A Probabilistic Approach to Reference Resolution in Multimodal User Interfaces

Joyce Y. Chai
Department of Computer Science
and Engineering
Michigan State University
East Lansing, MI 48864
E-mail: jchai@cse.msu.edu

Pengyu Hong
Department of Statistics
Science Center 601
Harvard University
Cambridge, MA 02138
E-mail: hong@stat.harvard.edu

Michelle X. Zhou
IBM T. J. Watson Research
Center
Hawthorne, NY 10532
E-mail: mzhou@us.ibm.com

ABSTRACT

Multimodal user interfaces allow users to interact with computers through multiple modalities, such as speech, gesture, and gaze. To be effective, multimodal user interfaces must correctly identify all objects which users refer to in their inputs. To systematically resolve different types of references, we have developed a probabilistic approach that uses a graph-matching algorithm. Our approach identifies the most probable referents by optimizing the satisfaction of semantic, temporal, and contextual constraints simultaneously. Our preliminary user study results indicate that our approach can successfully resolve a wide variety of referring expressions, ranging from simple to complex and from precise to ambiguous ones.

Categories & Subject Descriptors: H.5.2 (User Interfaces): Theory and method, Natural language

General Terms: Algorithms

Keywords: Multimodal user interfaces, reference resolution, graph matching.

1. INTRODUCTION

Multimodal user interfaces allow users to interact with computers through multiple modalities such as speech, gesture, and gaze. Since the first appearance of the "Put-That-There" system [1], a number of multimodal systems have been built, among which there are systems that combine speech, pointing [14, 21], and gaze [13], systems that integrate speech with pen inputs (e.g., drawn graphics) [4, 23], systems that combine multimodal inputs and outputs [2], systems in mobile environments [19], and systems that engage users in an intelligent conversation [7, 20]. Studies have shown that multimodal interfaces enable users to interact with computers naturally and effectively [16, 18]. Inspired by the earlier work, we are building an infrastructure called Responsive Information Architect (RIA) to facilitate a multimodal human-computer conversation [3]. Users can interact with RIA through multiple modalities, such as speech and gesture. RIA is able to understand user inputs and automatically generate multimedia responses via speech and graphics [25]. Currently, RIA is embodied in a testbed, called Real Hunter^{TM}, a real-estate application for helping users to find residential properties.

A key element in understanding user multimodal inputs is known as *reference resolution,* which is a process that finds the most proper referents to referring expressions. Here a *referring expression* is a phrase that is given by a user in her inputs (most likely in speech inputs) to refer to a specific entity or entities. A *referent* is an entity (e.g., a specific object) to which the user refers. Suppose that a user points to "House 6" on the screen and says "how much is this one". In this case, reference resolution is to infer that the *referent* "House 6" should be assigned to the *referring expression* "this one". As described later, in multimodal conversation systems that support rich user interaction as RIA does (e.g., providing both verbal and visual responses), users may refer to their interested objects in many different ways. Therefore, developing a systematic approach to reference resolution in such an environment is challenging.

To systematically resolve different types of references, we have developed a probabilistic approach to reference resolution using a graph-matching algorithm. Our approach finds the most probable referents for referring expressions by optimizing the satisfaction of a number of constraints, including semantic, temporal and contextual constraints. In this paper, we first describe the challenges associated with reference resolution and some of the related work. We then present our probabilistic approach to reference resolution. Finally, we discuss our evaluation results.

2. REFERENCE RESOLUTION

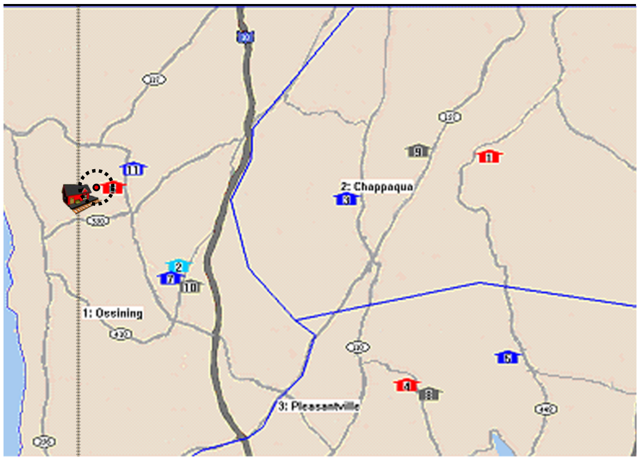
In a multimodal conversation, rich interaction channels allow users to refer to their interested objects in various ways. User references can be simple, precise, complex, or

