Integrated Manipulation:
Context-aware Manipulation of 2D Diagrams

Masaaki Honda, Takeo Igarashi, Hidehiko Tanaka, Shuichi Sakai
The University of Tokyo, Hongo 7-3-1, Bunkyoku, Tokyo, Japan
Tel: +81-3-3812-2111 ext 7413
E-mail: {honda,takeo,tanaka,sakai}@mtl.t.u-tokyo.ac.jp

ABSTRACT

Diagram manipulation in conventional CAD systems requires frequent mode switching and explicit placement of the pivot for rotation and scaling. In order to simplify this process, we propose an interaction technique called *integrated manipulation*, where the user can move, rotate, and scale without mode switching. In addition, the pivot for rotation and scaling automatically snaps to a contact point during moving operation. We performed a user study is performed using our prototype system and a commercial CAD system. The results showed that users can perform a diagram manipulation task much more rapidly using our technique.

KEYWORDS: Drawing editors, CAD, direct manipulation, interaction techniques.

INTRODUCTION

Conventional drawing editors and CAD systems have a number of editing commands and modes. The user has to combine them appropriately to construct a precise diagram [2]. For example, if the user wants to construct a figure 3f starting from figure 3a, he must switch back and forth between move, rotate, and scale modes. In addition, the user has to place the pivot for rotation and scaling manually.

To be specific, we observed the following problems:
1. The user has to change the editing mode frequently.
2. Snapping [1] is restricted to the current editing operation. If the user is moving a diagram, it never rotates even if a small rotation can satisfy a geometric constraint.
3. The user has to place the pivot for rotation and scaling manually before the actual operation.

This paper introduces an interaction technique to solve these problems, and reports the results of our preliminary evaluation.
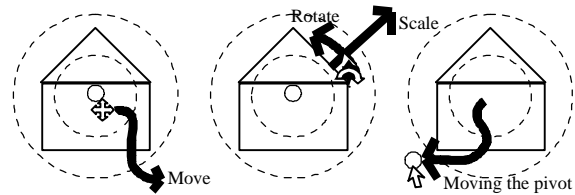

Figure1: Integrated Manipulation.
The user can move, rotate, and scale the diagram, and can move the pivot without mode switching.

INTEGRATED MANIPULATION

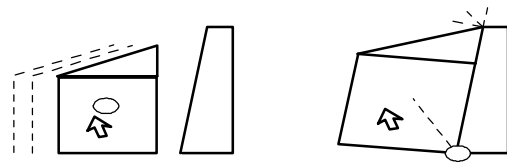Basically, integrated manipulation works as one editing

Figure2: The diagram rotates and scales to snap even while the user is moving it. The pivot automatically snaps to the contact point.

mode. The user first specifies this mode from the menu, and selects the target diagram. The user manipulates the selected diagram by dragging it. However, unlike standard editing modes, the user can move, rotate, and scale in this single mode[1]. In addition, the snapping works more flexibly, and the pivot for rotation and scaling snaps to a contact point automatically.

In the integrated manipulation mode, the user's dragging operation works differently based on the starting point's relative location to the target diagram (Figure 1). If dragging starts at the central part of the diagram, the system is in the moving mode, and the user can move (translate) the diagram. If dragging starts at the outer part of the diagram, the system is in the rotation and scaling mode. The user can rotate and scale the diagram simultaneously by single dragging. If the cursor moves toward or away from the diagram's center, the diagram scales. If the cursor rotates around the center, the diagram rotates.

---

[1] We assume one handed, single pointing device operation. Some systems [3] use two handed operation with a keyboard or another pointing device in the non-dominant hand.

a)Select b)Move c)Snap
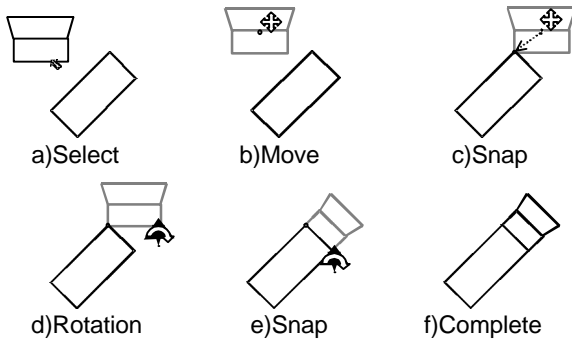
d)Rotation e)Snap f)Complete

Figure 3: Example diagram manipulation sequence.
a) The diagram is selected by clicking. b) It is moved by dragging the central part. c) It snaps to the other diagram. The pivot snaps to the contact point. d) It is rotated and scaled by dragging the outer region. e) It snaps to the other diagram. f) The user releases the diagram by clicking outside.

