EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling (2005)
Takeo Igarashi, Joaquim A. Jorge (Editors)

A Sketch-Based Gesture Interface for Rough 3D Stick Figure Animation

C. Mao, S. F. Qin, D.K. Wright

School of Engineering & Design, Brunel University, Uxbridge, Middlesex UB8 3PH, UK

Abstract

This paper introduces a novel gesture interface for sketching out rough 3D stick figure animation. This interface can allow users to draw stick figures with the system automatic assistance in figure proportion control. Given a 2D hand-drawn stick figure under a parallel view, there is a challenge to reconstruct a unique 3D pose from a set of candidates. Our system utilizes figure perspective rendering, and introduces the concept of 'thickness contrast' as a sketch gesture combined with some other constraints/assumptions for pose recovery. The resulting pose can be further corrected, based on physical constraints of human body. Once obtaining a series of 3D stick figure poses, user can easily sketch out motion paths and timing, and add their preferable sound/background. The resulting 3D animation can be automatically synthesized in VRML. This system has been tested on a variety of input devices: electric whiteboard, tablet PC, as well as a standard mouse.

Categories and Subject Descriptors (according to ACM CSS): H.5.2 [Information Interfaces and Presentation]: Graphical user interfaces (GUI); I.4.8 [Scene Analysis]: Depth cuing, motion; I.3.7 [Computer Graphics]: Animation.

1. Introduction

Animation, a creative and inspiring art form, has existed for over one hundred years. From the initial "hand-drawn animation" to modern computer animation, numerous technologies [WIK05][CM01] have been explored to boost the animation efficiency and quality. Recently, many IK/FK tools [Cho04][Fox04] have been developed to enable users to create animations for their own uses. However, making animation is still a painstaking and difficult task for ordinary (especially novice) users, who have no or very limited expertise and computer skills.

Instead, sketching is probably one of the most popular and easy ways to quickly rough out the imaginative characters and their motions. In fact, even children enjoy the doodling with pencil and paper. Meanwhile, comparing to full figure sketch, stick figure is distinctively fast and easy to draw, and powerful to illustrate the rough motions [Tin92][Li05].

To exert the power of paper based sketching into computer animation, we developed an intuitive sketching interface, which enables users to "draw" 3D animations. It allows users to interactively sketch stick figure key frames, graphically define motion path and timing, and finally "pop-up" their 2D characters into 3D animations by a single click. Allowing rapid 3D figure animation by simple sketching, our system is suitable for various levels of users (especially beginners), and may find wide applications in entertainment, education, cartoon storyboarding, etc.

Maintaining the right proportion and foreshortening is a common challenge in figure sketching for not only novice but also skilled artists. To help overcome this difficulty, we provide an on-line drawing assistance, which based on the utilization of template skeleton and the real-time body part recognition and length control.

Given a 2D figure and a parallel view, there is a challenge to identify a unique 3D pose from a set of candidates, which may be all consistent with the initial drawing. To solve this problem, we integrate the perspective rendering technique from figure drawing into system design and support multiple stroke drawing and incremental refining. A set of rendering (line thickness) gestures has been developed, together with other constraints and assumptions, for unique pose identification.

Since a quick and imprecise sketching may accidentally generate physically impossible poses, we offer an "overall pose checking/auto-correction" routine to ensure the physically valid poses complying with human joint range of motion (ROM).

Comparing with other similar systems [DAC*03][HH01][TBP04], our approach has the advantage of realizing the whole 3D animation process including rapid 3D key framing, overall motion/timing control, and final animation synthesizing, by almost pure sketch input. Meanwhile, sound/background can be added to the virtual world as well to improve the realistic effects. Although the resulting animations in VRML [ANM97] are simple and rough, it is rather inspiring for users to enjoy a quick 3D animation from 2D sketches, and to share it with the others via Internet.

2. Related works

The use of sketching in computer graphics may date back to the seminal SketchPad system [Sut63]. Gesture-based interfaces, frequently used in 2D pen-based applications [GD96][LM95][MCMK97], recognize specific stroke shapes as gestures and replace them with pre-defined primitives or invoke the subsequent editing operations. More recently, many sketch-based interfaces have been developed to combine the power of sketches and drawn gestures for fast geometric modelling. SKETCH [ZHH96] introduced a gesture-based interface for rapidly conceptualizing and editing approximate 3D scenes. Teddy [IMT99] extended the gesture-based sketching ideas from SKETCH into designing 3D freeform models, such as stuffed animals and other rotund objects. Skin [MCT*99] extended the principles of SKETCH, and used a particle-based surface representation with which a user can interactively sculpt freeform surfaces. However, all of these approaches are essentially static 3D modelling methods and do not embark on human figure related modelling and animation.

To realize sketch-based 3D figure animation, the major challenges are how to map from 2D freehand sketches into 3D posed models; and how to quickly and automatically transfer reconstructed key frames into 3D animation without user involvement. Several sketch-based systems [HH01][TBP04][DAC*03] have been developed recently to answer these questions.

