Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society, 1994. Atlanta, Georgia, August 13-16, 876-881.

Simulated Perceptual Grouping: An Application to Human-Computer Interaction

Kristinn R. Thórisson
The Media Laboratory
Perceptual Computing Section
Massachusetts Institute of Technology
20 Ames Street E15-410 Cambridge, MA 02139
kris@media.mit.edu

Abstract

The perceptual principles that allow people to group visually similar objects into entities, or *groups*, have been called the Gestalt Laws of perception. Two well known principles of perceptual grouping are *proximity* and *similarity*: objects that lie close together are perceived to fall into groups; objects of similar shape, size or color are more likely to form groups than objects differing along these dimensions. While the primary function of these "laws" is to help us perceive the world, they also enter into our communications. People can build on assumptions about each other's perception of the world as a basis for simplifying discourse: for example, we invariably refer to collections of objects simply by gesturing in their direction and uttering "those." The current work describes an algorithm that simulates parts of the visual grouping mechanism at the object level. The system uses feature spaces and simple ranking methods to produce object groupings. Computational aspects of this system are described in detail and its uses for enhancing multi-modal interfaces are explained.

Keywords: Perceptual grouping, gestalt perception, multi-modal, simulation, human-computer interaction.

Introduction

In natural dialogue a person may point to some objects, simply refer to them as *those* and ask someone to move *them*, remove *them*, tell about *them*, etc. The addressee's understanding of this behavior requires interpretation of a multi-modal act—combining the utterance ("those") with the area addressed by the gesture (McNeill, 1992; Goodwin, 1981). Perceptual grouping allows a person furthermore to resolve a reference to multiple objects without requiring that every item referenced be enumerated. If a number of objects lie close together in the direction pointed they will naturally be considered constituting the referred group.[1] If they lie scattered but have a striking visual feature in common, they will also combine to form a group. Interestingly, the gesture's form will generally not change the listener's interpretation of it because final resolution of the reference is based on Gestalt grouping principles unrelated to the gesture.

Research in gestalt perception dates back to Wertheimer (1923). Since then, perceptual organization research in psychology has grown to include both classical experimentation and computational modeling (Feldman & Ballard, 1983; Rock, 1983; Marr, 1982; Palmer, 1981; Tversky, 1977; Rosch et al., 1976). Several computational approaches to explaining perceptual grouping phenomena have been offered (Treisman, 1990; Palmer, 1981; Tversky, 1977) and the focus has been both on finding features in the visual scene that can be used to discern objects (Treisman, 1990), and on higher-level object recognition and classification (Marr, 1982; Rosch et al., 1976). The focus in this paper, however, is vision simulation within the confines of the computer's world, where objects and their attributes are well defined.

The notion of grouping displayed items according to Gestalt principles was proposed by Dr. Richard Bolt of the MIT Media Laboratory and first explored by Chin (1987) in a preliminary fashion. This paper describes a general computational model of perceptual grouping and discusses its use in human-computer interaction. The algorithm simulates the phenomena of *proximity* and *similarity* by coding object features into multi-dimensional feature space and ranking them according to proximity in that space, producing perceptual groupings that can help resolve references in multi-modal context.

Multiple Reference in Human-Computer Interaction

In the currently popular "desktop metaphor" for computer interfaces, the principle of perceptual grouping is heavily used: functionally similar objects are grouped into common spatial regions and marked with common visual features, allowing the user to locate them easily and reference them with pointing devices such as mice or trackballs. Usually such "references" are handled literally with no interpretation on the part of the interface. For situations where an artificially imposed layout is impossible, such as in terrain maps or architectural blueprints, referring to collections of objects becomes cumbersome. In these situations it would

[1] The focus here will be on *pre-attentive visual processing* and visual features—other factors contributing to reference resolution, such as dialogue history and functional attributes of objects, will not be discussed.

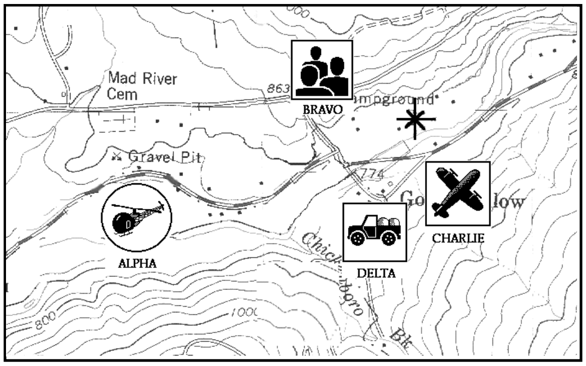

Figure 1: A typical pointing gesture may provide the computer with a single point (shown by star).

be helpful to be able to refer to objects multi-modaly, using the conventions of social communication.

