Sketching 3D Animations using CUDA

High Performance Computer Graphics Laboratory

Innfarn Yoo, Juraj Vanek, Nicoletta Adamo-Villani, Maria Nizovtseva, Bedrich Benes
Department of Computer Graphics Technology, Purdue University, 401 N. Grant St., West Lafayette, IN 47907
yooi@purdue.edu, vanek@purdue.edu, nadamovi@purdue.edu, mnizovts@purdue.edu, bbenes@purdue.edu

PURDUE UNIVERSITY

CUDA TEACHING CENTER

Abstract

Quick creation of 3D character animations is an important task in game design, simulations, education, training, and more. We present a framework for creating 3D animations using a simple sketching interface coupled with a large, unannotated motion database that is used to find the appropriate motion sequences corresponding to the input sketches. Sketches can be enhanced by motion and rotation curves that improve matching in the context of the existing animation sequences. Our framework uses animated sequences as the basic building blocks of the final animated scenes, and allows for various operations with them such as trimming, resampling, or connecting by use of blending and interpolation. A database of significant and unique poses, together with a two-pass search running on the GPU, allows for interactive matching even for large amounts of poses in template database. The system provides intuitive interfaces, an immediate feedback, and poses very small requirements on the user.

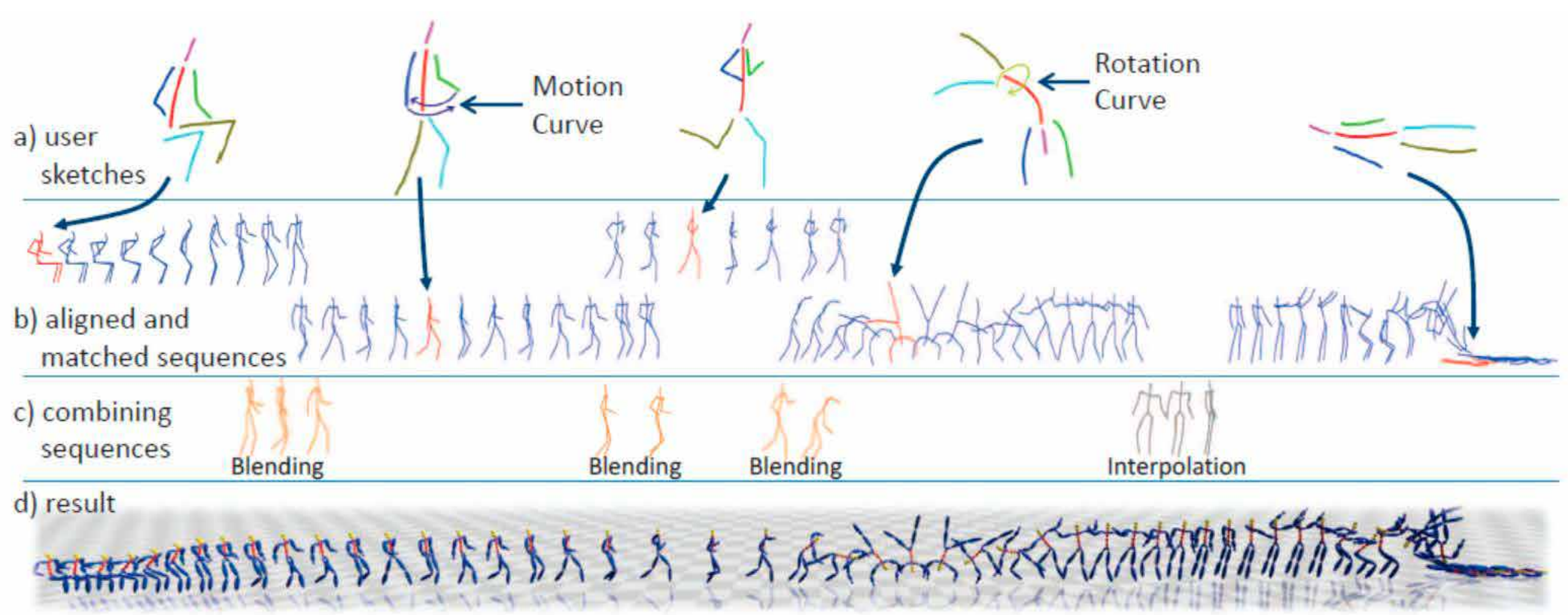

Figure 1. The user sketches poses with additional information about the desired motion of some joints (a). The system finds the best matching sequences in a large motion database (b) and the user defines if sequence blocks should be interpolated or blended partially together (c), which results in the final animation (d).

