Geometric Anticipation: Assisting Users in 2D Layout Tasks

Jessi Stumpfel
California Institute of Technology

James Arvo
University of California, Irvine

Kevin Novins
University of Auckland

ABSTRACT

We describe an experimental interface that anticipates a user's intentions and accommodates predicted changes in advance. Our canonical example is an interactive version of "magnetic poetry" in which rectangular blocks containing single words can be juxtaposed to form arbitrary sentences or "poetry." The user can rearrange the blocks at will, forming and dissociating word sequences. A crucial attribute of the blocks in our system is that they *anticipate* insertions and gracefully rearrange themselves in time to make space for a new word or phrase. The challenges in creating such an interface are three fold: 1) the user's intentions must be inferred from noisy input, 2) arrangements must be altered smoothly and intuitively in response to anticipated changes, and 3) new and changing goals must be handled gracefully at any time, even in mid animation. We describe a general approach for handling the dynamic creation and deletion of organizational goals. Fluid motion is achieved by continually applying and correcting goal-directed forces to the objects. Future applications of this idea include the manipulation of text and graphical elements within documents and the manipulation of symbolic information such as equations.

Categories and Subject Descriptors

H.5.2 [**Information Systems**]: Information Interfaces and Presentation—*User Interfaces*; H.1.2 [**Information Systems**]: Models and Principles—*User/Machine Systems*

General Terms

Algorithms

Keywords

Anticipatory interface, computer interface, eager recognition, fluid motion.

1. INTRODUCTION

In this paper we describe our initial experiments with a technique for assisting a user to interactively arrange two-dimensional information. We have created a prototype system that attempts to *anticipate* tasks that the user is about to perform, and to effect changes in advance that will assist

IUI'06, January 29–February 1, 2006, Sydney, Australia.
ACM 1-59593-287-9/06/0001.

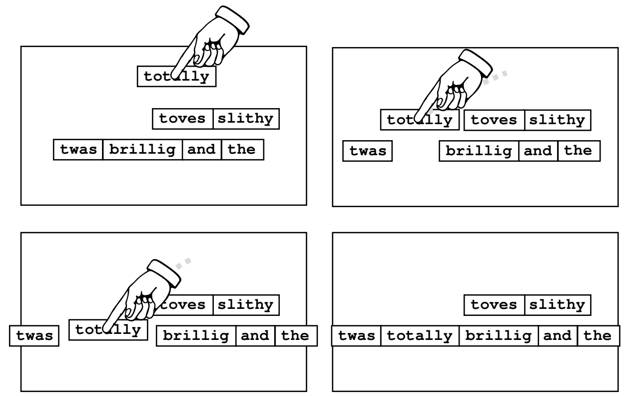

Figure 1: *The user selects and moves a magnet toward two abutting magnets. The system quickly makes room for the anticipated insertion, then aligns and joins the fragments once the piece is dropped.*

in performing the task. Specifically, our experimental system allows users to arrange words or phrases by selecting, dragging, and dropping them. The system exhibits anticipation in that it predicts when the user is attempting to insert a word or phrase into another and automatically makes room for it by rearranging the otherwise fixed words before the insertion occurs.

An inescapable property of anticipatory interaction is that the predictions must be based on scant and/or imprecise information. For example, if the system is to predict what arrangement the user is attempting to create, this prediction must be based on nothing more than the current arrangement and the recent history of the user's input, such as picking and dragging operations. Consequently, the system will invariably make some incorrect predictions. For such a system to be useful, the benefits conferred by correct predictions must outweigh the distraction and inconvenience of erroneous predictions. Errors must therefore be rectified quickly and unobtrusively. Our system achieves this by gracefully undoing changes it made that were never exploited by the user; that is, changes that were made as a result of apparently incorrect predictions. Smooth and natural-looking motion is achieved by continually applying and correcting goal-directed forces rather than instantaneously repositioning objects.

Our approach is analogous to "eager recognition" in which a gesture is recognized as soon as it becomes unambiguous [6]. However, our system reacts as soon as a given action is deemed *likely*, as changes are to take place in advance of the actual event. Fortunately, erroneous guesses are not troublesome in the layout task that we explore, as changes can be retracted easily and automatically by the system.

Previous systems that are somewhat similar to ours include Apple Computer's "Dock" interface [1] and Robertson *et al.*'s "Data Mountain" interface [5]. Both are capable of reacting to the insertion of new objects by rearranging existing objects. However, in both of these previous systems the reaction of objects depends exclusively on the current mouse position; recent history is not considered. Our approach differs in that future positions are anticipated by extrapolating from the cursor trajectory, which is estimated by analyzing the recent history of input events.

