A Generic Method for Geometric Contraints Detection

Marc Salvati[1], Benoît Lecallennec[2] and Ronan Boulic[2]

[1]Precision and Intelligence Laboratory, Tokyo Institute of Technology, Japan
[2]Virtual Reality Lab, Swiss Federal Institute of Technology, Lausanne, Switzerland

Abstract

In this paper, we present a generic method to automatically detect geometric constraints on motion capture animations. At each frame, elementary geometric constraints are computed with respect to a reference which can either be the world coordinate system or any moving object in the scene. We then use constraint-related concepts of union and intersection to merge the elementary constraints together and/or to generate new ones. Finally, our algorithm provides an exhaustive list of geometric constraints with an accurate evaluation of their duration. The detected constraints can characterize virtual human motion (e.g. footprints) as well as interactions with moving objects of the scene (e.g. a hand touching a ball). While our approach also detects sliding geometric constraints, we focus our discussion on detecting positional geometric constraints.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism

1. Introduction

Constraint-based motion editing techniques are designed to change existing motion sequences while retaining as many of their important characteristics as possible. These characteristics are often made explicit using geometric constraints. It is often useful to detect them automatically in order to simplify and speed up the process of motion editing. Indeed, defining geometric constraints by hand is time-consuming. For example, in most locomotion animations, defining each footprint may take several hours.

In this paper, we propose a generic algorithm to automatically detect geometric constraints on motions as well as interaction constraints with objects of the scene. Our method can be summarized as follows:

1. we first specify the objects for which we want to detect constraints (usually, we consider segments of the human characters in the scene);
2. for each frame, we compute their displacement (6 dimensions) to the next frame in predefined reference frames (usually, frames of moving objects and/or the world frame);
3. given these transformations, we compute the points (or geometries) that remain stationary from one frame to the next with respect to the reference objects previously defined;
4. finally, given these geometries, we construct the final set of constraints by evaluating their duration, by merging geometries together, and/or by generating new ones.

The remainder of this paper is organized as follows: in section 2 we review previous work on geometric constraints detection. In section 3 we detail our algorithm. Experimental results are presented in section 4. Finally, we conclude this paper by discussing results and future work.

2. Previous Work

Constraints Definition and Type: From our point of view, the most general formulation of constraints was introduced by Witkin et al. in [WFB87]. In their formulation, constraints are defined as energy functions which are then minimized.

Geometric constraints are more intuitive because they directly specify a goal for a specific body part: a point can be constrained to a specific position [Gle97], [Gle98], or can be constrained to move along a line [WFB87]. [LP02] and [WP95] used keyframes to specify motion features. Keyframes may be considered as a set of geometric constraints that determine the position and/or orientation of each joint.

© The Eurographics Association 2004.

delivered by EUROGRAPHICS DIGITAL LIBRARY www.eg.org diglib.eg.org

In order to look realistic, a motion should respect physical laws as much as possible. In [PW99], Newton's laws are applied on a simplified character to minimize computational costs. For motion synthesis, Rose et al. [RGBC96] minimized energy consumption to obtain realistic transitions between motions, while Liu and Popović used minimum mass displacement and momentum conservation in [LP02].

In [BB98], Bindiganavale and Badler introduced interaction constraints to map a motion from one character to another with different morphology. Gleicher ([Gle97], [Gle98]) introduced a kind of interaction constraint: a point that should have the same motion as another point, a constant distance between two points and/or, a constant orientation between two points. [WFB87] used a parametric model and allows the specification of geometrical constraints with interactions like surface attachment or collision constraints. A more specific interaction constraint has been introduced in [CKB99]: by being able to produce realistic behaviors of visual attention, the authors specify a kind of visual constraint with the environment.

Constraints Detection Methods: Some automatic techniques for geometric constraints detection have been introduced to address specific needs in footprints detection. In [Gle97], [Gle98] and [LK02] the authors used position/orientation between the foot and the ground to determine constraints. Liu and Popović [LP02] proposed a general method to detect geometric constraints on a character's body parts. This method does not consider the interaction with the objects of the scene. [CKB99] built a constraints detector of visual behaviors from agents' intentions. In [BB98], Bindiganavale and Badler used the proximity of predefined sites of interest to deduce interaction constraints. In our framework, we decided instead to rely on the motion capture data only.

