

REFERENCES

[1] I. Dokmanic, R. Parhizkar, J. Ranieri, M. Vetterli, *Euclidean Distance Matrices: Essential Theory, Algorithms, and Applications*, IEEE Signal Processing Magazine **32**(6), 12–30, 2015.

[2] Q. Dong, Z. Wu, *A Linear-Time Algorithm for Solving the Molecular Distance Geometry Problem with Exact Inter-Atomic Distances*, Journal of Global Optimization **22**, 365–375, 2002.

[3] J.K. Hodgins, W.L. Wooten, D.C. Brogan, J.F. O'Brien, *Animating Human Athletics*, Proceedings of the 22nd annual conference on Computer Graphics and Interactive Techniques (SIGGRAPH95), 71–78, 1995.

[4] N. Lever, *Real-time 3D Character Animation with Visual C++*, Taylor & Francis, 496 pages, 2001.

[5] R. Kulpa, F. Multon, B. Arnaldi, *Morphology-Independent Representation of Motions for Interactive Human-like Animations*, Proceedings of EUROGRAPHICS 2005, M. Alexa, J. Marks (Eds.), Computer Graphics Forum **24**(3), 343–351, 2005.

[6] L. Liberti, C. Lavor, N. Maculan, A. Mucherino, *Euclidean Distance Geometry and Applications*, SIAM Review **56**(1), 3–69, 2014.

[7] A. Mucherino, D.S. Gonçalves, *An Approach to Dynamical Distance Geometry*, to appear in Proceedings of the 3rd Conference on Geometric Science of Information (GSI17), Lecture Notes in Computer Science, 8 pages, 2017.

[8] F. Multon, L. France, M.P. Cani-Gascuel, G. Debunne, *Computer Animation of Human Walking: a Survey*, The Journal of Visualization and Computer Animation **10**(1), 39–54, 1999.

[9] F. Multon, R. Kulpa, L. Hoyet, T. Komura, *From Motion Capture to Real-Time Character Animation*, Proceedings of the First International Workshop on Motion In Games (MIG08), Lecture Notes In Computer Science **5277**, 72–81, 2008.