2. ANTICIPATION

To explore the idea of anticipatory interaction we have devised an application called *magnetic poetry*, inspired by the tiny magnets often found affixed to refrigerator doors. Each magnet contains a single word, collections of which can be arranged into phrases or sentences. The "magnets" in our system permit direct manipulation, just as their physical counterparts, but with one crucial difference; our system attempts to anticipate a user's intentions in several contexts, and to determine how the current configuration of magnets can best arrange themselves in preparation for the predicted action, as shown in Figure 1. (A demonstration of our prototype system is available at `www.ics.uci.edu/~arvo/demos`.)

As the time between meaningful events, such as selecting, moving, or inserting magnets can be quite short – on the order of tenths of a second – the time intervals for predictions are correspondingly brief. Moreover, the system must glean all information by sampling the cursor position alone, and attempt to predict what will happen, despite the fact that the user may be manipulating the magnets haphazardly.

Our system examines the recent history of user actions to estimate current object trajectories and thereby predict whether one or more magnets will collide. If so, the system also predicts where and when a collision will occur. At each time step, this prediction is based on a least-squares fit of positions sampled during the previous 500 milliseconds, which provides estimates of both direction and speed of the cursor as well as any magnets that are being moved. If the approximate trajectory indicates an imminent collision near the point at which two magnets abut, the system treats it as a likely insertion event. In this case, the system applies forces to the stationary magnets that will produce a gap at the right time to comfortably accommodate insertion of the moving magnets. We now describe how these forces are computed.

3. FLUID MOTION

When the system anticipates that a magnet will be inserted into an existing line of magnets, the stationary magnets will begin move, creating a gap at the predicted point of insertion with more than enough space to accommodate the new magnet or sequence of magnets. In forming the gap the moving magnets may also push other magnets to the side. The user can easily try the word in the new positions, moving it from place to place at arbitrary rates and times. The system attempts to assure that there is always adequate

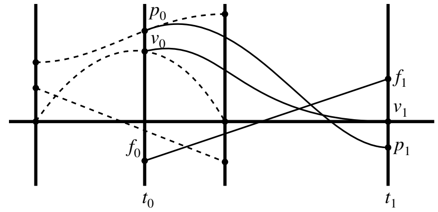

Figure 2: *Hypothetical plots of position, velocity, and force acting on a single magnet as a function of time. On the left are curves corresponding to one hypothetical goal. On the right are new curves corresponding to a goal with a new target position and time. New initial and final forces are chosen to meet the constraints, given the current position and velocity, while keeping the motion smooth. The current time becomes the initial time,* t_0*, of the new trajectory.*

space for the inserted magnet by the time it arrives, so the magnets accelerate and decelerate as necessary according to the expected time budget. Thus, a predicted insertion produces a change in other magnet positions by moving them out of the way quickly but not instantaneously, and damping their velocity to zero before the predicted insertion event. We refer to the target positions of the moved magnets and the associated time constraint as a *goal*. Each magnet may be assigned a single goal, which is generally replaced with a new goal, based upon more recent input events, before the current goal is actually attained.

To effect all necessary changes in position smoothly and naturally, we treat the magnets as though they have unit mass, and therefore inertia, and apply the appropriate forces in the spirit of teleological modeling [3]. In our context, linearly varying forces provide sufficient degrees of freedom to meet all goals. Supposing, for the moment, that the current time is $t_0 = 0$, and that the final time for a given goal state is $t_1 = 1$, we define the force function to be

$$f(t) \;=\; (1-t)f_0 \;+\; tf_1. \tag{1}$$

We then determine f_0 and f_1 such that when the resulting time-varying force is applied to a magnet, even one that is already in motion, the magnet will reach the desired position at the desired time. Integrating Equation (1) produces the velocity function $v(t)$, and integrating once more produces the position function $p(t)$. Thus, we have

$$v(t) \;=\; v_0 + f_0\,t + \frac{f_1 - f_0}{2}\,t^2 \tag{2}$$

$$p(t) \;=\; p_0 + v_0\,t + \frac{f_0}{2}\,t^2 + \frac{f_1 - f_0}{6}\,t^3, \tag{3}$$

where all quantities except for the time t are two-dimensional vectors. Given the initial velocity v_0 and position p_0, and imposing the constraint that the final velocity is zero, $v(1) = 0$, and the final position is $p(1) = p_1$, the initial and final forces, f_0 and f_1, must satisfy

$$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{6} \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \end{bmatrix} = \begin{bmatrix} -v_0 \\ p_1 - p_0 - v_0 \end{bmatrix}. \tag{4}$$

Generalizing this slightly to accommodate arbitrary start and end times, and expressing the result in terms of the changes in time, position, and velocity, Δt, Δp, and Δv,