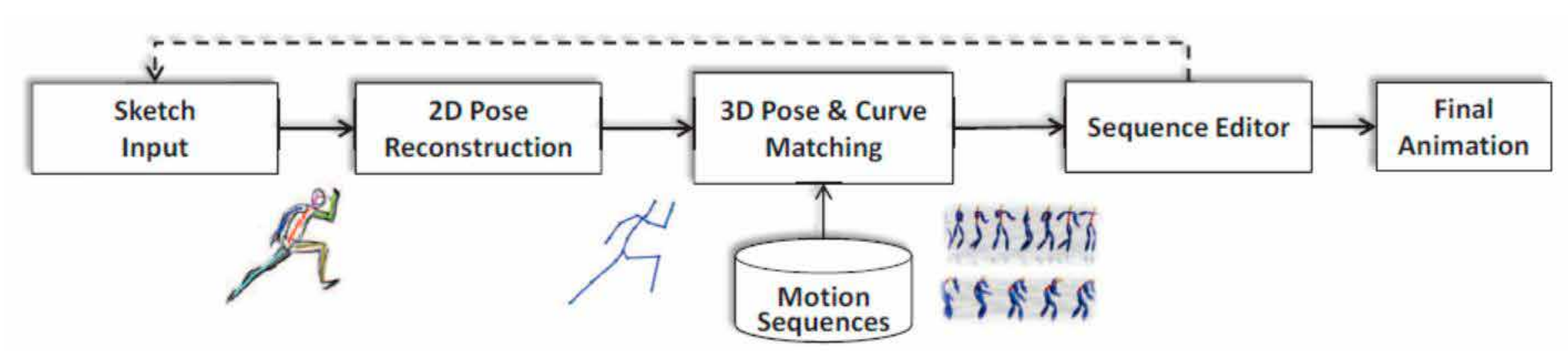

Figure 2. System Pipeline

1. Motivation

Existing 3D animation systems based on forward and inverse kinematics (FK/IK) or motion capture provide powerful tools but they can be expensive, have steep learning curve and creating good looking animation can take long time.

Using 2D sketching to create 3D animation allows the user to take advantage of the benefits of 2D modeling: **intuitiveness, fast pose definition**, and **quick production** of simple, first-pass animated scenes.

2. Database

We use the motion sequence database from the CMU Graphics Lab Motion Capture. Database includes 4 million unannotated poses in nearly 2, 400 different animation sequences occupying 3.1 GB of space and totaling 6.5 hours. The database includes sequences in more than 40 different skeleton formats.

From the MoCap database, we extract significant poses generate another types of database, called "motion snapshot database".