Existing methods have some limitations. Most of them were introduced to solve specific problems and could not be applied to more general cases. Therefore we propose in the next section a way to detect constraints within a unified formalism that can address the detection of a wide range of geometric constraints.

3. Automatic Constraints Detection Method

3.1. Overview of the Method

Detecting constraints on an object **O** could be stated as follows: given a transformation of **O** between $frame_i$ and $frame_{i+1}$, find all the points p remaining stationary with respect to a reference frame **R**. More formally, we want to solve:

$$\mathbf{p}_{R_i} = \mathbf{T}_i \mathbf{p}_{R_i}$$

where $\mathbf{p}_{R_i}$ is a point of **O** expressed in **R** at $frame_i$ and $\mathbf{T}_i$ the transformation of **O** from $frame_i$ to $frame_{i+1}$ expressed in **R**.

This is equivalent to solving:

$$(\mathbf{T}_i - I_d)\mathbf{p}_{R_i} = 0 \quad (1)$$

The solution of this linear system may be of dimension:

- 0 = only one point in space remains stationary
- 1 = all the points of a line in space remains stationary
- 3 = all the points in space remain stationary

No rigid transformation exists such that all the points belonging to a plane remain stationary.

Computation of T_i: to solve equation 1 we need to compute $\mathbf{T}_i$ precisely. So, if $\mathbf{M}_i$ is the matrix that transforms a point $\mathbf{p}_O$ from the **O** local coordinate system to **R** at $frame_i$, then we have:

$$\mathbf{T}_i = \mathbf{M}_{i+1}\mathbf{M}_i^{-1}$$

Finally, considering $\mathbf{W}_{O_i}$ the matrix that transforms from the **O** local coordinate system to the world coordinate system at $frame_i$ and $\mathbf{W}_{R_i}$ the matrix that transforms from the **R** local coordinate system to the world coordinate system at $frame_i$

We obtain:

$$\mathbf{M}_i = \mathbf{W}_{R_i}^{-1}\mathbf{W}_{O_i}$$

Thus:

$$\mathbf{T}_i = \mathbf{W}_{R_{i+1}}^{-1}\mathbf{W}_{O_{i+1}}\mathbf{W}_{O_i}^{-1}\mathbf{W}_{R_i}$$

Matrix $\mathbf{T}_i$ is a generalization of the matrix $\mathbf{T}_i$ in [LP02]. By introducing the reference coordinate system of the object **R**, we extend the constraint detection method to any reference. To detect constraints on the motion itself, we simply define the reference object as the world coordinate system ($\mathbf{W}_R = \mathbf{I}_d$).

3.2. Generation of Constraints Set

After previous stages, we can detect any geometric constraint from $frame_i$ to $frame_{i+1}$ (elementary constraints). However, this is not the minimal set of constraints we could find. Thus, our algorithm evaluates the duration of the final geometric constraints by merging elementary constraints and/or generating new ones using constraint-related concepts of intersection and union.

We compute the distance in zone of interest (usually near the studied joint). In case of null distance with parallelism, we can merge two constraints. In case of null distance without parallelism, we can compute the intersection and generate new constraints. We also use duration and dimension to decide on merging and generation of constraints. Finally we obtain a minimal and exhaustive set of constraints with an accurate determination of their duration.

© The Eurographics Association 2004.

3.3. Comparison with Previous Techniques

Lui and Popović [LP02] restricted the final set of constraints to those intersecting the rigid bodies of the character. In our case, we do not consider any polygonal mesh during the computation as we are also interested in interaction constraints with objects of the scene.

The method presented in [BB98] is dedicated to the detection of interaction constraints. Their approach needs to identify sites of interest to effectively compute the distance between them and the end-effectors. This method could not be easily extended to more general constraints detection. Other techniques such as [Gle97], [Gle98], and [Gle01] are too specific to be generalized.