During these operations, the system constantly searches for possible geometric constraints. If the system finds a geometric constraint, it snaps the diagram to satisfy the constraint. The current implementation supports connections between vertices and edges. If necessary, the diagram automatically rotates or scales even if the user is only moving it (Figure 2). In order to assist typical operations, scaling snaps to specific scales (x0.5, x1, x2, etc.) and rotation snaps to specific angles (30, 45, 90 degrees, etc.). The user can rotate the diagram without scaling provided the scale is nearly eaqual to 1. The same applies to the rotation.

The pivot for rotation and scaling is indicated as a small circle. The pivot moves with the diagram when the user is moving the diagram. The user can manually move the pivot by dragging, and it snaps to the nearby vertex in the scene (Figure 1c). In addition, the pivot snaps to the contact point when the current diagram snaps to another diagram (Figure 2). If multiple contact points appear at once, the system selects one randomly. The pivot remains at that point until the user explicitly moves it, or until the diagram snaps into another point.

Figure 3 illustrates the diagram manipulation sequence using our technique. Note that this sequence requires much fewer clicking and dragging operations than existing drawing editors.

Some commercial drawing editors[4] allow the user to move and rotate (move and scale) in a single mode, but they do not allow the combination of the three. In addition, the pivot does not automatically snaps in existing systems.

EVALUATION
We performed the evaluation experiment on a PC running Windows 98. We used our prototype system (Java program) and a commercial CAD system (Vellum™ by Ashlar). 12 subjects participated in the experiment, and we observed
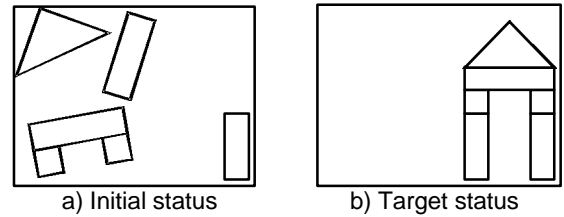

a) Initial status b) Target status
Figure 4: Diagram editing task used in the experiment.

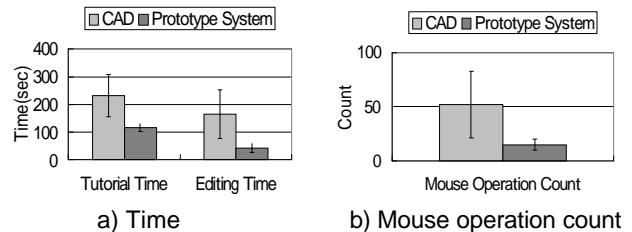a) Time b) Mouse operation count
Figure 5: The result of the experiment.

learning time and task completion time. First, we gave the users a brief tutorial of each system. For the CAD system, we taught the minimum operations necessary to complete the task (move, rotate, scale, and undo). After the tutorial, the subjects constructed Figure 4b starting from Figure 4a by rearranging the diagrams. Half of the subjects used the CAD system first, and others used the prototype system first.

Figure 5a shows the average time spent in the tutorial. Time spent in the tutorial for our system is significantly shorter than that for CAD system ($t_{(12)}=4.98, p<.0005$). Figure 5b shows the time spent and the number of mouse operations (click + drag) for completing the given task. The subjects completed the task significantly faster using our system ($t_{(12)}=4.52, p<.0005$), and the number of mouse operations were also significantly fewer ($t_{(12)}=3.92, p<.005$). These results show that our technique is effective for at least a certain class of tasks that involve move, rotate, and scale operations and require connection between diagrams.

LIMITATIONS AND FUTURE WORK
Flexible snapping can be annoying if the scene is complicated and many snaps occur simultaneously. The automatic snapping of the pivot might not be desirable for some users. Our current integrated manipulation does not support skew operation and scaling to one direction. In our future work we intend to work on these problems.

REFERENCES
1. Bier, E.A. and Stone, M.C.: Snap-Dragging, SIGGRAPH 86, pp.233-240, 1986.
2. S.K. Bhavnani, B.E. John, U.Flemming, The Strategic Use of CAD: An Empirically Inspired, Theory-Based Course, CHI'99, 1999.
3. Zeleznik, R.C., Forsberg, A.S., Strauss, P.S., Two Pointer Input for 3D Interaction, 1997 Symposium on Interactive 3D Graphics, pp. 115-120, 1997.
4. Smart Sketch™ FutureWave Software (http://www.futurewave.com)