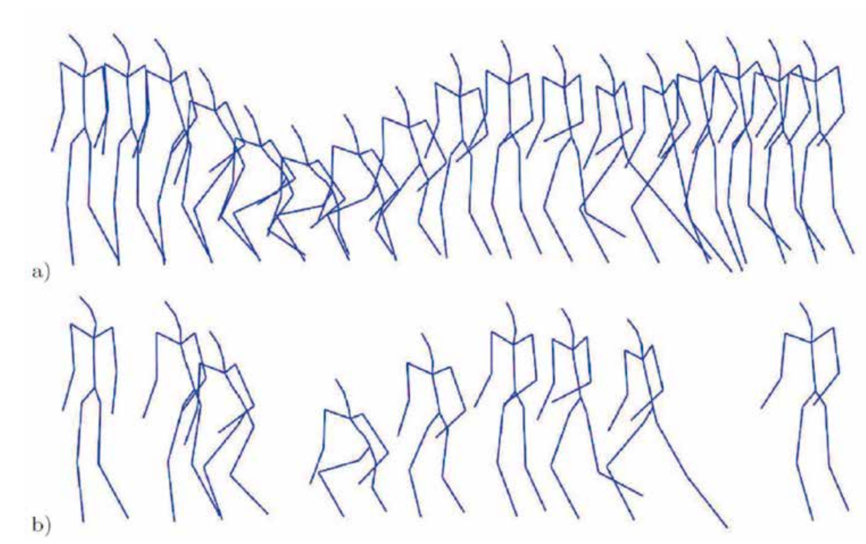

Figure 3. The initial sequence (a) and the frames that have significantly different poses (b).

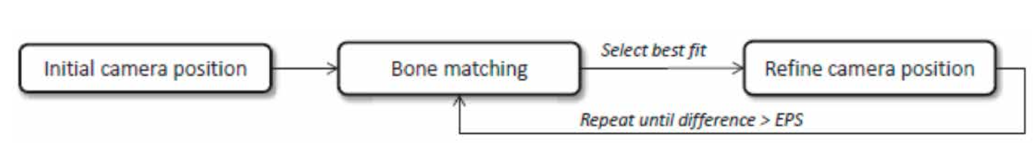

Figure 4. The pose matching process

3. Pose Matching

We detect 3D poses that match the 2D input pose. Because the sequences are independent, the 3D pose matching is done on the GPU in parallel. Sample results of the lookup can be seen in Figure 4.

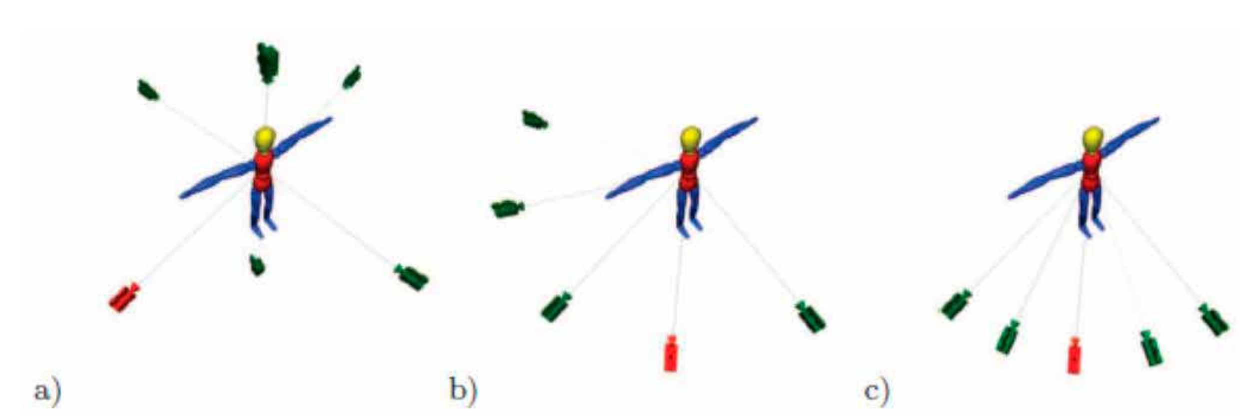

Figure 5. 2D pose matching. The pose is sampled from the six principle directions (a), the best match is found, and refined (b, c).