Multi-Modal Interaction

Work is underway to use perceptual grouping algorithms in conjunction with multi-modal systems that recognize speech, free-form hand gestures and analyze people's gaze (Sparrell, 1993; Koons et al., 1993; Bolt & Herranz, 1992, Thórisson et al. 1992; for related work see Tyler et al., 1991; Wahlster, 1991; Hauptman, 1989; Bolt, 1987; Chin, 1987; Bolt, 1984). Bolt, as early as 1980 (Bolt, 1980), described a system called *Put That There* where multi-modal reference to single objects was possible. In the *Iconic* system (Sparrell, 1993) utterances can be mixed with free-form gestures: if a user speaks the words *"Move the chair"* while showing direction and amount of motion with a hand gesture, the system can execute the action without any further input. A recurring problem in such interaction, however, is the inability of the computer to "see" the world in the same way as people do. For example, groups and groupings of objects that are obvious to users are invisible to the machine. As a result, discourse methods for reference that build on assumptions of similarity between speaker and hearer cannot be used.

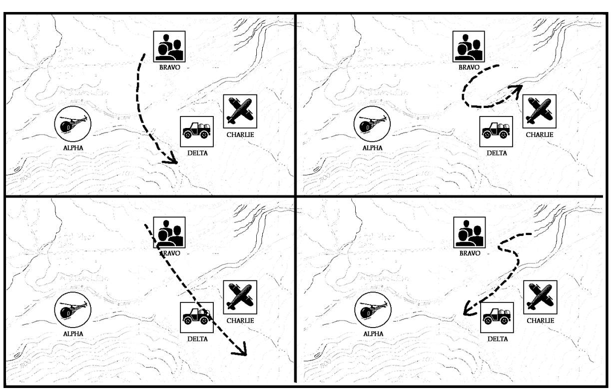

Figure 2: The user can say "Delete [gesture] these icons" and do a gesture (dotted arrows) near a group of objects. The simulated perceptual grouping algorithm enables the computer to infer which objects the gesture refers to—independent of its precise form.

A deictic gesture may in the simplest case produce a single point on the screen (Figure 1) where the person pointed (Thórisson et al., 1992). This gesture could be accompanied by speech or a key press that is predefined to mean "multiple reference." A perceptual grouping algorithm can make interpretation of the gesture both independent of the gesture's form and of the input method used: references can be made with a mouse, touch screen, data glove or even gaze (Koons & Thórisson, 1993; Thórisson et al., 1992)—these will all look equivalent to the computer (Figure 2).

The expectation we should have of any algorithm designed for this purpose is that its groupings are reasonably close to what we would expect another person to make in a conversation. It is also a necessary requirement that the algorithm takes no longer to produce an output than would be considered normal in a human interaction.

Perceptual Grouping

Real-world images gathered with cameras for computer vision systems are characterized by noise and unpredictability, with complicating factors such as zooming, panning and changes in lighting (Ballard & Brown, 1982). In contrast, computer-generated objects on a graphical display are free of these complicating issues. Since graphic objects have well-defined, accessible attributes such as size, color, etc., the approach taken here lies at the object level, taking the objects and their attributes as given.

The factors most often discussed in Gestalt perception research are (1) proximity, (2) similarity, (3) good continuation, (4) symmetry and (5) closure. Discussions of these can be found in Rock (1983) and Coren and Ward (1989); this paper focuses on the first two.

Proximity, Similarity and Perceptual Linearity

The features of proximity and similarity are best explained by example. In Figure 3 the proximity of objects results in the perceptual groupings marked *a*, *b* and *c*. Since objects can be similar in more than one way, similarity is a more complex measure than proximity. To simplify, we can say that the visual features of (1) shape, (2) size, (3) color (hue), (4) brightness (intensity), (5) orientation and (6) texture all can work toward making objects look similar. In this paper I will address the first three.

