A Bayesian Network Model of a Collaborative Interactive Tabletop Display

Mark S. Hancock
School of Computer Science
University of British Columbia
2366 Main Mall
Vancouver, BC V6T 1Z4
mhancock@cs.ubc.ca

December 9, 2003

Abstract

In this paper, I explore the use of Bayesian Networks to model the use of an interactive tabletop display in a collaborative environment. Specifically, this model is intended to extract user-profile information for each user including their location at the table as well as their handedness. The network uses input from a six-degrees-of-freedom stylus device as its source of observable information. This paper introduces a first attempt at a model to support these requirements as well as a preliminary evaluation of the model. Results show that the model is sufficiently accurate to obtain a user profile in real time in a Tabletop Display environment.

1 Introduction

Computer technology is moving off of the desktop and onto the tabletop, the walls, and into the environment. The once dominant paradigm of one user, one set of input devices, and one computer is being augmented by concurrent and distributed groups of users, multi-modal input, and loosely coupled networks of computing devices. Rather than being primarily a solitary activity, computing ought to also support highly collaborative group activity. Two challenges arise from this shift: the need to support the development of collaborative applications (how to provide support for simultaneous interaction for multiple users with diverse input and output capabilities) and the need to tailor displayed information to the specific circumstances of each user (how to format the display and provide interaction affordances for individual users).

I look at a specific aspect of these two problems, the modeling of users based on position and orientation information obtained from a 6-degree-of-freedom (6DOF) input device. This modeling technique can facilitate multi-person, multi-modal interaction by making it possible to tailor the display to the needs of each user. The focus in this paper is on supporting people who share horizontal surfaces, such as desks, workbenches, and tabletops during computer-assisted collaboration.

At an interactive tabletop display, many users can simultaneously access the same digital information at the same place and time. However, with many users sitting on different sides of the table, digital artifacts will appear differently to each user. Thus, it may be useful to automatically display objects, such as pop-up menus and dialog boxes, toward the user who invoked them. This adaptation requires that the system know the side of the table at which each user is sitting.

Furthermore, since the input device to the tabletop display is a pen, the users hand will obscure a portion of the display. It is therefore also beneficial to know the location of the users hand in relation to their pen (i.e. both their handedness *and* location). In general, to build an adaptive system of this nature, it is useful to have a profile of the current user. I propose to use a probabilistic model of the tabletop display environment to extract this information as the users interact using the table.

Determination of the side of the table at which the user is sitting has been determined probabilistically using a Feed-Forward Neural Network [2]. This approach has several drawbacks. The neural network has the disadvantage that it only can determine the side of the table and not the handedness of the user. Furthermore, it is not easy to extend the model to include other sources of information. Lastly, the model requires the use of all six degrees of freedom, and no clear understanding of how these variables interact can be determined.

1.1 System Description

The tabletop system used is top-projected and consists of a 150cm by 80cm white laminate surface onto which output from a Pentium IV 2.0 GHz computer is projected. The projected display is 90 cm by 67 cm with a resolution of 1024 by 768 (See Figure 1).

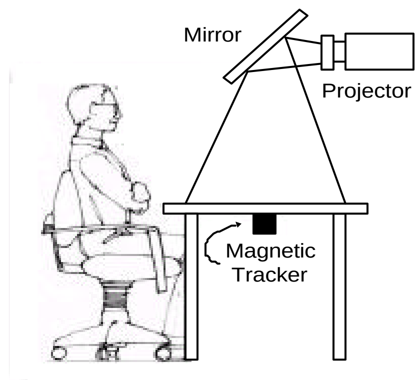

Figure 1: Top-projected Interactive Tabletop Configuration

2 Problem

Pen input is provided by a Polhemus Fastrak with styli receivers. A Fastrak is a six-degree-of-freedom magnetic tracking device that can detect the position and orientation of up to four input devices. I have developed Java-based software to process the Fastrak information sent to the serial port. To reduce magnetic interference, the table used is entirely made of wood and all magnetic objects are not placed on or near the table while in use. The tracker cube is mounted underneath the table in the center using industrial strength Velcro. By placing the cube in the center of the table, the average distance to each possible screen coordinate is minimized for improved accuracy.

2.1 Determining A User Profile

Several attempts have been made to develop a model of the user probabilistically. In [1], a user model is created for use in an intelligent interface agent called GESIA. The agent is intended to facilitate the use of an expert system. The Lumiere project [3] creates a user model using Bayesian networks to infer user needs in the domain of productivity applications. [4] utilize Bayesian Networks to determine the user's intent to aide the design of interface agents.

This work differs from previous work in that the user profile is needed *before* information is presented to the user. Thus, the application itself must be the intelligent agent and does not have the opportunity to interact with the user before making a decision about how to present information.

