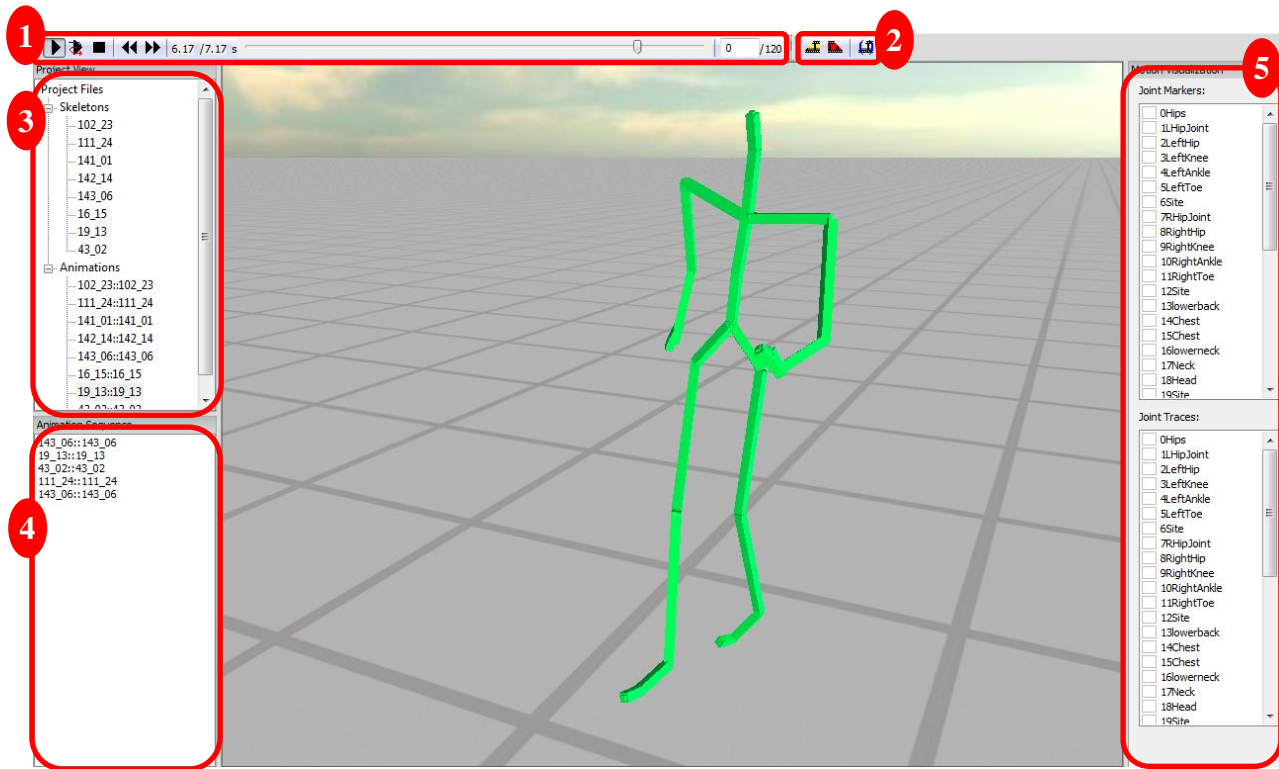


ZombieHorse Feature Overview



Testbed application interface. (1) Playback controls – Play/Pause, Play Sequence, Stop, Step to Previous/Next Frame, Timeline, Frame Rate. (2) Editing controls – Select/Deselect Animation Segment, Create New Animation From Selection. (3) Project View, showing all currently loaded skeletons and motions. (4) Animation Sequence – motions can be queued here by dragging and dropping them from Project View. (5) Motion Visualization – for specifying which joints should have markers and traces on them. Hotkeys: F9 – toggle root IK, F10 – toggle posture IK, F11 – toggle limb IK, F12 – toggle constraint prediction.

Parsing and playing back motions in BVH format. Currently we support 3ds Max Biped-friendly BVH files. This is convenient because the entire CMU database has been converted to this format and made available for download: <https://sites.google.com/a/cgspeed.com/cgspeed/motion-capture/3dsmax-friendly-release-of-cmu-motion-database>.

Displaying motions. The application renders motions using OGRE. The skeleton is represented as a set of bone segments visualized as oriented, non-uniformly scaled boxes.

Scrubbing and playing at different speeds. The application provides a scrubbable timeline for motion playback. The user can modify playback speed. We support frame-locked playback and manual stepping through frames.

Camera control. The application supports camera rotation, panning and zooming using the mouse.

Loading multiple motions. The application is currently set up to automatically load motions from a specific directory on startup. The loaded motions are then displayed in a list control, and can be selected for playback by double-clicking.

Joint marker and traces. The application can render purple markers on joints. Similarly, it supports joint "traces" – piecewise linear curves representing joint trajectories in the world space. The user can enable or disable markers and traces for any joint using appropriate checklists.

Splicing motions into sequences. The application can play multiple motions as a sequence. When transitioning from one motion to the next, the target motion is rigidly transformed so that root position and orientation are continuous. Our animation system can also perform blend-based transitions, but we do not expose this functionality in the application.

Creating new motion clips and writing them out. Our application allows the user to select a segment of an existing motion clip and create a new motion clip out of it. This can make those long motions easier to handle. The program does not yet

expose functionality for writing out motion files, but our animation system does have loaders and serializers for a custom XML file format, so adding this feature would be fairly trivial.

Motion retargeting. Our system can automatically retarget motions to characters with different proportions. To try out the retargeting system in the testbed application, manually annotate the environment with positions of "objects" that the character interacts with over the course of the motion.

Discovering similar motion clips. Our system can index a large motion dataset using a method called *match webs*. The user can then select a motion segment within the data set and the system will automatically discover similar motion clips. E.g., if the dataset contains multiple instances of walking and running cycles, our system can automatically find them all. However, this functionality is currently not exposed in the testbed application.

Building parametrized motion spaces. Given a class of similar motion clips, our system can create a *parametrized motion space*. The system can then blend the motion clips in the motion space to synthesize novel motions in the same class. E.g., multiple walking motions can be blended to obtain a walking motion in a new direction. However, this functionality is not exposed in the testbed application either.

Building motion graphs. Another features currently unsupported in the testbed application. Given a large motion dataset, our system can automatically discover high-quality transitions between motions.