In the current approach, feature spaces are assumed to be

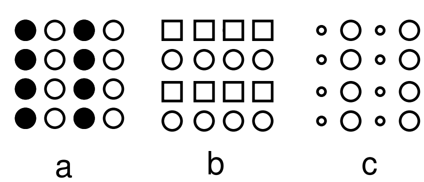

Figure 3: The perceptual Gestalt law of *similarity* comprises many features, three of which are shown here: In spite of all objects in *a*, *b* or *c* being equidistant from each other, *color* will make us see columns in *a*, *shape* will form rows in *b*, and *size* will form columns in *c*.
The law of *proximity* allows us to see *a*, *b* and *c* as three separate groups.

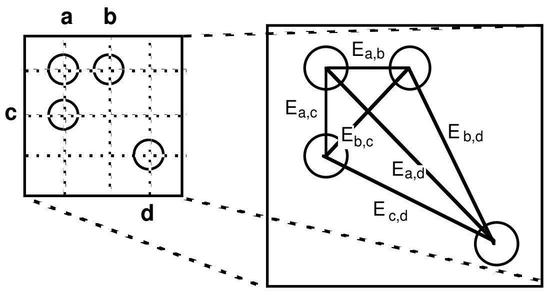

Figure 4: The objects {a, b, c, d} in the layout on the left are represented as vertices in two-dimensional space. All vertices are connected with edges ($E_{i,j}$), generating a fully connected graph.

perceptually linear. This means that the distance between any two points in that space is based on our perception of it rather than physical measurements. For spatial position there is close to one-to-one relationship between physical distance and the perceived distance (Coren & Ward, 1989). For other features, like brightness or loudness, the relationship between the physical stimulus and the perception of it is not linear; these have to be correlated in a procedure known as *magnitude estimation*. This is how the Munsell color space is constructed (Foley et al., 1990), which here provides the basis for brightness computations.

The Computational Model

The perceptual grouping algorithm works in four major steps. First (1), for all possible object pairings, it computes the objects' proximity (distance in the 2-D plane) and produces a list of edges for each pair, with an associated proximity score. This score is inversely proportional to the distance between objects. Then (2) it computes the similarity of each edge's object pair along one featural dimension (color, shape, brightness, size) present in the layout, assigns a high score if the objects are similar—a low if they are not, and weights the resulting score with that pair's proximity score. The result are edge lists, one for each feature, containing similarity scores weighted by proximity. These lists are subsequently ordered. It then (3) searches through each list looking for significant differences between adjacent edge scores. When such a difference is found, the edges so far compared are grouped. At last the algorithm (4) compares the groups produced for each feature and combines those that contain the same objects.

The approach taken here bears resemblance to some that have previously been proposed. Experimental evidence from

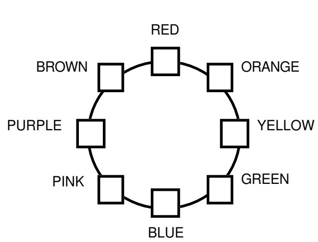

Figure 6: Color is represented as a circular, one dimensional space containing eight color values.

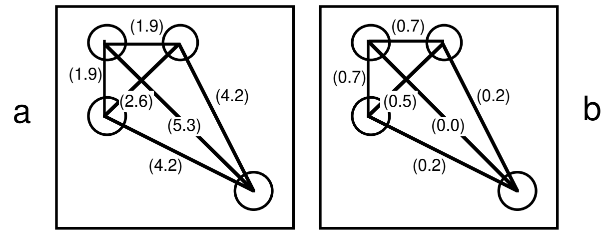

Figure 5. From the calculated distance between object pairs (a) a score is computed by normalization and inversion (b).

brain research seems to indicate the separate processing of proximity and other features (Kosslyn & Koenig, 1992; Treisman, 1990). Treisman (1982; 1980), and Palmer (1981), have proposed that in the brain, features are projected into separate spaces that can each contain one feature plus positional dimensions. This theory concurs with the algorithm described here.