Hoshino and Hoshino [HH01] presented an intelligent storyboard for CG animation prototyping. In their system, the 3D position and behavior of the characters are estimated from 2D views using the constraints optimization and example-based interpolation. Meanwhile, the perspective view is required, together with a pre-built 3D character/scene database.

Thorne's "motion sketching" interface [TBP04] enables users to sketch simple side view character key frames, define overall body motions by cursive gesture drawing and "act-out" the timing when doodling the motions. However, the general pose 2D-3D recovery is not addressed by this system, since only side view characters are accepted. Meanwhile, the comprehensiveness of motions is also limited based on a single motion gesture alphabet.

Davis et al [DAC*03] developed a sketching interface for 3D articulated figure animation and presumed a parallel view, which is in principle similar to ours. To solve the "back-front ambiguities" problem (two possible 3D poses existing for each foreshortened bone segment because of reflective amibiguity), a semi-automated method has been used. It reconstructs all possible 3D poses first and then presents users with the ranked choices. The user's selection will then help to refine this candidate list to the final desired 3D pose. This method provides a rapid 3D key-framing way, and to some extent, balances the user's guide and the system automation.

However, more emphases need be put on the sketches themselves, since they exhibit the best clues for pose identification. When sketching a figure, artists usually express a body pose by perspective rendering. For instance, they draw multi-strokes on a relatively closer body part to make it visually stronger [Tin92][Bro90]. In this sense, an intended pose has already been portrayed by the rendering strokes. Utilization of them into pose identification will make the key framing process more natural and closer to a real figure sketching in practice. Our previous user survey study [MQ04] (including questionnaire and sketching interviews with various users: artists, designers and animators) have also validated this fact. Rendering strokes were incrementally drawn to indicate perspective effects even during a very quick figure sketching. Depth differences were usually manifested by the size/thickness contrasts of joint/body parts. This fact is consistent with the NPR and depth cuing principles [SS02].

3. 2D stick figure sketching interface

To sketch 3D animations using our system, users need to go through the following three stages (as shown in Figure 1): 1) Draw freehand 2D stick figures; 2) Reconstruct 3D poses from 2D drawings; 3) Define motion path and timing for generating 3D animation.

3.1 Input 2D stick figure with on-line drawing assistance

Our system adopts a stick figure model (See Figure 2) containing 12 bone segments, and 13 joints (including a joint to link upper and lower body). In this section, we present the details for the first stage.

To start creating a new animation, users need to begin by specifying a template (reference) skeleton. Our previous user survey study [MQ04] reveals that, during rapid figure sketching (especially in imaginative drawing without models), people are usually more engaged in expressing an overall figure configuration. Thus, they end up with an figure drawing with incorrect body proportion and foreshortening due to less attention.

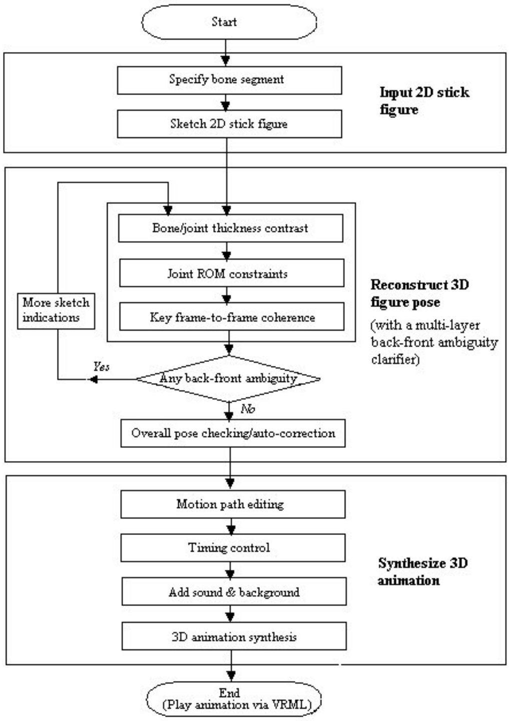


Figure 1: *Overview of system working pipeline.*

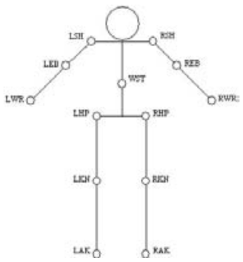


Figure 2: *The stick figure model containing 13 joints (labelled by their abbreviations) and 12 bone segments.*

Therefore, interactive assistance is demanded for effectively guiding user's proportion and foreshortening maintaining, while not distracting their inspiration and creativity during drawing. Thus, we provide an "on-line drawing assistance" based on the following assumptions:

1) A template skeleton is required as not only a graphic reference for users, but also to provide the length criterion for the system about 3D true length of bone segments.
2) Real-time recognition for an on-drawing body part is needed, in order to confine its maximum stretching length.
3) Foreshortened/non-foreshortened bone segments should be clearly distinguished whenever on the fly or drawn-up; visual indication is crucial for proportion and foreshortening maintaining.

Currently, we offer a set of pre-defined template skeletons [TD02] categoried by gender, ethnicity, and age. Users can pick-up templates directly through a menu selection, or they can customerize their own templates by modifying the pre-defined skeleton models in proportion. Once specified, template skeleton is displayed on interface as reference for subsequent key frame sketching (See Figure 3).

