CurveThis: A Tool to Create Controllable Massive Crawling

Pei-Zhi Huang†, Tsai-Yen Li‡
† Base FX Studio, ‡National Chengchi University

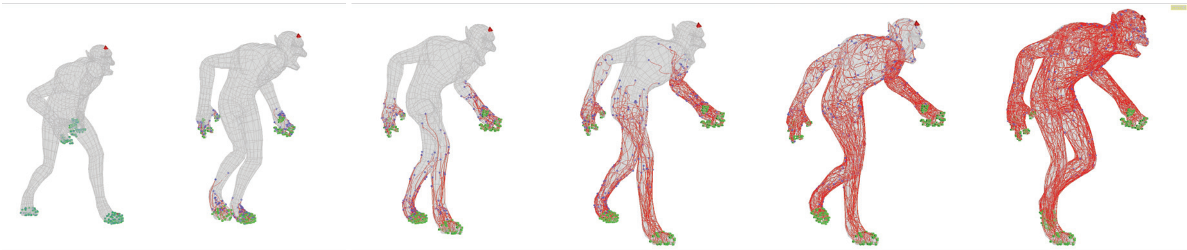

Figure 1. The controllable massive crawling effect on the body surface of an animated character. Green dots: starting points; Red triangles: goal points; Blue points: current endpoints of curves; Red curves: trace of crawling curves.

1. Introduction

Forming the shape of an animated character from hundreds of thousands of crawling bugs, as shown in the film of *The Sorcerer's Apprentice*, is a challenging shot for most animators. It is desirable to have the massive motions of the crawling bugs or growing vine generated automatically by procedures to avoid time-consuming key-framing. On the other hand, the motions also need to be controllable to achieve specific artistic effects. In other words, a desirable tool for creating this kind of animation should allow the director to specify desirable visual effects such as heading toward common destinations and following surface constraint, and then the animation should be generated automatically.

Several approaches had been proposed to solve the problem. For example, one can divide the creeping bugs into layers along the surface of a model to reduce the complexity of resolving intersections [1]. The same visual effect may be achieved by composing several layers into one. However, due to the speed variation of each vertex on a character model in an animation, the simulation along the surface may suffer from sliding or popping artifacts. Another way to avoid these artifacts is to perform the simulation in the UV texture space and then transform the simulation results to the world space for rendering. Although the method is straightforward and creates smooth motions, there is no easy way to know how to design UV layouts to achieve a specific visual effect.

2. Generating Curves on Surfaces

Our approach aims at facilitating the creation of these massive bug animations by generating goal-directed curves along the surface of a character model and allowing the dispersion of the curves to be controlled by the director. This approach consists of three phases: *gradient graph creation*, *path planning*, and *curve generation*.

Gradient Graph: Topological connectivity is an invariant of a geometric model in an animation. A cyclic connectivity graph *C* can be constructed by explicitly building the neighboring relation for each vertex on a mesh. On a uniform mesh, the number of edges between two vertices can be used as an approximation of Manhattan or Euclidean disctance on the surface. The connectivity graph can then be used as the workspace for finding a feasible path. One can define obstacles, such as protruding objects, in the workspace to prevent bugs from crawling over. The gradient field in the workspace can then be computed with the wave propagation algorithm (NF1) commonly used in the potential-field based motion planners by propagating a gradient field from the goals [2].

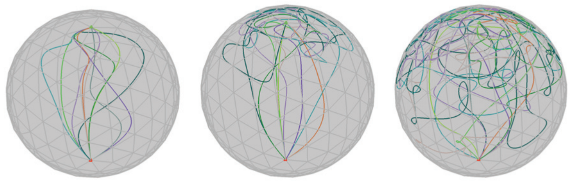

Figure 2. Path generation with different control parameters. Left: preferring less visited nodes; Middle: with coverage preference; Right: with Brownian motion

Path Planning: The gradient field is a directed acyclic graph that can be used to guide the search in the path planning phase. Unlike other path planning problems where a shortest path is desired, the path along the surface of a model in our tool should not only follow the gradient field but also satisfy physical constraints (such as linear and angular velocities, etc.) and respect design preferences (such as coverage, sibling order, etc.) for artistic control. In Figure 2, we show three examples of paths resulting from different control parameters for optimizing certain criteria.

Curve Generation: A generated path consists of a sequence of vertices on *C*. We then take these via points as the control points to generate smooth B-spline curves on or above the world space. Although the curves could be generated with dispersion preference, intersection between them may not be avoided. In case of collision, we displace the control points along the vertex normal to resolve the problem.

3. Conclusion

We have developed a tool, called CurveThis, which can generate goal-directed curves on animated model's surface which satisfy physical constraints and provide artistic controls. We use a gradient field to guide the search for a collision-free path, which is used in turn to generate a smooth curve above the model's surface. The efficiency of the method allows the users to interactively tune the design parameters and preview the generated results in real time.

References

DUNLOP, R. 2006. Sorcerer's Apprentice. Available at http://www.cgsociety.org/index.php/CGSFeatures/CGSFeatureSpecial/sorcerers_apprentice

LATOMBE, J.C. 1991. *Robot Motion Planning*. Kluwer Academic Publishers.

SIGGRAPH 2012, Los Angeles, California, August 5 – 9, 2012.
ISBN 978-1-4503-1435-0/12/0008