Computing Proximity

The proximity score is a number between 0 to 1 that represents a linear estimation of spatial proximity of objects in the 2-D plane; high scores indicate closeness and low scores separation. To compute this, we start with a layout of objects, as the example of circles shown in Figure 4. The position of each object is viewed as a point in two-dimensional (2-D) space (x, y). This space can be drawn as a non-directed graph, $G=\{V, E\}$, where V is a set of objects and E is the set of edges, or vertices connecting them. For the layout in Figure 4 we have $V=\{a,b,c,d\}$ and $E=\{E_{a,b}, E_{a,c}, E_{a,d}, E_{b,c}, E_{b,d}, E_{c,d}\}$. Each edge receives a value depending on the proximity of the two objects it connects. First, the absolute distance between pairs of objects in 2-D space is calculated: $D_{prox(i,j)}$, where i and j are the two objects (Figure 5*a*). This distance is then normalized by the longest distance ($max[D_{prox(i,j)}]$) and at last inverted by subtracting the normalized distance from 1 (Figure 5*b*). This represents the *proximity score* (S_{prox}) for any given pair of objects in the layout and is given by: $S_{prox(e)} = 1 - (D_{prox(i,j)} / max[D_{prox(i,j)}])$, where e is an edge and i and j are the objects it connects.

Computing Similarity

The color space used here is a circular, discrete space (figure 6) where colors were chosen based on the basic color terms in language (Kay & McDaniel, 1978). A *color score* (S_{col}) is given to an edge by computing the distance between the colors ($D_{col(i,j)}$) of object pairs that the edge connects, normalizing with the longest edge and subtracting from one: $S_{col(e)} = 1 - (D_{col\ (i,j)} / max[D_{col(i,j)}])$.

Size and shape are also treated as discrete spaces. While color space is circular, however, shape (Figure 7) and size spaces are not. For simple shapes, such as the ones used

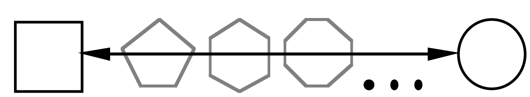

Figure 7: The square and circle can be represented as opposite ends on a shape continuum.

```
E: A list of ordered edges.
N: The number of edges in E.
F: The feature being computed.
G[COUNT,F]: 2-D array containing groups of edges.
D: Difference between adjacent scores in the list.
T1, T2: First- and second-order difference buffers.
COUNT and i are counters.
G: Number of objects in E that should be grouped.
CEILING: Variable (see text).

  D ← 0
  T1 ← 0
  T2 ← 0
  COUNT ← 0
  i ← 0

  Procedure FIND-DIFFS
        FOR i ← 1 TO (N - 1)
                D ← (Swi - Swi+1)
                IF (D > T1)
                    AND (D < CEILING)
                    AND (T2 < D - T1)
                        THEN   T2 ← (D - T1)
                               CEILING ←D
                               T1 ← D
                               G ← i

  Procedure GROUP
        G[COUNT,F] ← first G edges in E 3
        COUNT ← (COUNT + 1)
```

Procedures 1 and 2.

here, size can be represented along a single, linear dimension from small to large.

The scores in continuous spaces such as brightness are weighted by an exponential function, thus emphasizing high scores and de-emphasizing low scores: $S_{bright(e)} = 1 - \mathrm{sqrt}(D_{bright(i,j)}^2 / \max[D_{bright(i,j)}]^2)$.

Finding Gestalts

Once a feature score has been computed for an edge, $S_{(f)}$, it is weighted by the 2-D spatial proximity score (S_{prox}) of that edge: $S_{w(f,e)} = S_{(f,e)}\ S_{prox(e)}$, where f is the feature (color, shape, size, brightness) and e is the edge. The weighted scores of every feature are listed in descending order such that for each feature we have a list $L_{(f)}$: $L_{(f)} = \{S_{w(f,e')} >= S_{w(f,e'')} >= S_{w(f,e''')} >= \dots \}$. A search procedure compares adjacent pairs of edges in these lists, starting from the greatest value, looking for the largest difference on each pass (Procedures 1 and 2). A second-order difference (the difference between these differences) is used to determine when the preceding edges should be considered as constituting a significant group. Because the edges are ordered we are ensured that the *most perceptually significant* groups are *found first*.

The extent of the search in each feature list is determined by a constant (STOP, Procedure 3). It's value will depend on the range of values used in the feature spaces and how long we want to continue partitioning the layout into subgroups. A variable (CEILING) keeps track of the highest difference

```
CEILING is initialized to a value higher than the
largest possible difference between adjacent edge
pairs.
STOP is a constant.

  Procedure MAIN
        WHILE (CEILING > STOP)
              T1 ← 0
              T2 ← 0
              FIND-DIFFS
              GROUP first G edges in E
```

Procedure 3.

found so far and limits the search to a value below this every time a new search is done on the same list.