Our current system accepts bone segment input as a 2D straight line. The newly drawn segment snaps to the previous one automatically to ensure the connectivity. Like artists refining their figure drawing by incrementally adding details, users can render extra strokes on their drawings at anytime, to indicate depth information when posing a figure.

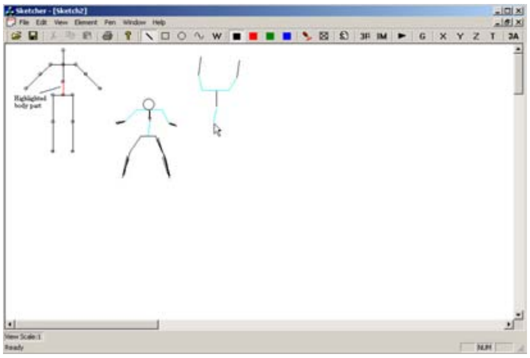


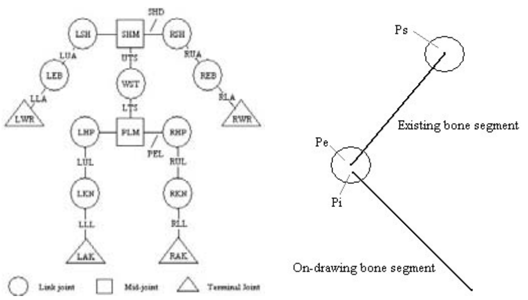
Figure 3: *The sketching interface with template skeleton and free-hand figure drawings: The on-drawing segment is recognized and highlighted on the template. Foreshortened and non-foreshortened segments have been displayed in black and green respectively. Perspective effects can be rendered incrementally by multiple strokes.*

Given a starting part for skeleton drawing (default as shoulder), the other body parts can be recognized in real-time and highlighted on the given template when they are still being drawn (See Figure 3). Length control is implemented based on the retrieved bone (true) length. Once a bone segment is being stretched to its extreme length (or say "becoming non-foreshortened"), the stroke will be displayed in a different color to indicate this status. Once a bone segment has been drawn-up, system will automatically record its features (including *bone name/2D drawn length*, *foreshortened/non-foreshortened status,* *ending joint name/XY coordinates*, etc). Our current body part auto-recognition works well with any sequence of skeleton drawing, based on the fixed human body topology.

3.2 Recognize on-drawing body part

An on-drawing body part is recognized during sketching, based on the human body topology. Currently, we classify 13 body joints into two categories: *link joint* (9 in total) and *terminal joint* (4 in total). For recognition, we have also defined two additional joints (middle point of *shoulder* and *pelvis*), and categorized them as *mid-joint*. Figure 4(a) gives these categoried joints, which are presented in different shapes. As defined, *Link joint* is the joint which connects two neighbour bone segments (such as *elbow/knee* joint, *shoulder* joint, etc.). *Terminal joint* means the joint (such as *wrist* and *ankle* joint), which is located at one end of bone segment and has no other extension based on our current stick figure model. We set activation flag for each joint; only the joint in active status can be linked with another bone segment.

In general, the first part (default as shoulder) is specified with all link and/or middle joints active. Afterwards, the *body part recognition* starts when a new line segment added. The system firstly gets the initialzing point *Pi* of this line, then checks if it is close to any existing (activated) body joint (See Figure 4(b)). This is operated by searching through each of the drawn bone segment(s), and verifying if *Pi* is within the pre-defined bounding area of its active joints such as *Ps* or *Pe*. If yes, the system retrieves the name of the connected joint/bone segment and classifies the joint into its corresponding category. If the connecting joint is a *terminal joint*, the recognition process will automatically terminate since no extension part is permitted. Otherwise, known a *link* or *middle* joint and its connected bone segment, the on-drawing body part can be easily estimated based on the body topology (See Figure 4(a)). The associated joints can be labeled according to their relative positions. Once a bone segment has been drawn-up, current connecting joint will be set as inactive to avoid the invalid connection.



(a) (b)

Figure 4: (a) *Abstracted human body structure labelled by the abbrieviations of bones and joints.* **(b)** *On-drawing body part recognition.*

3.3 Record bone/joint thickness value

In our system, we extract and process *bone/joint thickness contrasts* from figure sketches as main clues for depth indication. Since rendering strokes are normally located around a specific bone segment/joint to make the visual effects, we define a bounding (offset) area for each bone/joint and simply count the number of strokes inside the pre-defined areas to get the corresponding thickness values. Whenever a new rendering stroke added, system will firstly check if it is located in any of the pre-defined bounding areas of the on-drawing figure. If so, system will automatically update the corresponding (bone/joint) thickness value, as well as tidy-up the indication stroke by fitting it into a straight line. Otherwise, it will be left on interface as it was, and not cause a further processing. The thickness contrasts of bones/joints will be extracted and utilized later for 3D pose recovery, until a figure drawing completed.

4 Reconstruction of 3D Stick Figure

In this section, we present our 3D pose reconstruction strategy and algorithms, which associates with an "overall pose checking/auto-correction" process.

