

Pointing: A component of multimodal instruction

Henry Hexmoor

Department of Computer Science

University of North Dakota

Grand Forks, ND 58202

hexmoor@cs.und.edu

Jade Yang

Computer Graphics Technology

1419 Knoy Hall

Purdue University

West Lafayette, IN 47907

jyang@tech.purdue.edu

Abstract

We introduce an algorithm for visual selection of objects when guided by a physical pointer. This is part of a multimodal instruction given to a robot. Our algorithm is implemented and tested with images gathered by a camera mounted on a mobile robot.

1 Introduction

Interfaces among living creatures utilize the inherent multimodality of perception in organisms. Redundancy and cross-cueing of senses in each creature provides robustness for perception. Natural language, speech, vision, and physical gestures are becoming common modalities of interfaces between humans and robotics systems. Such an interface will be very useful in communicating with a robot like the Personal Satellite Assistant being developed at NASA Ames (<http://ic-www.arc.nasa.gov/ic/psa/overview.html>). We expect that a common scenario will involve an astronaut asking a PSA to inspect an object beyond the astronaut's reach. The astronaut might say, "inspect the lever to the right of the red panel at the end of this hose," while extending his arm, pointing to a hose, and looking in that direction. Meanwhile, the PSA would be looking in the general direction of the astronaut's arm. Its task would be to concurrently process the language and gesture input while visually picking out the hose in question. Once the PSA completes the task of understanding the command, it must figure out the navigation and selection tasks needed to carry out the instruction.

In this paper we report on extension of our methodology of interface between a robot and human operators [Hexmoor and Bandera, 1998]. Rich Human-computer interfaces are becoming popular [Oviatt, et al 2000]. Our interface allows a human commander to use natural language and indexical noun phrases combined with physical pointing to communicate locations and places to the robot. We will limit our focus to visual selection of objects guided by a pointer. First we will

present the vision algorithm needed for selecting objects pointed to by a pointer and then a few experiments that compare human vision with results of our algorithm.

2 Vision Algorithm

In our approach to gesture interpretation of pointers we have made several assumptions [Hexmoor and Valverde, 1997; Yang 1998]. We assume that our robot sees the entire pointer and the pointer is pointing roughly in the direction of the view. A pointer is an oblong object with a distinguished pointing end and a base. In other work we will consider developing a generalized pointer that may relax this assumption.

Consider the simplified 2D diagram depicting a top-down view of a pointer and several objects in Figure 1. This diagram shows about a 120-degree field of view. The pointer is approximately a foot in length. Two fields of regard are shown around the pointer. The field of regard denoted by α degree aperture is the largest range of consideration for the pointer. In our experiments we have considered α to usually be in the range 30-60 degrees. The field of regard denoted by β degrees is the range where human vision is somewhat cognitively uncertain. In our experiments, we have considered β to be roughly 0-4 degrees. This variation is due to acuity in human vision.

Six objects are shown in the scene. We will narrate our choice of the object being selected by the pointer. We are describing an implemented algorithm that has been tested on many sample images and compared with human responses [Yang, 1998]. β in Figure 1 is shown as unrealistically large. This is only for ease of illustration and it is usually much smaller. The algorithm is very sensitive to variations in α and β . The algorithm takes x,y,z coordinate values for each object in the image and x,y,z coordinate values of end points of the pointer as input and outputs an object. With known objects, it is simple to collect a table that gives correspondence with typical locations of an object in space and how the object

appears in the image. We used this for depth estimation. The table is used for interpolating object depths from image parameters.

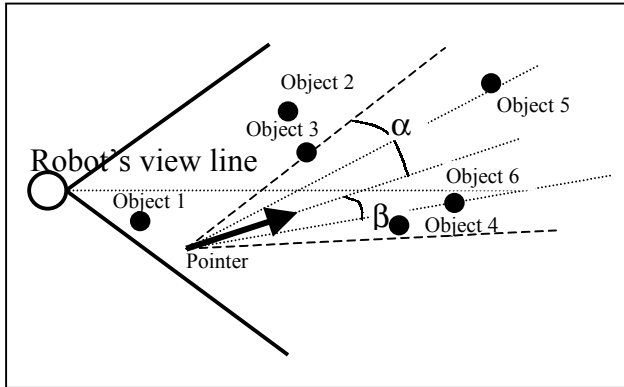


Figure 1. A simplified 2D diagram depicting top-down view of a pointer and several objects