Forming Groups

Once this algorithm has been run on each feature list, $L_{(f)}$, it has produced one or more lists of edges for each feature that constitutes a significant "perceptual group." A comparison is then done among all the resulting groups. If a group occurs more than once (for example, if there is a group of objects in the scene where all objects are circles *and* of the same color), the two groups are merged. An important point is that when applying the above algorithm, only the features that vary among the objects should be used in the computations. This means that if, for instance, all objects are of the *same size*, size scores will *not* be used in the computations. To make the groupings useful in human-computer dialogue it is necessary to add a *goodness score* that indicates how well each group stands out from the rest.[2] Such a score allows the algorithm to rank groups according to how "perceptually good" they seem, and subsequently make hypotheses about a user's multi-modal references based on context and dialogue history.

Results

The above algorithm produces surprisingly good results. Two typical examples for demonstrating Gestalt principles are shown in Figures 8 and 9 (groups are ordered according to goodness in descending order from left to right, starting at the top). The third example (Figure 10) shows how the algorithm avoids improper assumptions about group adherence in semi-structured layouts.

In human-computer interaction, this level of performance is sufficient to simulate multiple-object reference resolution quite similar to that observed in human interaction. The computer can look at the ranked groups and take the best fit to the current point on the screen. If there is doubt about

---

[2] This score is currently computed by averaging the values of all links ($Sw_{(f)}$) involved in creating a group.

[3] Once the edges have been stored in G[COUNT,F], an additional search is required to see if the number of edges is equal to or larger than the number of objects they connect minus one ($N_e \geq N_o - 1$). If not, the graph has isolated nodes and two or more perceptually significant groups of objects will be contained in a single cell of G[COUNT,F].

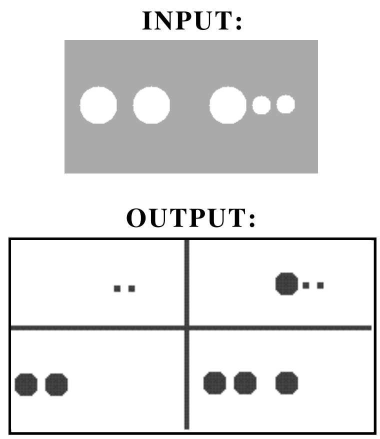

Figure 8: In this classical example demonstrating the principles of proximity and similarity of size, the system produces the same groupings that people are prone to make.

which group is being referred to, it can go down the group list to find the next-best fit. A system could also highlight the candidate groups and ask the user (with synthesized speech or printed text) which one it is. This way references can be resolved without requiring users to repeat a command.

Limitations

Even given the very restricted nature of the input that this algorithm was designed to handle, it still has a number of limitations. One of these is using a fully connected graph to compute groupings, generating an exponential growth of edges with a linear increase of objects. The problem can be dealt with in part by using only edges that fall below a certain percentage of the longest one. Preliminary tests have indicated that performance significantly deteriorates if the longest edge used in the computations is less than 60% of the largest distance between objects. Thus, up to 40% of the edges can seemingly be discarded without a serious effect on the results.

The algorithm can take into account proximity and

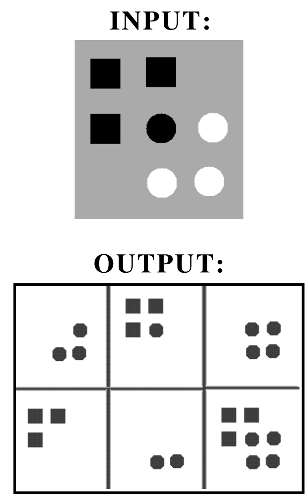

Figure 9: Example demonstrating the interaction of features competing for group adherence. (After finding the four most obvious groups, the system continues the search and comes up with one subgroup before grouping all objects together. Notice that the two white circles at the bottom are a tad closer together than the other objects.)

similarity of objects—two of the five well-known grouping principles of Gestalt perception. An important third candidate would be *good continuation*, which would make recognition of lines formed by rows of objects more robust than it is now. Whether this, and other principles can be incorporated remains to be seen.