4.1 Back-front ambiguity involved 3D pose reconstruction

As previously discussed, recovering a 3D pose from its 2D measurements is challenging because of "back-front ambiguity". Assuming a scaled orthographic projection, the 2D image coordinates (x, y) can be related to 3D world coordinates (X, Y, Z) through a matrix transformation in Eq. (1):

$$\begin{pmatrix} x \\ y \\ 0 \end{pmatrix} = s \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad (1)$$

Where s is scale factor. Thus given two end joint **p1**=(x1, y1) and **p2=**(x2, y2) of a drawn bone segment of and the corresponding bone true length l, the relative depth (dZ) between **P1** and **P2** can be computed by Eq. (2):

$$dZ = \sqrt{l^2 - ((x1 - x2)^2 + (y1 - y2)^2)/s^2} \quad (2)$$

Where the two possible solutions represent the "back-front ambiguity" we have mentioned. Thus, given a figure with n foreshortened bone segments, the possible 3D body configurations would be 2^n. The identification of a unique pose from this huge set of candidate poses is a problem.

Three different approaches have been addressed to solve this problem. Instead of the fully manual [HP92][Tay00][FA03] or automated methods [DCR99][LC85][GMHP04], we chose to follow a semi-automated [DAC*03] approach, which is more suitable for our subject. From a large amount of observation and analysis of our user survey results, it has been found out that figure sketch with perspective renderings can explicitly reveal the drawer's intended pose. Meanwhile, it facilitates the understanding of viewers on a given pose as well. Thus, we developed our multi-layer back-front ambiguity clarifier. It utilizes the information from user's sketches and a set of constraints/assumptions, to clarify the orientation ambiguity of each foreshortened bone segment. Once each of these bones becomes clearly posed, the final 3D figure pose is achieved. In our system, the *s* is set as 1.0.

4.2 Multi-layered back-front amibiguity clarification

When a reconstruction process starts, the system begins by checking and getting a list of all foreshortened (orientation uncertain) bone segments. Then, they will be imported into the above-mentioned *multi-layer back-front ambiguity clarifier* (See Figure 1). When processed through this multi-layered clarifier, foreshortened bone segments get clarified gradually layer by layer. In our system, there are currently three clarifying layers supported by both user's sketching and a series of constraints/assumptions. The top layer is designed to determine all foreshortened bone segments, which can be posed by user's perspective rendering (bone/joint thickness contrast). Remaining segments will then be passed down to the second layer, which utilizes *Joint ROM* (Range of Motion) constraints to further clarify the uncertain bone segments. After that, all uncertain segments (if any) will be passed to the final *key frame coherence checking* layer, which identifies orientations by referring to the coherence between neighbouring key frames (given some reconstructed key frame drawings as prerequisite).

After this clarification process, the system checks again for the posing of foreshortened bone segments. If there are still some bone segments keeping orientation uncertain, they will be highlighted to users for further depth indication. The clarifying process will start again when users give more sketch input and select to do so.

4.3 Utilization of bone/joint thickness contrast

As previously mentioned, thickness values of bones/joints can be indentified from 2D sketch by simply counting the rendering strokes. Since the style of rendering varies in individuals, the depth meaning implied by a given thickness contrast is not unique. To handle this, we have generalize a set of rendering gestures, which is easy to master and efficient for pose inferrence by system. In essence, we arrange bones and joints into different groups and correlate thickness contrast features to depth relationships. Generally, the thicker one is closer. Figure 5 gives an illustration for the grouping scheme.

Currently, we group bone segments into 7 pairs (P1-P7). Pairs P1 to P4 comprise neighbouring bones within each limb; P5 comprises upper and lower torso; and P6 and P7 contain two symmetric upper limbs (i.e. left and right upper arms). Correspondingly, we arrange joints into 7 groups (G1-G7). For Group G1 to G5, every three joints within each bone pair (P1 to P5) automatically form one group. G6 and G7 are special cases, where two ending joints of shoulder (or pelvis) are settled as one group alternatively.

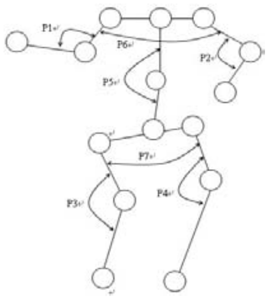


Figure 5: *Bone and joint grouping scheme.*

Generally, for pairs P1 to P5, the bone thickness contrast between each paired bones determines the spational relationships of their involved three joints. If a bone is identified as visually thicker, we assume that every point of this bone is located in front of those on its pair bone (vice versa). Thus, the far end (joint) of the thinner bone is located as the furthest, followed by the link joint in the middle, then the end joint of the thicker bone as the closest (See Figure 6(a)-(c)). For pairs P6 and P7, the thickness contrast between two symmetric upper limbs determines their linked shoulder (or pelvis) orientation. That is, when an upper limb is visually thicker than the other, the link joint between it and its connected shoulder (or pelvis) is closer than the link joint on the other side (See the pelvis orientation in Figure 6(a)).