IUI'04, January 13–16, 2004, Madeira, Funchal, Portugal.
Copyright 2004 ACM 1-58113-815-6/04/0001…$5.00

ambiguous. Next we use a concrete example (Table 1) to illustrate several common referring patterns. In this example, Real Hunter^{TM} shows a collection of objects (e.g., houses, a train station, etc.) on the map of three towns Chappaqua, Ossining, and Pleasantville.

First in U1, the user's gesture input results in a position near a house icon and a train station icon[1]. From the gesture alone, the system does not know whether the user points to the house, the train station, or the town of Ossining[2]. Continuing in U2, the user's utterance does not use any referring expression[3]. Using this input alone, it is hard for the system to identify which object the user is talking about. In U3, there are two speech referring expressions ("this" and "these two houses") and two gestures (pointing and circling). If only using the temporal ordering information, the system may not be able to decide between two cases: 1) aligning the pointing gesture with "this" and the circling gesture with "these houses"; or 2) aligning both gestures with "these two houses".

Resolving the references illustrated in the above example requires combining information from the multimodal inputs and from the interaction contexts, such as the conversation history, the system visual feedback, and the domain knowledge. For example, using domain knowledge, our system is able to infer that in U1 the user is actually referring to the house, since both the train station and the town of Ossining do not have the *price* attribute. Although no explicit referring expression is given in the speech utterance in U2, combining the conversation context and the visual properties, our system can infer that the user is referring to the size of the highlighted house on the screen.



The screen shows a collection of objects (e.g., houses, train station, etc.) on the map of three towns: Chappaqua, Ossining, and Pleasantville.

U1:	***Speech***: How much is this? ***Gesture***: Point to a position between a house icon and a train station icon at Ossining (see map).
R1:	***Speech***: This house costs 250,000 dollars. ***Graphics***: Highlight the targeted house.
U2:	***Speech***: How large?
R2:	***Speech***: It has 2200 square feet. ***Graphics***: The previous house stays highlighted
U3:	***Speech***: Compare this with these two houses. ***Gesture***: Point, and Circle
R3:	***Speech***: Here is the comparison chart. ***Graphics***: Highlight three houses and show a chart

Table 1. A conversation fragment between a user and Real Hunter^{TM}. U1, U2, and U3 are the user inputs. R1, R2, and R3 are the responses from Real Hunter^{TM}.

Reference resolution becomes more complicated when processing inputs that involve multiple referring expressions and multiple gestures (e.g., U3). Given a referring expression, it may be accompanied by a sequence of different gestures (e.g., pointing and circling in Figure 1a-b), or by a sequence of similar gestures (e.g., pointing in Figure 1c). Figure 1 shows three possible variations of gesture inputs accompanying the speech input in U3. Depending on the gesture recognition results, there may be different alignments between the gestures and the referring expressions in each case. In Figure 1(a), using the temporal information, one possible alignment is to pair the pointing gesture with "this" and the circling gesture with "these two houses". However, this alignment may be incorrect, depending on the number of houses selected by the circling gesture. If the user circles two houses, it is most likely that "this" refers to the house selected by the pointing gesture and "these two houses" refers to the two houses selected by the circling gesture. On the other hand, if the circling gesture only results in one object, the system may infer that "this" most likely refers to the highlighted house on the screen (which is also the focus of attention from the previous interaction), and "these two houses" refers to the two houses selected by

