Generalizing Multi-Touch Direct Manipulation

Jason L. Reisman        Philip L. Davidson        Jefferson Y. Han
Perceptive Pixel, Inc
jason@perceptivepixel.com

Introduction

The appeal of direct manipulation with multi-touch interfaces stems from the experience it offers. As the user slides their fingers along a touch surface, objects react by rotating, translating, and scaling themselves so that the same point on an object always remains underneath the same fingertip. Since objects move in a predictable and realistic fashion, users are given the impression of “gripping” real objects. Direct manipulation essentially provides an intuitive and controllable mapping between points in local space and points in screen space, without the need for any explicit gesture processing.

Recently we have seen an explosion in the number of multi-touch applications which support 2D direct manipulation. In addition, many ways exist to manipulate 3D objects in a multi-touch environment. However, as far as we know none of them provide direct control. This is in contrast to what many users have come to expect when manipulating 2D objects such as photos, maps, documents, etc. In this talk we describe a general method for performing direct manipulation of 2D as well as 3D objects on a multi-touch surface.

Method

Our method supports direct manipulation in both 2D and 3D by minimizing a quadratic energy function which attempts to match points in the object’s local space to pixels in screen space. We employ a transform similar to the quaternion camera model described in Gleicher et al.. Unlike Gleicher et al, however, we do not solve for the derivatives of transform values in time. Rather, we solve for the transform values which directly minimize the energy. By doing so, we avoid the issues which arise when forward integrating the transform derivatives in time.

Results

Our method provides a controllable mapping between points in the object’s local space and pixels on screen. It supports translation in 3D as well as rotation about arbitrary 3D axes. When applied to 2D contexts, our method completely encompasses the semantics of the standard 2D multi-touch rotate and translate (“RNT”) manipulators. Nearly all modes of 2D RNT interaction are easily replicated simply by restricting the set of DOFs that our method is allowed to operate upon. Our approach also naturally extends RNT into 3D, producing exact solutions for interactions up to three contact points, and best-fit solutions when used with four or more points. Constraining DOFs in 3D allows for a great deal of interaction flexibility as well.

Upon testing our method shortly after our initial implementation, two interesting and useful bimanual interactions emerged. We quickly noticed that any two fingers can be used to define an axis, while a third finger can be used to control how much to swing the object about that axis. (In reality it turns out that the axis is not constant, but varies to compensate to perspective effects.) Because our method can match three points exactly, the user has total control over the object’s new orientation. Such three point rotations thus provide an easy and intuitive mechanism for users to orient objects in arbitrary ways.

The second, and perhaps more novel interaction, is a 4 point interaction which we call a perspective rotate. In this interaction the user begins by placing four fingers on the object in roughly rectangular configuration. The user then decreases the distance between two of the contact points while simultaneously increasing the distance between the other two. The object then rotates to best achieve the perspective described by the new hand configuration.

Discussion

Initial testing also revealed some unexpected and sometimes unintuitive behavior. Because our method supports rotations about axes parallel to the image plane, rotational ambiguities may occur. That is, although our method always produces valid solutions, the energy minimization may favor rotating the object out of the screen when the user expects the object to rotate into the screen (or vice-versa). Additional unintuitive behavior may arise when the user exhausts a particular degree-of-freedom (DOF) during an interaction and is surprised when the minimization starts to engage another DOF which it has previously left relatively untouched. This is particularly noticeable when the object has been rotated out of the screen until the normal at the contact point is practically parallel to the camera’s Z axis. The user then cannot continue to move the points apart without either engaging translational or rotational DOFs about a different axis.

Despite these difficulties, which we believe are mitigable, there likely exist many interesting applications for 3D direct manipulators.

References