However, in the case that two paired bones are not both stretching either forward or backward, bone thickness contrast becomes unclear, and the contrast among joints tends to be crucial for depth identification. For groups G1 to G5, we compare the thickness value among each three (end-link-end) joints within a group. Thus, if the link joint has the maximum thickness value, it will be considered as completely in front of the other two end joints (See Figure 6(c)). Otherwise, we assume that the middle joint is behind the others. As for groups G6 and G7, we simply compare the thickness between two end joints of shoulder (or pelvis) to identify its orientation. As previously mentioned, the thicker joint will be spationally closer.

Considering potential conflicts when using bone and joint thickness contrasts together, we check and utilize the bone thickness contrast first, then the joint's. When no thickness contrast can be recognized between paired bones, joint contrasts will be processed alternatively.

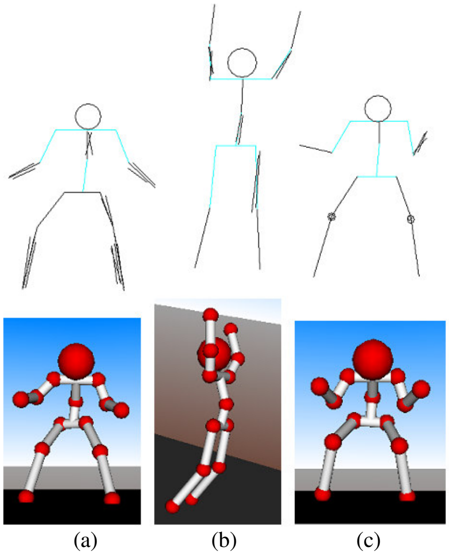


Figure 6: *2D key drawings of a jumping motion and their corresponding 3D models.*

Our informal user tests showed that the current rendering (thickness contrast) gestures are easy to master and handy to use. The resulting 2D drawings provide a better perception of 3D motions, comparing with the single stroke figure drawings. Moreover, to specify the 3D pose of a figure containing 12 foreshortened bone segments (the worst case), averagely only 6 rendering strokes are required if without referring to the joint ROM and frame coherence. It is efficient comparing with others, such as [HP92], which requires 12 indications to get to the final pose. Obviously, the number of strokes could be extraordinarily reduced if referring to the other two factors for pose identification. In practice, users may draw more rendering strokes than required for better visual effect.

4.4 Utilization of joint range of motion constraints and key frame coherence

After considering bone/joint thickness contrast, we apply physical constraints of ROM (Range Of Motion) on each human body joint to clarify bone orientation ambiguity. In our skeleton model, there are basically four categories of joints: hinge (elbow/knee), ball-and-socket (shoulder/hip), irregular (wrist/ankle) and torso link joint (pelvis joint). Each of them has its specific range of motion (angle constraints), which can be used to cull the invalid bone orientation. Various approaches have been investigated to apply joint ROM constraints. Here, we follow the method in [LC85], and use their joint ROM meaturements.

Figure 7 shows an abstraction of human body as a hierarchy tree structure, where upper torso (containing SHD and UTS) and lower torso (containing PEL and LTS) have been assumed as rigid planes. Given an (uncertain) foreshortened bone segment, its two symmetrical orientations (considering the "orthographic projection model") will be both evaluated against the joint ROM between it and its parent bone. Then, if only one of those two candidate orientations is valid, this foreshortened bone gets clearly posed. Otherwise, it remains orientation uncertain. Moreover, when a parent bone is uncertain too, its orientation should be identified first by referring up to its own parent node (and so on). Once the parent bone gets clarified, a reverse process will start, to apply this orientation down to determine its child's. For the upper (lower) torso plane, the potential orientations could be up to four (when its two component segments are both uncertain). Meanwhile, we evaluate the balance of body as well [DAC*03]. Figures 6(b) and (c) illustrate how joint ROM has been used for pose recovery. In Figure 6(b), the left lower leg has been identified as pointing backward because the opposite case is against the ROM for knee joint. In Figure 6(c), the upper torso segment and left lower arm have been similarly posed by utilizing joint ROM constraints.

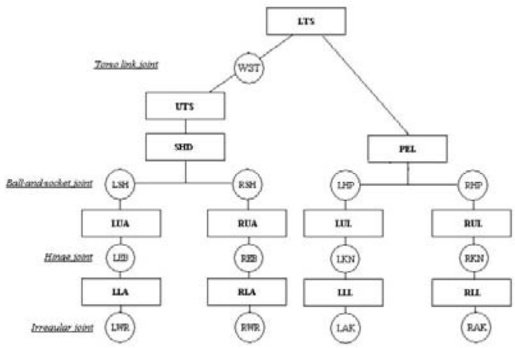


Figure 7: *Human body hierarchy tree structure*

After checking the range of motion, the left orientation uncertain bone segments will go through the final *key frame coherence checking* layer. Here, we follow the method [CL92] for coherence checking. Since keyframe figure drawings express a continuous motion over time, coherences are expected between one frame and its previous frame. Thus, given an uncertain bone segment in frame *k*, the angular difference between its previous direction (in reconstructed frame *k-1*) and its two new candidate directions will be computed. The directionally closer one will be considered as the correct case.