The algorithm operates in three phases. In the first phase, either the pointer tip is clearly on the object or the pointer cannot possibly select certain objects. Objects 1 and 2 are eliminated since they are outside the pointer's fields of regard. In fact, any object that is located behind the plane at the pointer tip and perpendicular to the pointer line is not considered. An exception is when the pointer tip is literally touching or within infinitesimal proximity to the object. In that case, the object is selected and we are finished.

In the second phase, we consider the region in the α range and let's assume there is no β field of regard. Figure 1 shows objects 3, 4, 5 and 6 in this region. In this phase, we select the object that forms the smallest angle to the pointer line. If there are objects with the same angle to the pointer line, the object that has the shortest distance to pointer tip is selected. Naturally, object 6 is the most preferred, and then object 5. If object 5 and 6 did not exist, we would have to choose between objects 3 and object 4. We would choose object 4 since it has the smallest angle.

In the third phase, we consider the field of regard in the β range. We assume vision cannot select objects based solely on angles. In this range, perpendicular lines (PL) from objects to the pointer line are considered. Next, consider the following ratio we call *candidacy* for each object O . $candidacy(O) = (PL(O) - \min(PL))/PL(O)$. $PL(O)$ is the size of PL for the object O . $\min(PL)$ is the PL for the object that produces the smallest PL. For each object O , $candidacy(O)$ is compared to the ratio $(\alpha - \beta)/\alpha$ and if it is smaller or equal, we say the object is a *candidate*. After considering all objects, if there is a

single candidate, the object is selected; otherwise, the situation is ambiguous and no object is selected.

In Figure 2, objects 5 and 6 have the same angle, $PL(O6) = 4/16$, $PL(O5) = 7/16$, $\min(PL) = PL(O6) = 4/16$, $candidacy(O5) = 0.4$, and $candidacy(O6) = 0.0$. With $\alpha = 30^\circ$ and $\beta = 4^\circ$, the situation is ambiguous which is intuitively correct.

We have implemented two variants of our algorithm that work with 2D and 3D images. For 2D images there is no need for depth information. In 3D images, we recover depth information using our single camera and a calibration technique. Calibration is simple. We build a table of object parameters in images by taking picture of objects at known locations. Here are the steps for an entry in the table.

- (1) Measure the size of the real object in the scene to be analyzed, such as the diameter of the ball.
 - (2) Measure the distance from the object to the camera.
 - (3) Measure the projection of the distance from the object to the camera.
 - (4) Take a picture for this object. Then after this picture is processed with the CVIptools, measure the size of the element in the image and the distance from the element to the center of the image.
- By applying steps 1-4 to many objects at various locations, we build our calibration table. Using this table we look up the table and interpolate depth for new situations.

3 Experiments

We used the Sony camera mounted on a Nomad mobile robot. The images are all generated from a static camera position. The lighting consists of the overhead fluorescent room lights to provide enough illumination for the elements in the picture. The objects in the scene are a pointer and several similar sized balls. To simplify the process of identifying the figure from the ground, the balls are placed on a homogeneous color background.

We used CVIptools [Umbaugh, 1998] image processing for finding object locations. As the input to the software is the digitized image, the output of the CVIptools, after filtering out less important details, is a highly simplified image representation and contains only important graphics primitives of the original scene.

We will illustrate the algorithm by the following experiments (for more experiments see [Yang, 1998]). In each example we will show the raw gesturing image followed by its image representation. We then discuss consistency with human perception.



