Methods for Fast Skeleton Sketching

Martin Poirier* Eric Paquette†

Multimedia Lab, École de technologie supérieure, Montreal, Canada

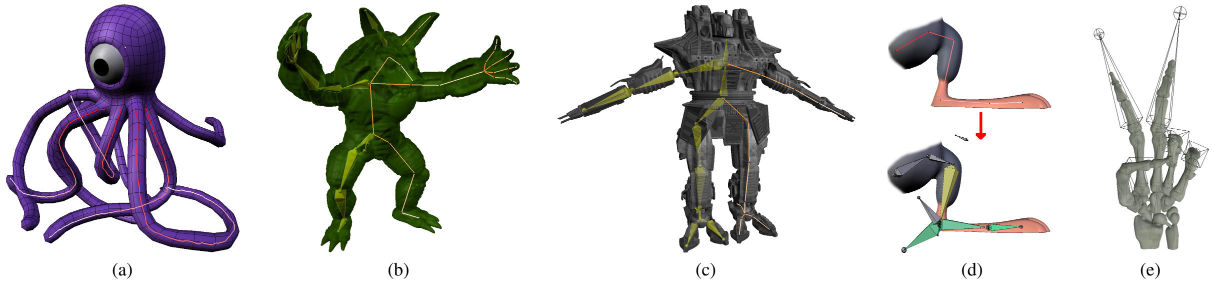

Figure 1: *Sketched skeletons on organic (a-b) and mechanical models (c). Template retargetting (d-e)*

1 Introduction

Rigging characters for animation is a tedious task traditionally accomplished by extrusions of joints and direct bone manipulations. This work introduces different methods and concepts for a fast and effective sketching interface for skeleton creation. Unlike previous work on sketching motion paths [Thorne et al. 2004] or spline deformation [Blanco and Oliveira 2008], this is a direct way to create and adapt complex deformation skeletons on any character.

2 Sketching and Gestures

As can be seen in the accompanying video, the sketching interface provides several traditional 3D sketching tools for freehand strokes or polylines. Overdrawing (sketching on top of an existing stroke) can be used for simple stroke corrections. Gestures are also provided to cut, trim or delete strokes.

Embedding is used to facilitate sketching by automatically positioning strokes inside limbs. This is done through a variant of the classic depth peeling algorithm, which can be applied inter objects or intra object. The first is better suited for mechanical parts (Fig. 1 c) while the second is more geared toward organic shapes (Fig. 1 a-b).

3 Bone Creation

Next, strokes have to be converted into skeletons. Joints are inserted at vertices of polyline strokes while automatic subdivision is used to insert joints along freehand strokes. Rules are available for subdividing into a fixed number of bones, subdividing into bones of fixed length or adaptively adding more bones in areas of strong curvature. The resulting bones are fully oriented in 3D space using the direction of the stroke and the orientation of the viewport when it was drawn.

Template retargeting (Fig. 1 d-e) is useful for complex and commonly used setups, and works by adapting a user-created skeleton to a stroke. This is done by minimizing the penalty function of Eq. 1, controlling for fitness to the original skeleton and fitness to the stroke.

$$weight = \sum_{i=1}^{t-1} \underbrace{\gamma_\theta \theta_i}_{\text{angle}} + \sum_{i=1}^{t} \underbrace{\gamma_l l_i}_{\text{length}} + \underbrace{\gamma_x x_i}_{\text{distance}} \qquad (1)$$

If the skeleton contains control bones (inverse kinematic solver, pole target, action control, *etc.*), the retargeting process automatically repositions them with respect to the appropriate deformation bone.

4 Results and Conclusion

Our prototype was implemented in Blender (blender.org) and was tested by a number of professional users. Testers had to accomplish predefined tasks corresponding to different types of rigging work.

Of all the bone creation methods, users found the templating techniques most useful, followed by the length and fixed subdivision techniques. Templating was found to be particularly good at reducing the time needed for rigging (by an estimated 10 to 50%) in cases where many similar limbs were needed, such as when rigging fingers or insect legs. The interface was seen to be more natural and fluid than traditional methods, especially for tablet users.

We demonstrated that a sketching metaphor can be adapted to deformation skeleton creation in order to definitely speed up the creation and adaptation process in a fluid and effective way. Our implementation can rig typical characters with the use of an effective stroke embedding method and with automatic template retargeting.

We would like to thank ProMotion studios, Stanford University Computer Graphics Laboratory, and Jean-Sébastien Guillemette for models and skeletons as well as everyone who participated in the usability tests.

References

Blanco, F. R., and Oliveira, M. M. 2008. Instant mesh deformation. In *I3D '08: Proceedings of the 2008 symposium on Interactive 3D graphics and games*, ACM, 71–78.

Thorne, M., Burke, D., and van de Panne, M. 2004. Motion doodles: an interface for sketching character motion. *ACM Trans. Graph. 23*, 3, 424–431.

*e-mail: martin.poirier.4@ens.etsmtl.ca

†e-mail: eric.paquette@etsmtl.ca