4.5 Overall pose checking and auto-correction

Current multi-layer reconstruction routine clarifies "back-front ambiguity" of foreshortened bone segments. However, incorrectly posed figures may sometimes be created accidentally, by a quick and imprecise figure sketching. Therefore, an automated checking and correction process is designed; to detect all ill-posed body parts, highlight them and give proper corrections based on human body joint ROM and balance. In our system, users can choose to accept system's auto-correction or not. The final correction will be executed only if users select to follow it.

According to the predefined body hierachy, we process body joints layer up layer till the root (torso link) joint. Thus, *hinge* (elbow/knee) joints will be checked and adjusted first, followed by *ball-and-socket* (shoulder/hip) joints, and finally torso link joint, since the child bone movement will never affect its parent's. All corrected body joint coordinates will be reserved separately and finally applied after users' acceptance of the auto-correction.

For *hinge* joint processing (start from left elbow joint), we compute the current joint angle first, then evaluate it against the elbow *flexion/extension* angle range. If the current angle is within the valid range, the system will keep it, and proceed to check the next hinge joint (right elbow joint). Otherwise, the *dZ* value of lower arm is swapped to its opposite case (considering the "orthographic projection model"). The system will compute the joint angle, and check it again if this alternative angle fulfills the given range of motion. If accepted, the new absolute Z value is calculated for the related terminal (wrist) joint. Otherwise, a boundary correction process will start. That is, to adjust the given lower limb from its original position to the boundary position of the corresponding flexion/extension angle range. A sequence of transformation matrices is applied here for computing the new terminal joint position. After that, the system will move on to the other hinge joints, and handle them one by one following the same routine.

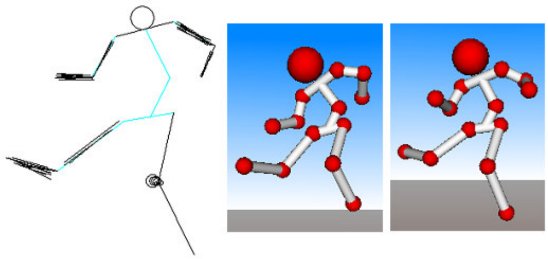


Figure 8: *An original drawing with its ill-posed and auto-corrected 3D figure models: The original joint angles of left knee and right elbow are both extension angles, which are not allowed for hinge joint. So, the corresponding lower limb is swapped to its opposite orientation. The upper torso is initially over-bent to the left, which cannot be corrected by swapping the upper torso direction. Thus, a boundary correction is executed to adjust it to the boundary position of the valid bending range.*

After *hinge* joint checking, a similar process for *ball-and-socket* joint will start, where two categories of angles (*flexion/extension*, and *abdustion/adduction*) should be satisfied. Meanwhile, joint coordinates updating needs to be performed on both hinge joint and terminal joint. At last, the torso link joint will be handled, where three categories of angles (*flexion/extension*, *bending*, *rotation*) are to be evaluated. The coordinate updating should be executed layer down layer till the leaf joint. After that, the whole checking/correcting process terminates. The system highlights all ill-posed body parts and requires user's permission for the final pose updating. Figure 8 illustrates a stick figure drawing, its ill-posed 3D model, and the adjusted model after an auto-correction process.

5 Figure animation and motion control in 3D

Once a series of reconstructed figure key frames have been obtained, the final 3D animation and motion control can be achieved by editing the overall motion path and keyframe timing via interactive sketching. A special routine is designed to improve the ground contact of key figures. Sound and background can be integrated to build a more realistic 3D world. The resulting animation is synthesized in VRML, and can be triggered by a single user click.

5.1 Motion path editing and timing control

In our system, the keyframe animation is defined in VRML by a series of *Transform* nodes, *OrientationInterpolators* and *PositionInterpolators*, timed by a *TimeSensor*, which generates events to control the animation. The motion path of an overall body can be defined graphically by drawing trajectory curves as in [Fox04] (See Figure 9). In our system, the body root has been defined as the mid-point of lower torso. If no motion path(s) specified, the stick figure will be located by its interface positions as default, where Z equals to zero.

As for timing, the system sets a default cycle length for each animation as 1×N seconds (N=the number of frames). This default value can be changed by users through adjusting an interface slider. By default, the overall time is evenly distributed over each keyframe. The system also provides a way for the users to 'act-out' the timing [TBP04][TM04] by drawing the last motion path with varied sketching speed. Alternatively, they can draw a separare timing curve to realize the control (See Figure 10).

5.2 Improve the ground contact and 3D virtual world

To ensure the proper ground contact of key figures, we implement a *ground snapping* routine. Assuming a body never moves below the ground, the lowest point among all key drawings defines the base position. Then the vertical distance between ground line and each key figure bottom point will be evaluated. If the offset distance is within the pre-defined tolerance range, the body will be moved down to snap with the given ground. Otherwise, it will remain in the original position without contact adjustment. From Figure 10, we can see that this method effectively avoids the fluctuation of body when it contacts with the ground.