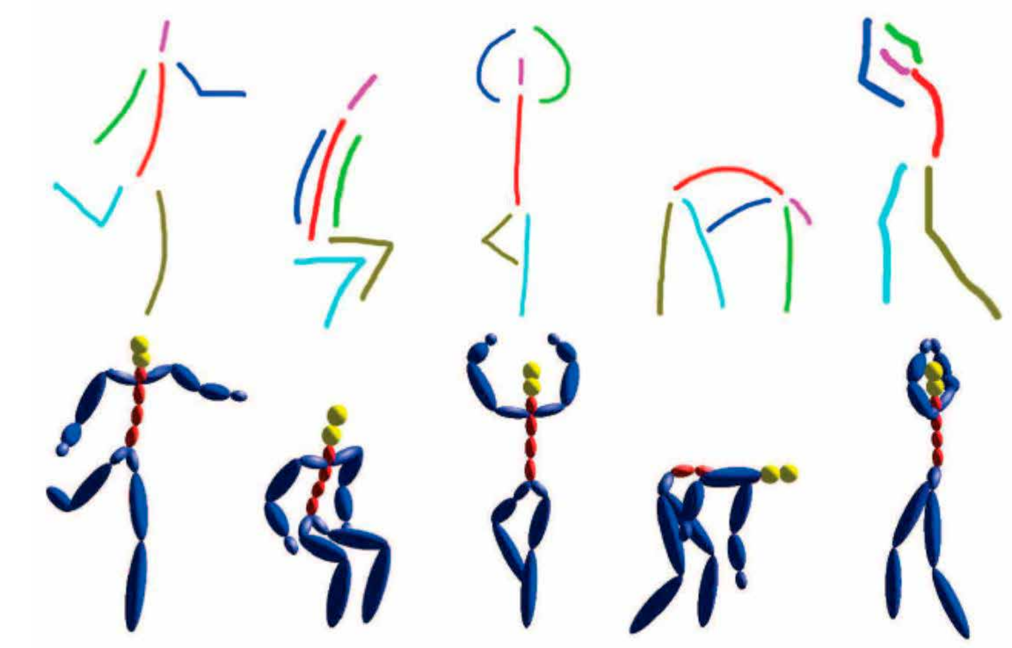

Figure 6. Sketched poses (up) and poses found in the database.

5. Refining matching results

The static pose matching gives reliable results for static poses, but it can be further improved by considering the pose in the context of the motion sequence. The user can define two kinds of additional motions – motion or rotation of a joint by sketching a curve in the proximity of a joint (see Figures 7 and 8).

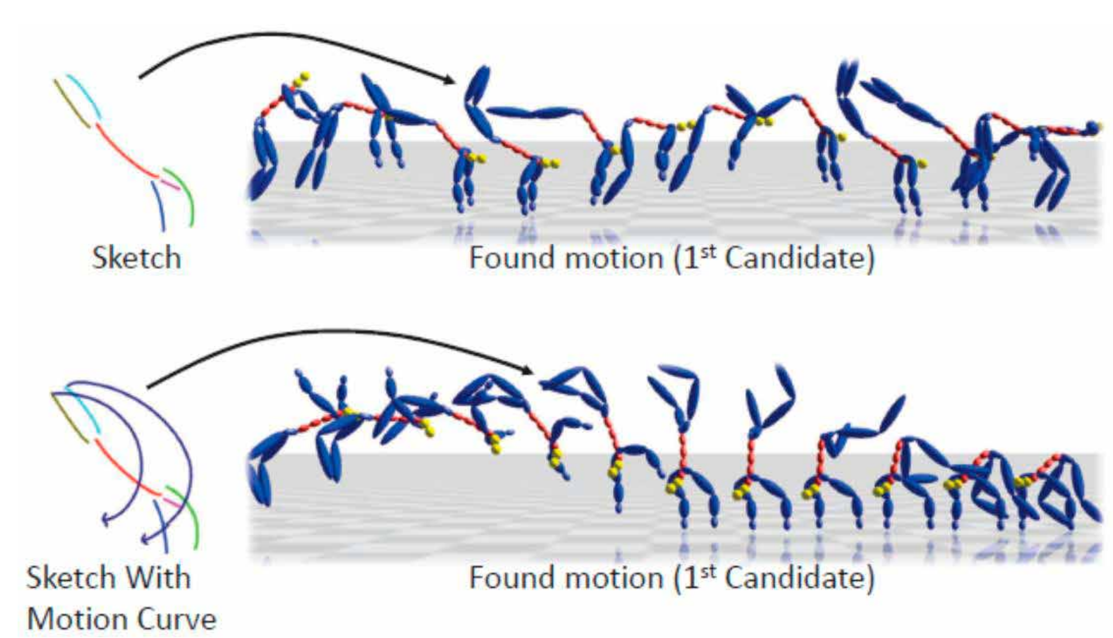

Figure 7. Pose definition with motion curve (down) helps to find the intended flip sequence.