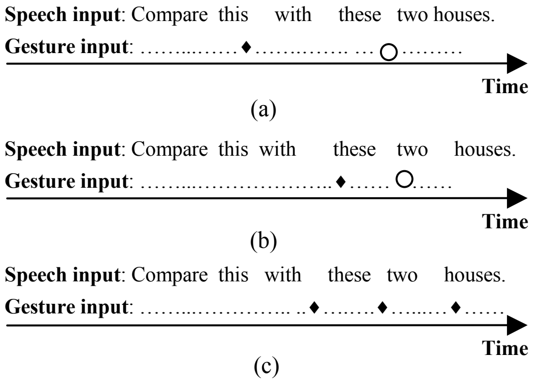

Speech input: Compare this with these two houses.
Gesture input: ……...……♦……..……. … ○………
Time
(a)
Speech input: Compare this with these two houses.
Gesture input: ……...………………..♦…… ○……
Time
(b)
Speech input: Compare this with these two houses.
Gesture input: ……...………….. ..♦……..♦…...…♦……
Time
(c)

Figure 1. Three variations of U3. The timing of the point gesture is denoted by♦ and the timing of the circle gesture is denoted by O.

[1] User gesture inputs are often imprecise especially when a touch screen is used, where a human finger is normally larger than the objects shown on the screen.

[2] The house icon and the train station icon are on top of a graphic entity, in this case a region representing the town of Ossining.

[3] More precisely, the user did not use any referring expression here (i.e., similar to zero anaphora, but "is it" is what is missing).

the pointing and the circling gestures, respectively.

Similarly, in Figure 1(b), since the pointing and the circling gestures are temporally closer to the expression "these two houses", it is more likely for both gestures to be aligned with "these two houses". In Figure 1(c), three consecutive pointing gestures are used. Although all three gestures are temporally closer to the expression "these two houses", this expression specifies that only "two" objects be the potential referents. Thus it is more likely that the house selected by the first pointing provides the referent to "this", and the following two pointing gestures supply the referents to "these two houses". To resolve all these complex references described above, we need to consider the temporal relations between the referring expressions and the gestures, the semantic constraints specified by the referring expressions, and the contextual constraints from the prior conversation. Any subtle variations in any of the constraints, including the temporal ordering, the semantic compatibility, and the gesture recognition results will lead to different interpretations.

3. RELATED WORK

Previous work on multimodal reference resolution includes the use of a focus space model [15], the use of the centering framework [24], and the use of contextual factors [8]. Approaches to multimodal integration [10, 11], although focusing on a different problem, provide effective solutions to reference resolution by combining inputs together from different modalities. For example, the unification-based multimodal fusion approach can identify referents to referring expressions by unifying feature structures generated from the speech utterances and from the gestures through a multimodal grammar [10]. The unification mechanism enforces the semantic compatibility between different inputs. In addition, it also applies temporal constraints, which are specified by a set of pre-defined integration rules [17]. Using this approach to accommodate various situations such as those described in Figure 1 will require adding different rules to cope with each situation. If a specific user referring behavior did not exactly match any existing integration rules (e.g., temporal relations), the unification would fail and therefore references would not be resolved.

A recent study by Kehler reported that the interpretation of referring expressions can be achieved with a very high accuracy using a visual context (e.g., visual focus) with a simple set of rules [12]. In our experiments, we found that those rules are very effective in processing inputs with a single referring expression accompanied by precise gestures. Since those rules assume that all objects can be deterministically selected by gestures, this approach does not support ambiguous gestures as described in our case.

Our approach is inspired by both unification-based and context-based approaches to multimodal interpretation. On the one hand, as in unification-based multimodal integration [10], our approach considers both semantic and temporal constraints. On the other hand, our approach considers the constraints from the conversation context as in a context-based approach. In particular, our approach focuses on a new aspect of reference resolution, which involves finding the most probable referents to all unknown references using information from multiple sources. Unlike earlier work, our approach always provides a solution that maximizes the overall satisfaction of semantic, temporal, and contextual constraints. To achieve this goal, we developed a probabilistic approach using a graph-matching algorithm. Specifically, we represent all information gathered from multiple input modalities and the contexts as attributed relational graphs (ARGs) [22], and model reference resolution as a constrained probabilistic graph-matching problem.