Our approach provides, in a unified formalism, the detection of constraints onto the motion itself as well as interaction constraints with the objects of the scene.

4. Results

We applied our algorithm on various motions and scenes. For more clarity, we only show the constraints we are interested in.

Geometric constraints: footprints Detecting footprints is a key problem in many situations. We applied our method to a raw motion capture walking animation composed of 500 frames.

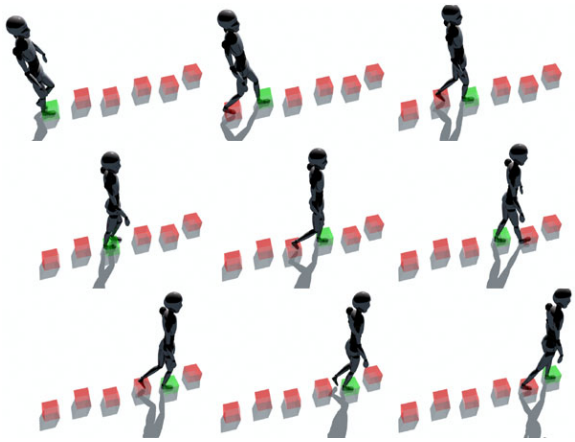

Figure 1: *Walking motion*

For more clarity, figure 1 shows detected footprints for the 125 first frames only. As in most motion capture animations, the input data is noisy and does not contain any strict footprints. However, in our case, good footprints are accurately detected without introducing wrong ones.

Note that, even if this walking motion was taking place on an animated ground like a treadmill, we would also detect the footprints, although as interaction constraints instead. This kind of interaction could not be detected by previous methods without adding more information to the scene.

Interaction constraints: dancing with a ball To detect interaction constraints, we added a ball to a raw motion capture dance motion composed of 125 frames. Results are shown in figure 2.

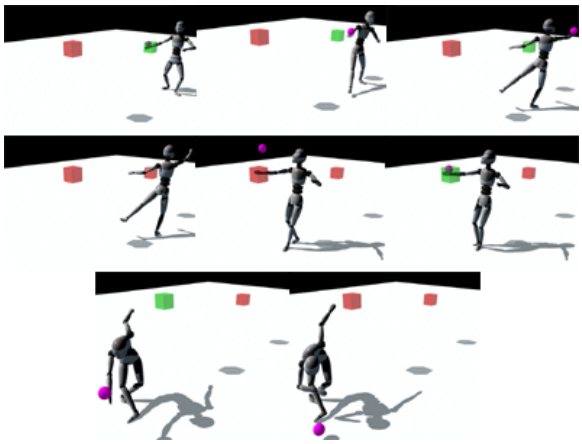

Figure 2: *Dancer throwing and catching a ball*

The dancer performs some dance steps while carrying a ball. After throwing and catching it, he drops it on the floor. The algorithm detects two interaction constraints between the hand and the ball according to the throw, the catch and the drop. These constraints could not be detected by [LP02]'s technique. We could have used [BB98]'s approach to find these constraints, but it requires more work from the animator as sites of interest need to be defined on the hand and on the ball.

4.1. Computational times

An important issue for techniques aiming at improving the process of motion editing is interactivity. Indeed, animators are disinclined to wait several minutes to have a first feedback.

	Nb. Frames	Computational times [s]
Walking Motion	500	4.69
Dancing with a Ball	125	0.2
Waiting Man	100	0.12
Dance motion	125	0.45

Table 1: *Computational times*

The computational times given in table 1, shows that our method is compatible with an interactive manipulation.

© The Eurographics Association 2004.

5. Discussion and future work

Automatic or semi-automatic? All constraints detection algorithms still have to be monitored by the animator as, in some specific cases, it is not possible to precisely detect whether a constraint is of real interest or not. Basically, our method detects points that remain stationary without being able to decide whether it is an important characteristic or not.

Adding high-level constraint detection Our approach is able to detect simple geometric constraints. However, these are hard to use for animators who generally need high-level constraints such as footprints or objects to reach.