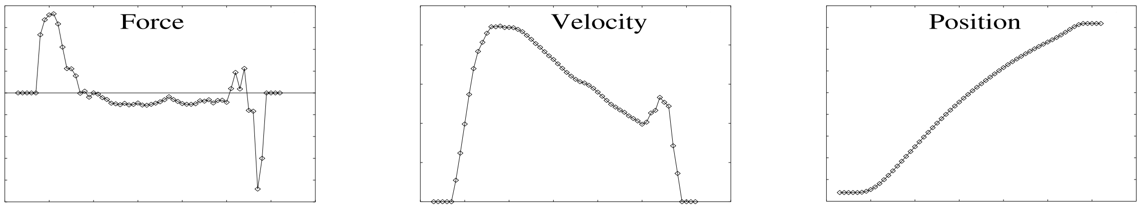

Figure 3: ***(left) The force (or acceleration) of an actual magnet as it is being moved. At each time step the goal is re-computed based on the current best guess as to the user's intent and when it will be achieved. This creates a chaotic-looking graph. (middle) The current velocity. (right) One coordinate of the resulting magnet position.***

respectively, we obtain

$$f_0 = 6\frac{\Delta p}{\Delta t^2} + 4\frac{\Delta v}{\Delta t} \tag{5}$$

$$f_1 = -6\frac{\Delta p}{\Delta t^2} - 2\frac{\Delta v}{\Delta t}. \tag{6}$$

Note that Δv is always $-v_0$, since the effect of each goal state is to leave the magnet at rest, which implies that $v_1 = 0$. Given the initial and final forces, the current force is incrementally computed, which in turn incrementally updates the velocity, and ultimately the position. This approach smoothly accommodates continually changing goals, even as multiple magnets travel on trajectories toward outdated targets, for chaotic input events and forces are smoothed by multiple integrations; that is to say, by inertia. This process is illustrated in Figure 2.

Figure 3 shows actual data associated with a magnet in our system as it responds to the magnet being dragged by the user. The graphs depict the x-component of the force, velocity, and position associated with a single magnet over time as it is being moved out of the way. The goal position and the time at which it is to be reached is re-computed each time the user's input is sampled; that is, whenever new predictions and estimates are available. Although this leads to chaotic-looking forces, as shown on the left of the figure, the velocity remains continuous and the position changes smoothly. In this example the rate at which the user drags a magnet varies wildly, which causes abrupt changes in the predicted time of collision and commensurate changes to the force function.

Another important job performed by our system is to maintain an interpretation of the current configuration and to determine the appropriate response when a magnet is picked, dragged, and dropped. When a magnet is dropped it automatically aligns with and attaches itself to other nearly-abutting magnets in its immediate proximity, which assists in the task of creating nicely-aligned rows of magnets. Dragging a magnet drags all magnets attached to it, while shaking it breaks its association with attached magnets. In general, the appropriate behavior associated with such actions will depend upon context as well as recent history. Such interpretations will depend heavily on the application; in the following section we suggest several other possible applications of the ideas we have explored in this paper.

4. DISCUSSION AND FUTURE WORK

A magnet in our system can exhibit surprisingly complex and subtle behavior due to changing predictions, feedback, and context, all of which can affect pending goals and thereby result in constantly-changing forces. We wish to apply the idea of geometric anticipation to problems such as the manipulation of text in both documents and mathematical expressions. For example, we envision an equation editor that anticipates algebraic transformations [2] such as permuting terms, factors, or subscripts, inserting expressions, and moving subexpressions. The approach explored in this paper applies nicely in the context of manipulating mathematical expressions, as the inherent two-dimensional nature of such expressions provides ample opportunities for anticipation and for fluid motion, which may include graceful changes in scale and typeface as well as spacing. Finally, we believe that more accurate and robust predictions may be obtained through the use of Kalman filtering [4] as opposed to linear least squares.

5. REFERENCES

[1] Apple Computer, Inc. Mac OS X 10.2. Operating System Software, 2002.

[2] R. Avitzur. Direct manipulation in a mathematics user interface. In *Computer-Human Interaction in Symbolic Computation*, pages 43–60. Springer-Verlag, New York, 1998.

[3] A. H. Barr. Teleological modeling. In N. Badler, B. Barsky, and D. Zeltzer, editors, *Making Them Move: Mechanics, Control, and Articulation of Articulated Figures*. Morgan Kaufmann, 1991.

[4] E. Kalman, Rudolph. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[5] G. Robertson, M. Czerwinski, K. Larson, D. C. Robbins, D. Thiel, and M. van Dantzich. Data mountain: using spatial memory for document management. In *UIST '98: Proceedings of the 11th annual ACM symposium on User interface software and technology*, pages 153–162, New York, NY, USA, 1998. ACM Press.

[6] D. Rubine. Combining gestures and direct manipulation. In *Proceedings of CHI '92*, pages 659–660, Monterey, CA, 1992.