Although the approach currently applies only to two-dimensional layouts, it may very well extend to three-dimensional spaces. An important question then is how well the computer's simulated perception of grouping will match the viewer's, who is provided with only a projection of the scene, leaving out some of the strongest cues for depth such as stereopsis and motion parallax.

Another limitation in the current version of the algorithm is the simplified treatment of shape space. With complex

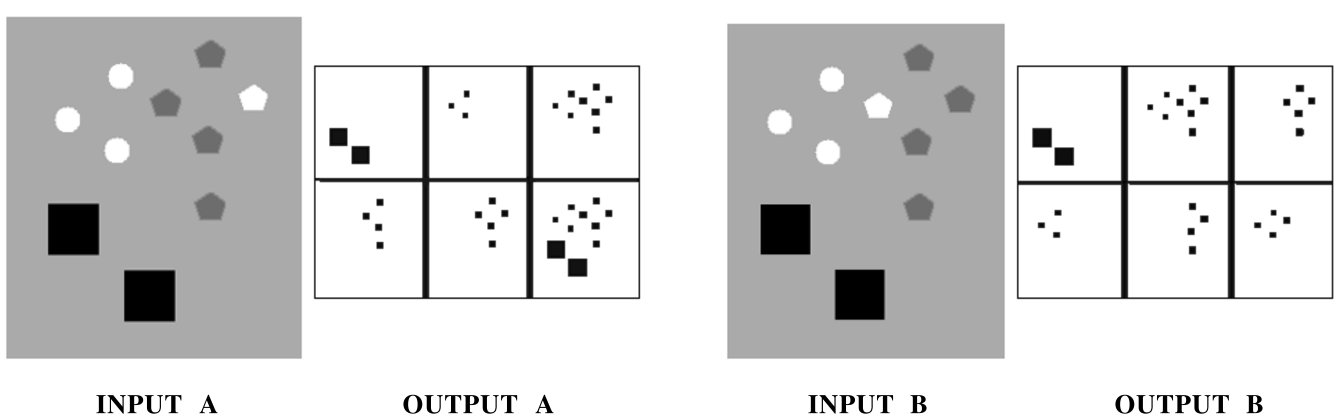

Figure 10: With unstructured input—objects varying along the dimensions of size, shape and brightness—the system's ability to compute realistic groupings becomes readily apparent. In A the white objects are not all considered constituting a "white" group because the gray pentagons intersect them. When the white pentagon is moved closer to the white circles (input B), the system will consider these a perceptually significant group, even though their shapes differ.

shapes, many issues, such as orientation and viewpoint, will start to play a significant role. It is not clear if these problems can be handled in a simple way.

Conclusion

The current work is an attempt to bring human-machine interaction closer to human-human communication by drawing on research in discourse and computer vision. The simplicity and relative computational inexpensiveness of the algorithm described allows it to produce reasonable perceptual groupings in real-time during interaction. It highlights the point that in spite of limited computer intelligence, human-computer interaction need not be limited to direct manipulation or arbitrary communication languages.

Acknowledgments

The author would like to thank Dr. Richard Bolt for suggesting this line of research and Jerome Lui, Tony Ezzat and Walter Lee for programming support. Also thanks to Edward Herranz, David Koons and Carlton Sparrell for useful suggestions. Richard Bolt, Joshua Bers, Jennifer Jacoby and Alan Wexelblatt provided valuable comments on drafts of the paper. This work was supported by the Advanced Research Projects Agency under Rome Laboratories, contract F30602-89-C-0022.

References

Ballard, D. H. & Brown, C. M. (1982). *Computer Vision*. Englewood Cliffs, NJ: Prentice-Hall.

Bolt, R. A. (1980). "Put-That-There": Voice and Gesture at the Graphics Interface. *Computer Graphics*, *14(3)*, 262-70.

Bolt, R. A. (1984). *The Human Interface*. New York: Van Nostrand Reinhold.

Bolt, R. A. (1987). The Integrated Multi-Modal Interface. Invited paper for *The Transactions of the Institute of Electronics, Information and Communication Engineers*, (Japan), November, Vol J70-D (11), 2017-2025.

Bolt, R. A. & Herranz, E. Two-Handed Gesture in Multi-Modal Natural Dialog. Proceedings of *UIST '92, Fifth Annual Symposium on User Interface Software and Technology*, Monterey, CA, November 15-18. New York: Academic Press.