Figure 2. A 2D image

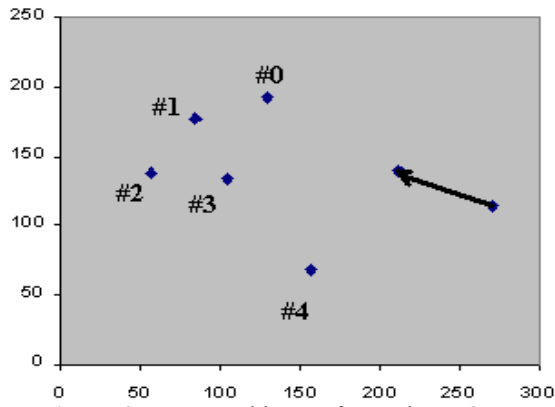


Figure 3. Processed image from Figure 2



Figure 4. a 2D image

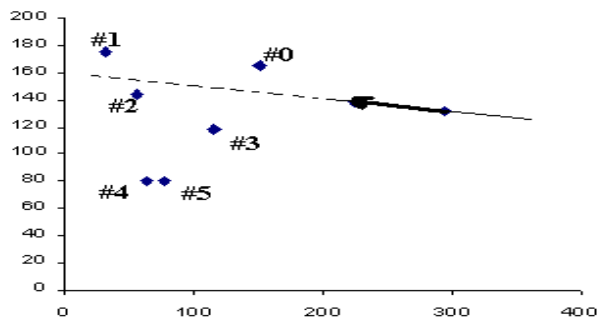


Figure 5. Processed image from Figure 4

Experiment 1: Figure 2 shows a two-dimensional image and Figure 3 shows the processed image with $\alpha = 30^\circ$, $\beta = 4^\circ$. Our algorithm shows that object #1 is being pointed at. Our 12 human subjects all agreed that object 1 is being pointed at. An interesting observation is that if we change β to any value less than α , the answer does not change.

Experiment 2. Figure 4 shows a two dimensional image and Figure 5 shows the processed image with $\alpha = 30^\circ$, $\beta = 4^\circ$. Our algorithm decides that the pointing is ambiguous. 58% of human subjects decided it is ambiguous, 25% picked object #2, and 17% picked object #1. If we decrease β to about 4.6° , the output of the algorithm is object #2. This is the second choice of our human subjects.



Figure 6. A 3D image

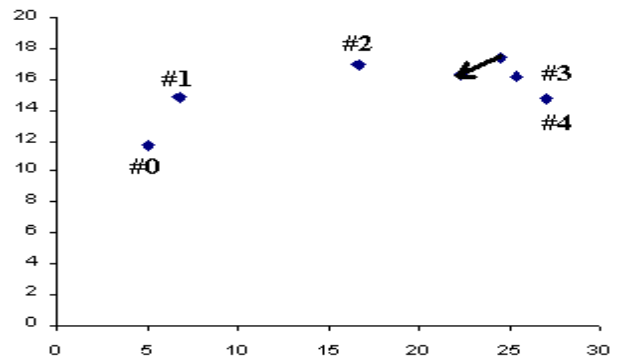


Figure 7. Processed image from Figure 6

Experiment 3. Figure 6 shows a 3D image and Figure 7 shows the processed image with $\alpha = 30^\circ$, $\beta = 4^\circ$. The output of our algorithm is object #0. This was consistent with the human subjects.

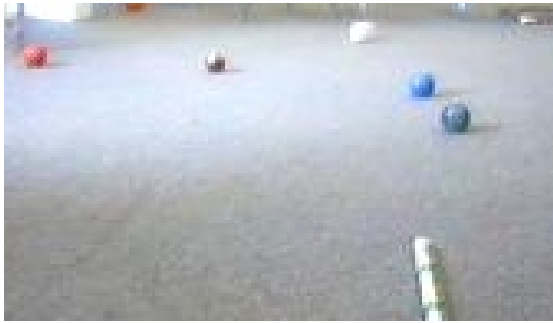


Figure 8. A 3D image

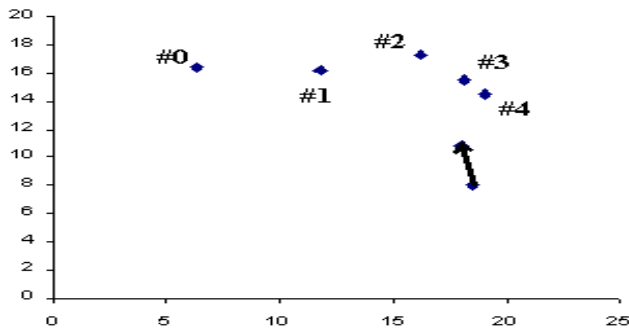


Figure 9. Processed image from Figure 8

Experiment 4. Figure 8 shows a three dimensional image and Figure 9 shows the processed image with $\alpha = 30^\circ$ and $\beta = 4^\circ$. The output of our algorithm is ambiguous. 67% of human subjects decided that the pointing is ambiguous, 16% picked object #3, and 17% picked object #2.