To determine user profile information in this environment, I propose to use a Bayesian Network [6] to model each user at the interactive tabletop display. The model will contain information about the side and handedness of the user, as well as the position and

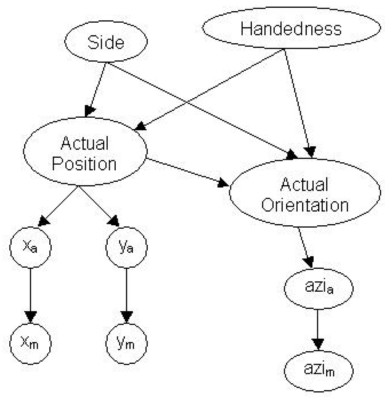

Figure 2: Bayesian Network Model of the Interactive Tabletop Display System

orientation of the input device.

3 Description of Network

The model of the tabletop display system contains ten variables, four discrete and six continuous (See Figure 2).

Using this model, the system can observe the measured coordinates from the Polhemus Fastrak device and eliminate variables to determine any of the following information about the person using the given input device:

- The side of the table at which the user is sitting
- The handedness of the user
- The true position and orientation of the user's device

3.1 Discrete Variables

Probability tables for each discrete variable were obtained by training from a dataset of 10 users. For preliminary testing, data were only collected from right-handed users. 3662 instances of data were collected in total. This data is later used to test the network (see Section 4).

Each side and each handedness value were given equal prior probabilities. Although this may not reflect a true tabletop display environment, these probabilities were used to test the ability of the network to detect the true value of these variables *without* prior knowledge. If used in practice, these prior probabilities should be adjusted accordingly.

3.2 Continuous Variables

Each of the measured input coordinates (x_m,y_m,$azimuth_m$) are (essentially) continuous and must be discretized in some way. To achieve this discretization, I chose to model each measured coordinate with a continuous Gaussian distribution representing the likelihood of error in the device. Each actual position (x_a,y_a,$azimuth_a$) coordinate was then given a uniform distribution across a 3x3 grid of the horizontal display surface. The actual azimuth angle was uniformly distributed over fifteen discrete ranges of angle. The size of the grid and the number of discrete angle ranges were varied, but these values were chosen to be sufficient to obtain accurate results.

For both the x- and y-coordinates, the same Gaussian probability distribution was used. The measured coordinates are assumed to be normally distributed with a mean at the *actual* x-coordinate with a standard deviation of 10 pixels. For the azimuth angle, the measured angle is similarly normal with a mean of the actual angle, but a standard deviation relative to the measured elevation of the stylus (See Section 3.4).

Thus, the probabilities are as follows:

$$P(x_m|x_a) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x_a - x_m)^2}{2\sigma}}$$

(y is similar)

$$P(azi_m|azi_a) = \frac{1}{\sqrt{2\pi} f(elev_m)} e^{\frac{-(azi_a - azi_m)^2}{2f(elev_m)}},$$

where

$$f(elev_m) = \frac{elev_m}{9}$$

3.3 Elimination of Variables

The continuous variables are eliminated from the network first. Observing each measured coordinate results in the functions seen above, so no work need be done. To then eliminate the remaining continuous variables (the actual coordinates), the integral of each function is taken in the appropriate interval for each quadrant of position and each class of orientation angle. The Gaussian integral is approximated using the continued fraction:

$$\int_0^a e^{-t^2}\,dt = \frac{\sqrt{\pi}}{2} - \frac{\frac{1}{2}e^{-a^2}}{a + \frac{1}{2a + \frac{2}{a + \frac{3}{\cdots}}}}$$

After eliminating the coordinate variables, the remaining variables are all discrete. Thus, variable elimination can be done normally and is executed in the following order(s): orientation, position, side, then handedness (to get handedness) and orientation, position, handedness, followed by side (to get side). The first two eliminations need only be executed once. Furthermore, if either side or handedness is known, the network's performance can be improved by observing the variable before eliminating the remaining three.

3.4 Simplifications

Because user profile information is extracted as the user interacts with the table, the system must still respond in real-time as the information is collected. Thus, the variable elimination must be done quickly so that the user does not notice any delay in the performance of the pen-input device. For this reason, a simpler model is preferred.

Firstly, the measured z-coordinate and roll angle of the stylus is ignored. These degrees-of-freedom are not likely to be useful in determining either handedness or location. Furthermore, the elevation angle of the stylus is not directly modeled by the network, but rather included by varying the standard deviation of the error in azimuth angle in relation to the measured elevation.