Sounds/music and panoramas, have been provided to enhance the 3D virtual world. Alternatively, users may import their own selections. The sound can be triggered by a touch sensor (See Figure 10) located inside the world. As moving through the world, the volume of sound varies according to the viewer's location.

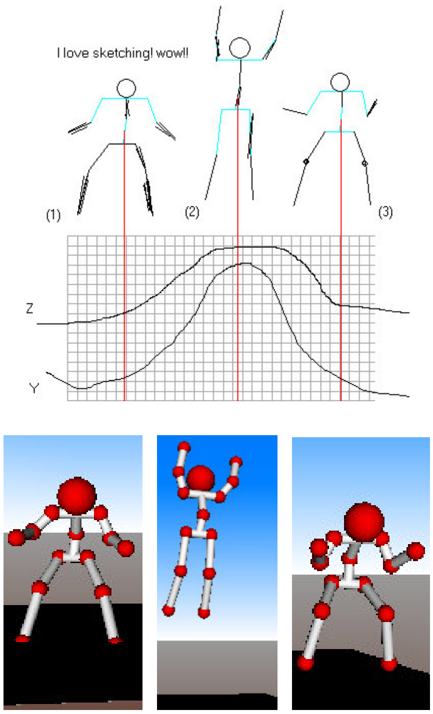


Figure 9: *Motion definition: Y and Z curves specify a motion, and the drawing speed of Y curve indicates motion timing.*

6 Implementation and examples

Our prototype modelling and animation system is implemented by Visual C++ program and VRML. It provides an intuitive sketching interface generated by Visual C++, which handles 2D figure sketches, 3D pose recovery, and animation data preparation. The resulting 3D figure model and animation are synthesized by VRML, which enables building 3D virtual world on the internet. The system has been tested on a variety of input devices: electric whiteboard, tablet PC, as well as a standard mouse.

The current system supports an interactive design process, through which 3D figure models can be viewed and continuously updated responding to user's incremental sketching. (See Figure 11) This has been shown to be intuitive and natural for rapid prototyping/evaluating 3D figure models and the final animation. Different users have used our sketch interface to create 3D animations. Although varying in ages and background, they enjoy this interactive process, and feel ease and fun to "pop-up" a live 3D animation within several minutes. Some animations have been shown in this paper together with their original drawings (See Figure 9-10, 12).

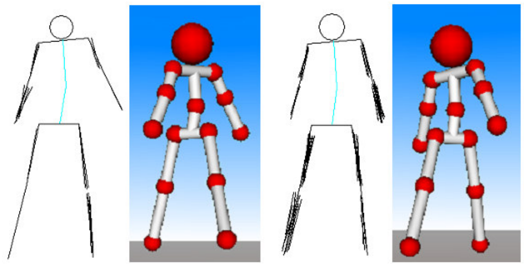


Figure 11: *(a) The initial drawing and its corresponding 3D model. (b) After incremental sketching, the figure pose is changed based on the modified perspective rendering.*

7 Conclusions and future work

In this paper, we have presented a novel sketch-based gesture interface for 3D stick figure animation. The highlight of this interface is that it supports a natural and incremental sketching process and employs sketch itself to infer 3D poses. Moreover, an on-line drawing assistance is offered to maintain a right figure proportion. The animation synthesis is simple and straightforward. Although the resulting animation in VRML is simple and rough, it is still vivid and inspiring for users to pop-up their 3D characters from 2D sketches.

More recently, the fast development of powerful superworkstations has led to new areas such as multi-media, interactive games and Virtual Reality, where interactive and real-time animation has become a key issue [CM01]. Although current professional packages produce highly attractive and entertaining 3D animations, viewers are still treated as audiences. While impressed by the marvelous visual impact, they are rarely given opportunities to get involved. Thus, it is believed that we provide an interactive and approachable tool to address this need. Users can create and drive their own characters by only 2D sketching. In their imaginary 3D world, they become creators, or even actors. This interactive process greatly inspires user creativities, and makes 3D animation more accessible and enjoyable for everybody.

Comparing with other related systems [DAC*03][HH01][TBP04], our system features by a better utilization of sketching information (such as thickness contrast) to assist the general pose identification. Although perspective rendering is more likely to be used in full figure drawing rather than stick figure, its application in our current system has shown the feasibility of recovering 3D pose and generating 3D animation from pure 2D sketches. The resultant sketching interface is acceptable, as well as the natural and sketchy stick figure drawings. The investigation of perspective rendering into full figure drawing and the sketch-based skin suface modelling are our future works.

References

[ANM97] AMES A. L., NADEAU D. R., MORELAND, J. L.: *VRML 2.0 Sourcebook.* 2nd edition, John Wiley & Sons Inc., (1997).

[Bro90] BROWN M.H.: *Bodyworks: A visual guide to drawing the figure*, North Light Books, Cincinnati, Ohio, U.S.A. (1990)

[Cho04] CHOI J. J.: *MAYA character animation.* 2nd edition, San Francisco, Calif.: London: SYBEX (2004).