Chin, A. L. (1987). An Eyetracker-and-speech Interface to Object-Oriented Computer Graphics. Senior Thesis. Cambridge, MA: Massachusetts Institute of Technology.

Coren, S. & Ward, L. M. (1989). *Sensation & Perception*. San Diego: H. B. J.

Feldman, J. A. & Ballard, D. H. (1983). Computing with Connections. In J. Beck, B. Hope & A. Rosenfeld (eds.), *Human and Machine Vision*, 107-155. Orlando, FL: Academic Press.

Foley, J. D., van Dam, A., Feiner, S. K. & Hughes, J. F. (1990). *Computer Graphics: Principles and Practice*. Reading, MA: Addison-Wesley Publishing Company.

Goodwin, C. (1981). *Conversational Organization: Interaction Between Speakers and Hearers*. New York: Academic Press.

Hauptman, A. G. (1989). Speech and Gestures for Graphic Image Manipulation. *SIGCHI Proceedings '89*, 241-245. New York: ACM Press.

Kay, P. & McDaniel, C. K. (1978). The Linguistic Significance of the Meanings of Basic Color Terms. *Language*, 54, 610-46.

Koons, D. B. & Thórisson, K. R. (1993). Estimating Direction of Gaze in Multi-Modal Context. Presented at *3CYBERCONF—The Third International Conference on Cyberspace*, April 14-15, Austin, Texas.

Koons, D. B., Sparrell, C. J. & Thórisson, K. R. (1993). Integrating Simultaneous Input from Speech, Gaze and Hand Gestures. In M. Maybury (ed.), *Intelligent Multi-Media Interfaces*, 252-276. Cambridge, MA: AAAI Press/MIT Press.

Kosslyn, S. M. & Koenig, O. (1992). *Wet Mind: The New Cognitive Neuroscience*. New York: The Free Press.

Marr, D. (1982). *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. New York, NY: W. H. Freeman and Co.

McNeill, D. (1992). *Hand and Mind: What Gestures Reveal about Thought*. Chicago: University of Chicago Press.

Palmer, S. E. (1981). The Psychology of Perceptual Organization: A Transformational Approach. In J. Beck, B. Hope & A. Rosenfeld (eds.), *Human and Machine Vision*, 269-339. Orlando, FL: Academic Press, Inc.

Rock, I. (1983). *The Logic of Perception*. Cambridge, MA: MIT Press.

Rosch, E., Mervis, C. B., Gray, W. D., Johnson, D. M. & Boyes-Braem, P. (1976). Basic Objects in Natural Categories. *Cognitive Psychology, 8*, 382-439.

Sparrell, C. J. (1993). Coverbal Iconic Gesture in Human-Computer Interaction. M.S. Thesis. Cambridge, MA: Massachusetts Institute of Technology.

Thórisson, K. R., Koons, D. B. & Bolt, R. A. (1992). Multi-Modal Natural Dialogue. *SIGCHI Proceedings '92*, April, 653-4. New York: ACM Press.

Treisman, A. (1990). Features and Objects in Visual Processing. In I. Rock (ed.), *The Perceptual World: Readings from Scientific American*. New York: W. H. Freeman and Co.

Treisman, A. (1982). Perceptual grouping and attention in visual search for features of objects. *Journal of Experimental Psychology: Human Perception and Performance, 8*, 192-214.

Treisman, A. & Gelade, G. (1980). A Feature Integration Theory of Attention. *Cognitive Psychology*, *12*, 97-136.

Tversky, A. (1977). Features of Similarity. *Psychological Review, 84*(4), 327-352.

Tyler, S. W., Schlossberg, J. L., & Cook, L. K. (1991). CHORIS: An Intelligent Interface Architecture for Multimodal Interaction. *AAAI Workshop Notes*, 99-106.

Wahlster, W. (1991). User and Discourse Models for Multimodal Communication. In J. W. Sullivan & S. W. Tyler (eds.), *Intelligent User Interfaces*, 45-67. New York: ACM Press, Addison-Wesley Publishing Co.

Wertheimer, M. (1923). Untersuchungen zur Lehre von der Gestalt. *Psychologische Forschung, 4*, 301-50. Translation in W. D. Ellis (ed.), *A Source Book of Gestalt Psychology*. New York: H. B. J., 1938.