4 Related work

[Perzanowski, Schultz and Adams, 1998] describes control of a semi-autonomous mobile robot by giving commands through integrating spoken natural language and natural gesture. In contrast to the synthetic methods of human-computer interface such as a DataGlove, where a human has to learn a set of special language to communicate with the robot, the system they developed is a progressive step towards a user-friendlier human-computer interface in the robotics domain. A limitation of their current system is that it cannot handle very small movements for the purpose of disambiguation of a verbal command. Therefore, the gestures they use for that system are limited to dramatic movements so that the visual system can discriminate. In contrast, the system we developed is sensitive in the sense that it is designed to adjust its response upon even a slight shift of the pointing gesture since a minute change of the pointer position might change the target of pointing.

Perseus [Kahn and Swain, 1997] is the architecture for Chip's visual system developed at the University of Chicago. Perseus is a real-time system that can determine when a person enters the scene, track the relevant parts of the person including head and hands, and recognize when the person is pointing. Once the person points with his finger, the object pointed to is located. Unlike this system, instead of using a person to make the pointing, we use a pointer device. Thus, we track the tip and the end points of the pointer instead of the positions for the hand and head of the person in Perseus. Upon detecting the pointing gesture, a line from the person's head to the pointing hand is found and a cone (a field of regard) is centered on it starting at the hand and projecting away from the head. The cone is perfectly aligned with the object if the person's line of sight to the object passes through their hand. The cone is examined for each object in a list of possible objects. If none of the objects are found, the width of the cone is increase and search is repeated. Unlike Perseus, we decrease the width of the field of regard as searching proceeds instead of enlarging it.

5 Conclusions

We have implemented an algorithm that selects objects pointed to by a physical pointer. We have tested our algorithm with promising results. We plan to do many additional experiments in order to refine our algorithm. Our algorithm is very sensitive to ranges of field of regard. We want to investigate dynamic selection of these fields that mimic human vision.

An interesting area of work we are pursuing is interaction between vision and language. Use of language is a source of knowledge for visual disambiguation and use of vision is complimentary to language understanding with sentences that contain referents in the world. The latter is known as Deixis.

References

Hexmoor, H., and Bandera, C. 1998. Architectural Issues for Integration of Sensing and Acting Modalities, ISIC/CIRA/ISAS 1998, IEEE press, NIST, DC.

Hexmoor, H., Valverde, F. 1997. Toward Object Selection with a Pointer, SUNY Buffalo, TR.

Kahn, R., and Swain, M., 1996. Understanding People Pointing: The Perseus System, Technical report, Department of Computer Science, University of Chicago.

Oviatt, S.L., Cohen, P.R., Wu, L., Vergo, J., Duncan, L., Suhm, B., Bers, J., Holzman, T., Winograd, T., Landay, J., Larson, J. & Ferro, D. 2000, Designing the user interface for multimodal speech and gesture applications: State-of-the-art systems and research directions for 2000 and beyond, To be reprinted in Human-Computer Interaction in the New Millennium, (ed. by J. Carroll), Addison-Wesley, in press.

Perzanowski, D., Schultz, A., and Adams, W. 1998. Integrating Natural Language and Gesture in a Robotics Domain, In the proceedings of the IEEE International Symposium on Intelligent Control: ISIC/CIRA/ISIS Joint Conference, Sept 14-17, 1998, Gaithersburg, MD, 247-252.

Yang, J. 1998. *Machine Vision Routines for Finding Objects Guided by a Pointer*, University of North Dakota, MS thesis.

S.E. Umbaugh, 1998. Computer Vision and Image Processing: A Practical Approach Using CVIPtools, Prentice Hall PTR, Upper Saddle River, NJ.