Instantaneous constraints One class of constraint could not be detected by our method: the *instantaneous constraints* (e.g. contact of ball against racket). These constraints are so short in time that our algorithm would fail to detect them as we consider a minimal set of consecutives frames to decide whether a geometric constraint is of interest or not.

6. Conclusion

In this paper we have presented a unified framework to detect a wide range of constraints on the motion itself as well as interactions with objects of the scene. Our method directly relies on the information contained in the scene. Thus, we do not need any additional data nor preprocessing. We have also demonstrated the robustness of our method on raw motion capture data.

It is important to emphasize that animators actively participate in the constraints detection process, as the concept of "importance" of constraints is difficult to determine automatically. In [SLSG01], Shin et al. used some heuristics to estimate the importance of some constraints (end-effectors position, joint angles...). However, this method is dedicated to motion retargetting and could not be used in a general motion editing process. Thus, the final result is always left to the appreciation of the animator, who confirms, adjusts or deletes constraints depending on their subjective "importance".

In the previously presented examples, we have demonstrated that our method can provide good results for the detection of constraints contained in a scene. Micromanagement of constraints by the animator is considerably reduced, allowing them to focus more on artistic tasks.

7. Acknowledgement

We would like to thank Stéphanie Noverraz for carefull proof-reading, Nicolas Elsig for the design of virtual humans. We express our thanks to Alias for the granted Maya SDK licences through the research donation program. This research has been supported by the FNRS under the grant $N^o 2000 - 061999.00$.

References

[BB98] BINDIGANAVALE R., BADLER N. I.: Motion Abstraction and Mapping with Spatial Constraints. *Lecture Notes in Computer Science 1537* (1998), 70–83.

[CKB99] CHOPRA-KHULLAR S., BADLER N. I.: Where To Look? Automating Attending Behaviors of Virtual Human Characters. In *Proceedings of Conference on Autonomous Agents* (may 1999), pp. 16–23.

[Gle97] GLEICHER M.: Motion Editing with Spacetime Constraints. In *Proceedings of ACM Symposium on Interactive 3D Graphics* (apr 1997), pp. 139–148.

[Gle98] GLEICHER M.: Retargetting motion to new characters. *Computer Graphics 32*, Annual Conference Series (Aug. 1998), 33–42.

[Gle01] GLEICHER M.: Motion path editing. In *Proceedings of ACM Symposium on Interactive 3D Graphics* (2001), pp. 195–202.

[LK02] LUCAS KOVAR JOHN SCHREINER M. G.: Footskate Cleanup for Motion Capture Editing. In *Proceedings of ACM SIGGRAPH Symposium on Computer Animation* (jul 2002), pp. 97–104.

[LP02] LIU C. K., POPOVIC Z.: Synthesis of complex dynamic character motion from simple animations. In *SIGGRAPH 2002 Conference Proceedings* (2002), Hughes J., (Ed.), Annual Conference Series, ACM Press/ACM SIGGRAPH, pp. 408–416.

[PW99] POPOVIC Z., WITKIN A.: Physically Based Motion Transformation. In *Proceedings of ACM SIGGRAPH, Annual Conference Series* (1999), pp. 11–20.

[RGBC96] ROSE C., GUENTER B., BODENHEIMER B., COHEN M. F.: Efficient Generation of Motion Transitions using Spacetime Constraints. In *Proceedings of ACM SIGGRAPH, Annual Conference Series* (1996), pp. 147–154.

[SLSG01] SHIN H. J., LE J., SHIN S. Y., GLEICHER M.: Computer puppetry: An importance-based approach. *ACM Transactions on Graphics 20* (2001), 67–94.

[WFB87] WITKIN A., FLEISCHER K., BARR A.: Energy constraints on parameterized models. In *Proceedings of ACM SIGGRAPH, Annual Conference Series* (jul 1987), pp. 225–232.

[WP95] WITKIN A., POPOVIC Z.: Motion Warping. In *Proceedings of ACM SIGGRAPH, Annual Conference Series* (aug 1995), pp. 105–108.

© The Eurographics Association 2004.