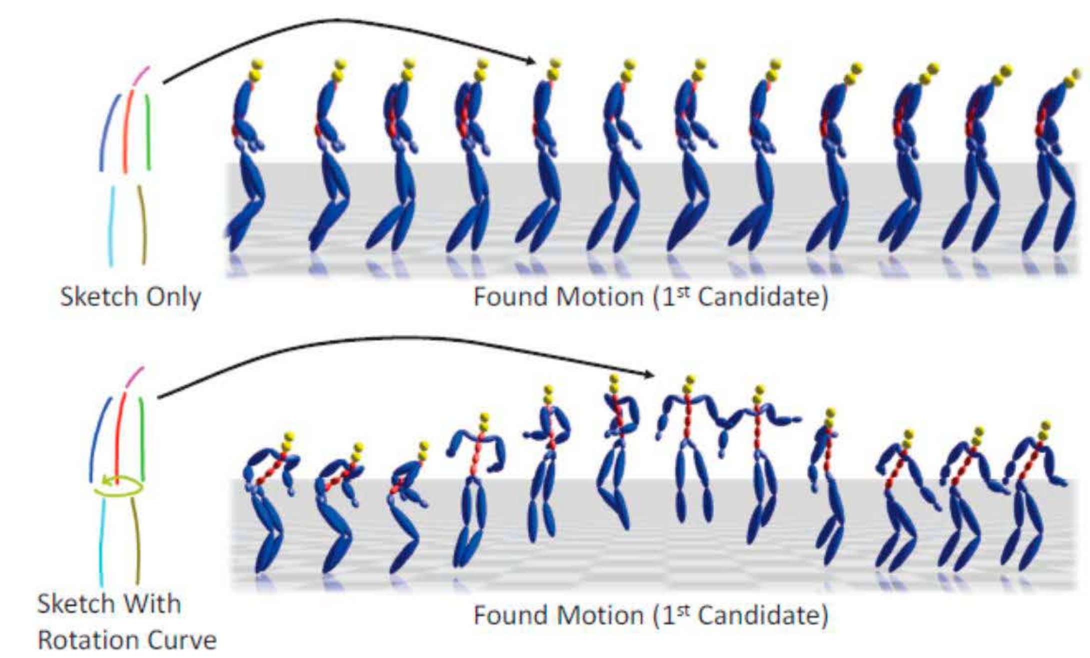

Figure 8. The rotating character (down) was found with the help of the sketched rotation around the hip.

6. Sequence Editing

The sequence editor allows the user to align the sequences' view directions by rotating the selected sequence and also to efficiently combine the selected sequences into the final animation. The sequences can be trimmed, resampled, blended, and interpolated. The operations that use multiple sequences also use the motion and rotation curves, if supported, to assist the transition.

7. Results

The GPU-oriented implementation provides immediate results and stores the order of the entire database of 6.5 hours on an off-shelf computer. In addition, our approach does not require additional database pre-processing, so that any motion capture data can be used immediately. Performance comparisons between the GPU and the CPU matching with various iterations during the camera searching process is shown in Table 1. GPU comparison is 500-800× faster than matching using only the CPU and it allows for real-time search of the best fitting poses.

Iterations	32	64	128	256
CPU (Core i7 820, 2.66 Ghz)	9,351 ms	18,647 ms	32,711 ms	75,193 ms
GPU (Nvidia Gefore GTX 480)	17 ms	28 ms	48 ms	88 ms
GPU Speedup	550.0588	665.9643	785.6458	854.4659

Table 1: Performance comparison between CPU and GPU implementation

ACKNOWLEDGMENTS

We would like to thank NVIDIA for providing graphics hardware. The data used in this project was obtained from mocap.cs.cmu.edu. The database was created with funding from NSF EIA-0196217.