4 Evaluation

To evaluate the performance of the algorithm, data were collected from 10 right-handed people while interacting with the tabletop display. 3662 instances of data were collected in total. I use the 10-fold cross-validation technique to test the accuracy of the network [7, 5]. The missing data were accounted for in two different ways. Firstly, missing probabilities were uniformly replaced with a probability of 0.01. Secondly, each instance of data for right-handed users was mirrored for left-handed users before training. That is, the azimuth angle was flipped across the y-axis and the x- and y-coordinates were unchanged for each instance of data resulting in 7324 instances of data. Results for both methods are reported below.

Each test was performed using a 2x2, 3x3, and a 4x4 grid discretization. Also, several uniform discretizations of azimuth angle were considered so that the number of divisions varied from 4 to 20.

5 Results

The results show that the network is sensitive to the discretization used. Using a 2x2 grid and 4 divisions of azimuth angle, the network performs poorly. As the size of the grid and the number of divisions increases, so does the performance. Results appear to stabalize with a 3x3 grid and 15 divisions of azimuth angle. Results for this discretization are presented below.

For determination of side, the network is 98.5% accurate on average (σ=0.9%) with the first method of dealing with missing left-handed data. Using the second method, the network is 98.5% accurate on average (σ=0.4%). This result suggests that the network could be used in practice to determine user location.

For determination of handedness, the network is 99.5% accurate on average (σ=0.3%) using the first method. Note that, using this method, the determination of handedness accuracy estimate is confounded by the fact that only right-handed users were tested. Thus, the network simply has a bias for right-handed users. With this bias taken into consideration, the high accuracy rate still suggests that determination of handedness is possible using the network. On the other hand, with mirrored data, the network is accurate only 39.5% of the time on average (σ=1.0%). Despite the fact that this data is manufactured, this result is perhaps indicative a defect in the model for detection of handedness. Thus, no conclusion about handedness determination can validly be drawn with the dataset used for these tests.

Furthermore, each elimination takes on average 10.9 ms (σ=1.3 ms). This result suggests that the network could be used in real-time to determine handedness of users at a tabletop display.

Further tests should be conducted using a more complete dataset before concluding that the network is a sufficient model of the tabletop environment, but these preliminary results are promising.

6 Conclusion

Acheiving an accurate profile of users in a tabletop environment can allow the system to adapt to the user and display objects on the screen appropriately. Using the Belief Network described in this paper, a sufficiently accurate profile can be automatically extracted from the measured orientation and position of the input device. The results of the evaluation of the network show that the model can accurately predict the side at which a right-handed user is sitting. Further testing is required to determine if the accuracy level can be maintained with the addition of handedness information.

7 Future Work

Firstly, the existing data can be further analyzed to group these classes more appropriately. A probabilistic method could be used to classify the surface of the table into regions as well as to classify the ranges of azimuth angle. These empirically determined regions and ranges could lead to more accurate results. Secondly, to further improve accuracy, more observable data should be included in the model, such as information about digital artifacts currently being displayed on the table. For example, given information about what components are visible to the users, and which components are being used by a particular user, the users' locations can more accurately be determined. Also, I would like to explore the use of a history of user events to better extract profile information over time.

Furthermore, it would be beneficial to extend the model to include multiple users and the causal relationships that occur between the individuals at the display. With a collective understanding of the environment, the system could better adapt to more complicated interactions. For instance, the system could attempt to display information in a manner suitable for the collective group, or to avoid cluttering areas of the screen that may be easily obscured by others at the table.

The use of probabilistic models in collaborative environments can be useful in developing adaptive systems. Such models allow for an improved understanding of group activity. With the power of such artificial intelligence techniques, perhaps the computer system can respond appropriately to complex interactions that cannot be exhaustively accounted for by an application programmer.

Bibliography

[1] Scott M. Brown, Robert A. Harrington, Eugene Santos Jr., and Sheila B. Banks. User models, intelligent interface agents, and expert systems. In Vadim Stefanuk and Chia Laguna, editors, *UM 97 Workshop: Embedding User Models in Intelligent Applications*, June 1997.

[2] Mark S. Hancock. A feed forward neural network for determining a user's location. Technical Report TR 2001-2, Simon Fraser University, 2001.

[3] E. Horvitz, J. Breese, D. Heckerman, D. Hovel, and K. Rommelse. The lumiere project: Bayesian user modeling for inferring the goals and needs of software users. In *Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 256–265, 1998.

[4] Eugene Santos Jr., Scott M. Brown, Moises Lejter, Grace Ngai, Sheila B. Banks, and Martin R. Stytz. Dynamic user model construction with bayesian networks for intelligent information queries. In *12th International FLAIRS Conference*, pages 3–7, 1999.

[5] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International Joint Conference on Artificial Intelligence*, 1995.

[6] Judea Pearle. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kauffman, 1988.

[7] S. M. Weiss and C. Kulikowski. *Computer Systems That Learn*. Morgan Kaufmann, 1991.