[CL92] CHEN Z., LEE H.J.: Knowledge-guided visual perception of 3-D human gait from a single image sequence. *IEEE Transactions on Systems, Man, and Cybernetics* 22, 2 (1992), 336- 342.

[CM01] CANI M. P., MAGNENAT-THALMANN N., THALMANN D.: *Computer Animation and Simulation*. Springer, (2001).

[DAC*03] DAVIS J., AGRAWALA M., CHUANG E., POPOVIĆ Z., and SALESIN D. **A** sketching interface for articulated figure animation. In *Proc. Eurographics / SIGGRAPH Symposium on Computer Animation* (2003), 320-328.

[DCR99] DIFRANCO D. E., CHAM T. J., REHG J. M.: Recovery of 3D articulated motion from 2D correspondences, Compaq Cambridge Research Laboratory, Cambridge, MA (1999).

[FA03] FABIO R., ANDREAS R.: 3D reconstruction of human skeleton from single images or monocular video sequences, *25th Pattern Recognition Symposium, DAGM 03*, Magdeburg, Germany, (2003), 100-107.

[Fox04] FOX B.: *3ds max 6 Animation: CG Filmmaking from Concept to Completion*. McGraw-Hill Osborne, California, U.S.A. (2004).

[GD96] GROSS M. D., DO E.Y.L.: Ambiguous intentions: A paper-like interface for creative design. In *Proc. UIST'96* (1996), pp.183-192.

[GMHP04] GROCHOW K., MARTIN S. L., HERTZMANN A. and POPOVIC Z.: Style-based inverse kinematics. *ACM Transactions on Graphics (TOG) 23*, 3 (2004), 522- 531.

[HH01] HOSHINO J., HOSHINO Y.: Intelligent storyboard for prototyping animation. In *Proc. IEEE Int. Conf. On Multimedia and Expo* (2001), Conference CD-ROM FAI.03.

[HP92] HECKER R., PERLIN K.: Controlling 3D objects by sketching 2D views. *SPIE – Sensor Fusion V 1828* (1992), 46-48.

[IMT99] IGARASHI T., MATSUOKA S., TANAKA H.: Teddy: a sketching interface for 3D freeform design. In *Proc. SIGGRAPH '99* (1999), 409-416.

[Li05] LI W.: Figure drawing: Basic pose and construction, FARP,http://elfwood.lysator.liu.se/farp/figure/williamlibodyconstruction.html (2005)

[LC85] LEE H. J., CHEN Z.: Determination of 3D human body postures from a single view, *Computer Vision, Graphics and Image Processing 30* (1985), 148-168.

[LM95] LANDAY J.A., MYERS B.A.: Interactive sketching for the early stages of user interface design. In *Proc. CHI'95* (1995), pp.43-50.

[MCT*99] MARKOSIAN L., COHEN J. M., THOMAS C. , HUGHES J.: Skin: a constructive approach to modeling freeform shapes. In *Proc. SIGGRAPH '99* (1999).

[MCMK97] MORAN T.P., CHIU P., MELLE W.V., AND KURTENBACH G.: Pen-based interaction techniques for organizing material on an electronic whiteboard. In *Proc. UIST'97* (1997), pp.45-54.

[MQ04] MAO C., QIN S. F.: User survey study internal report. Brunel University, Uxbridge, U.K. (2004).

[QWJ00] QIN S. F., WRIGHT D. K., JORDANOV I. N.: From on-line sketching to 2D and 3D geometry: a system based on fuzzy knowledge. *Computer-Aided Design 32* (2000), 851-866.

[SS02] STROTHOTTE T., SCHLECHTWEG S.: *Non-photorealistic computer graphics – modelling, rendering and animation*, Morgan Kaufmann Publishers, San Francisco, U.S.A. (2002).

[Sut63] SUTHERLAND I.E.: Sketchpad: A man-machine graphical communication system, Pro. SFIPS Spring Joint Computer Conf. IFIP, vol.23 (1963) pp.329-345.

[Tay02] TAYLOR C. J.: Reconstruction of articulated objects from point correspondences in a single uncalibrated image. *Computer Vision and Image Understanding 80*, 3 (2000), 349-363.

[Tin92] TINER R.: *Figure drawing without a model*, David & Charles plc, Glasgow (1992).

[TBP04] THORNE M., BURKE D., VAN DE PANNE M., Motion doodles: An interface for sketching character animation. *ACM Transactions on Graphics (TOG) 23*, 3 (2004), 424- 431.

[TD02] TILLEY A. R., DREYFUSS H.: *The measure of man and woman: human factors in design*, revised edition, Wiley, NJ, U.S.A. (2002).

[TM04] TERRA S. C. L., METOYER R. A.: Performance timing for keyframe animation. In *Proc. Eurographics/ SIGGRAPH Symposium on Computer Animation*, (2004), 253-258.

[WIK05] WIKIMEDIA FOUNDATION, INC.: Traditional animation information, http://en.wikipedia.org (2005).

[ZHH96] ZELEZNIK R. C., HERNDON K. P., HUGNES J. F.: SKETCH: an interface for sketching 2D scenes. *Computer Graphics Proceedings, Annual Conference Series* (1996), 163-170.