4. GRAPH-BASED INFORMATION REPRESENTATION

Reference resolution relies on not only the properties of referring expressions and potential referents, but also on their inter-relations. ARG is an ideal representation for such properties and relations. In particular, we use three ARGs to represent the information collected from the speech input, the gesture input, and the conversation context. Next we introduce our ARG representation, we then describe how to automatically generate the three ARGs.

An ARG consists of a set of nodes that are connected by a set of edges. Each node represents an entity, which in our case is either a referring expression to be resolved or a potential referent. Each node is associated with a feature vector encoding the properties of the corresponding entity. Each edge represents a set of relations between two entities, and is also associated with a feature vector encoding the properties of such relations.

Currently, the feature vector of a node contains the semantic and temporal information. The feature vector of an edge describes the temporal relation and the semantic type relation between a pair of entities. In the future, more relations (e.g., spatial relation)) can be easily added. A temporal relation indicates the temporal order between two related entities during an interaction, which may be one of the following:

- *Preceding*: Node A precedes Node B if the entity represented by Node A is mentioned right before the entity represented by Node B in a specific modality. For example, "this" precedes "these two houses" in the speech input of U3 in Table 1.
- *Concurrent*: Node A is concurrent with Node B if the entities represented by them are referred or mentioned simultaneously in a specific modality. For example, a circling gesture may select a group of objects. All selected objects are considered concurrent with each other.
- *Non-concurrent*: Node A is non-concurrent with Node B if their corresponding objects/references cannot be referred/mentioned simultaneously and the preceding relation does *not* hold between them.

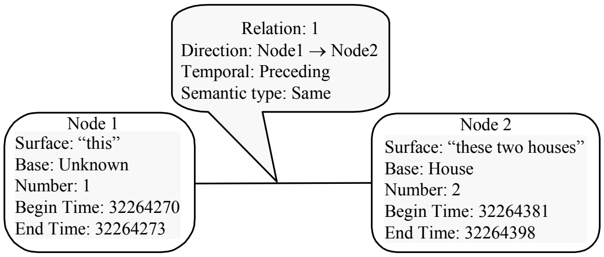


Figure 2. The speech ARG of U3

- *Unknown*: The temporal order between two entities is unknown, it may take the value of any of the above.

A semantic type relation indicates whether two related entities share the same semantic type. When a semantic type relation cannot be identified from the entities themselves or from the contexts those entities are referred to, it is set to "*unknown*".

4.1 Construction of Speech ARG

A *speech ARG* captures the information about referring expressions that occur in a speech input. Figure 2 shows the speech ARG created by processing the speech input in U3. In a speech ARG, each node represents a referring expression and each edge represents a semantic relation and a temporal relation between the referring expressions. To create this graph, our system first uses IBM ViaVoice^{TM} to transcribe a speech utterance into text, and then applies a grammar-based parser to identify key phrases and their relations.

For example, from the speech input in U3, our system identifies three key phrases "compare", "this", and "these two houses". Among these three key phrases, two referring expressions "this" and "these two houses" are identified. Accordingly, two nodes are created for these two referring expressions as shown in Figure 2. From each referring expression, our system then identifies a set of pertinent semantic features, such as the semantic type of the potential referents and the number of potential referents[4]. Furthermore, we use ViaVoice^{TM} to extract the time stamps of each word when it is uttered, and derive the time duration for each referring expression.

Our system extracts the following semantic features and the temporal features of each expression:

- The identifier of the referent. The unique identity of the potential referent. For example, the proper noun "Ossining" specifies the town of Ossining.
- The semantic type of the potential referents indicated by the expression. For example, the semantic type of the referring expression "this house" is *house*.
- The number of potential referents. For example, a singular noun phrase refers to one object. A plural noun phrase refers to multiple objects. A phrase like "three houses" provides the exact number of referents (i.e., 3).
- Type dependent features. Any features, such as size and price, are extracted from the referring expression.
- The time stamp that indicates when a referring expression is uttered.
- The syntactic categories of the referring expressions (e.g., a demonstrative vs. a pronoun). This information helps the system to draw correlations between the type of references used and the status of objects referred.

As shown in Figure 2, node 2 (corresponding to "these two houses") specifies that the potential referent should be a house object (Base: house), and the number of potential referents should be two (Number: 2).

Each edge in a speech ARG captures the temporal relation and the semantic type relation between two referring expressions. The temporal relation is decided based on the time when the two expressions are uttered. The semantic type relation is either directly derived from the expressions themselves, or inferred from the entire speech input where these expressions are uttered. For example, in U3 (Table 1), the user does not provide a specific semantic type for the potential referent in the expression "this". The whole speech utterance shows that the user is asking for a comparison of multiple objects. Based on the domain knowledge that the comparable objects most likely share the same semantic type, our system is able to infer that the semantic type relation between "this" and "these two houses" be the *same*.

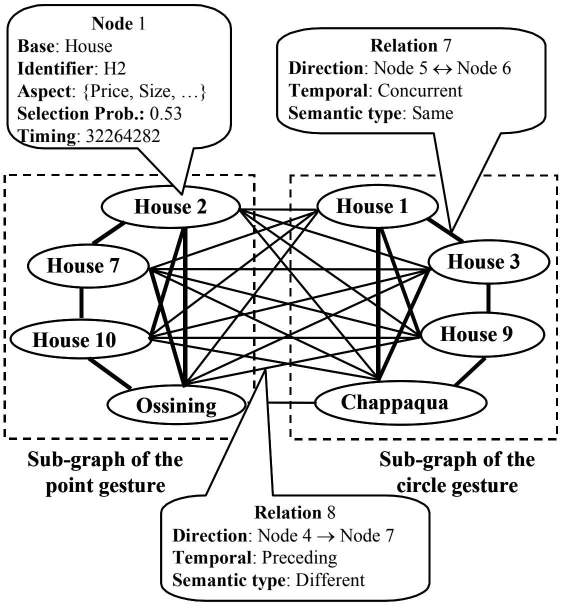


Figure 3. The gesture ARG of U3

[4] We use a set of rules to extract semantic features. For example, one rule "Num(X) Noun(Y) -> BASE: Y, NUMBER: X" indicates that if a number is followed by a noun, then this number and the base form of the noun will be extracted as the number feature (NUMBER) and the semantic type feature (BASE).

4.2 Construction of Gesture ARG

A gesture ARG encodes the information about the objects selected by gesture inputs. Currently, we focus on two types of deictic gestures, pointing and circling, both of which select objects from a graphic display. Gesture inputs may be inaccurate especially when a touch screen is used, where displayed objects are often too small for a human finger to pinpoint at. Therefore, our gesture recognition module assigns a probability to each object that is likely to be selected by a gesture.

For pointing gestures, given a selection point (x, y) on the screen, our system uses the following method to select a set of potential objects and calculate their selection probabilities. If (x, y) intersects with any polygons that constituent an object on the screen, then the selection probability of this object is 1.0. Note that the point (x, y) may intersect with multiple objects simultaneously, especially when objects overlap with each other (e.g., two nearby house icons may overlap). Otherwise, we set a circular effective region, which centers at (x, y) and has the radius r. The radius r is equal to the minimum dimension of the bounding rectangles of the objects on the screen. If the object intersects the circle, the selection probability of the object is $e^{-2d/r}$, where d is the minimal distance between the boundary points of the object and (x, y).

To recognize a circling gesture that selects one object or a group of objects, our system traces the trajectory of the user's mouse movements during interaction. We consider a trajectory a circle if it intersects with itself or its two end points are close enough. If an object icon falls inside the circle, its selection probability is 1.0; the probability is 0.0 if the object falls outside of the circle. If an object icon intersects with the circle, its selection probability is equal to the ratio of its area inside the circle to its whole area.

Using the results produced by the gesture recognition module, our system builds a gesture ARG including all gestures that occur during one interaction[5]. At each interaction, one input may contain a sequence of gestures: g_1, g_2, …, g_K. For example, the gesture input of U3 in Table 1 consists of a pointing gesture and a circling gesture. For each gesture g_i, our system creates a sub-graph. In each sub-graph, each node, known as a *gesture node*, represents an object selected by this gesture. The feature vector of a gesture node contains the following information pertinent to the object represented: the identifier, the semantic type, the attributes (e.g., a house object has attributes of price, size, etc.), the time stamp when the object is selected (relative to the system start time), and its selection probability. Each edge in a sub-graph represents the semantic type relation and the temporal relation between two nodes. The semantic type relations can be easily identified because the identities of the objects are known. The temporal relations between two nodes within a sub-graph are labeled as "*concurrent*", since within a single gesture all objects are considered being selected simultaneously. As shown in Figure 3, there are two sub-graphs generated for the pointing gesture and the circling gesture, respectively. We use the thicker edges to represent the relations between the nodes within the sub-graphs.

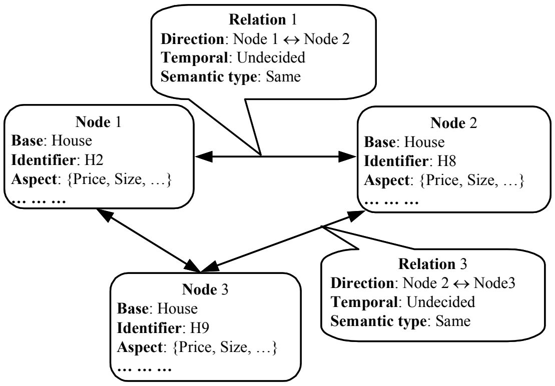


Figure 4. The history ARG after U3 is processed

By connecting sub-graphs together, our system then creates the final gesture ARG. To connect the sub-graphs, new edges are added based on the temporal order between the gestural events. These new edges link the nodes of the sub-graph for one gesture g_k to the nodes of the sub-graph for the next gesture g_{k+1}. The semantic relation of the new edges can also be easily identified. The temporal relations of the new edges are set as "*preceding*". As shown in Figure 3, the sub-graphs of the pointing gesture and the circling gesture are connected to form the final gesture ARG. Here the new edges are depicted using thinner lines.

4.3 Construction of History ARG

To exploit the conversation context, we use a *history* ARG to capture the information about objects that are in focus during the last interaction. This provides another source for finding the potential referents. Specifically, a history ARG consists of a list of objects that are in focus during the most recent interaction. Each node, called a *history node*, contains information related to the object in focus, including its identifier and its semantic type. Since the objects in the history can be referred by a user in an arbitrary order, the temporal relations between any two history nodes are labeled "*unknown*". Furthermore, the semantic relations are decided based on the known identities of the objects. Figure 4 shows the history ARG created after U3 (Table 1) is processed. This history ARG will be used to process the next input (U4). Since there are three houses in focus in U3, there are three nodes in the ARG. All three nodes are connected by the semantic type relations and the temporal relations.

[5] Currently, we use the inactivity (i.e., 2 seconds with no input from either speech or gesture) as the boundary to delimit an interaction turn.

5. GRAPH-MATCHING PROCESS

For each multimodal user input, we create three graphs: a speech ARG, a gesture ARG, and a history ARG as described above. Given these three ARGs, we formulate reference resolution as a probabilistic graph-matching problem under the following assumptions:

- A speech utterance may include multiple referring expressions. Some of them may refer to the objects selected by the gesture. Some of them may refer to the objects mentioned in the prior conversation.
- A referring expression may refer to an object that is not in the gesture ARG or in the history ARG. Some objects may be distinguished from the others by their unique features. For example, if there is only one red house shown on the graphic display, the user may use the phrase "the red house" to precisely refer to the intended target. Hence, our system also maintains an object list, called *secondary object list*, which records the objects that are not in the gesture ARG or in the history ARG, but are currently visible on the graphic display.
- A referring expression may refer to a group of objects (e.g., "these houses"). In such cases, we assume that referents should all come from one single information source, either from the gesture ARG, the history ARG, or the secondary object list mentioned above.

5.1 Referring Graph and Referent Graph

Under the above assumptions, we arrange the three ARGs into two graphs: a referring graph and a referent graph. A *referring graph* is basically the speech ARG. It contains a collection of referring expressions to be resolved. A *referent graph* is the aggregation of a gesture ARG and a history ARG. To build a referent graph, we add new edges to connect the gesture ARG and the history ARG. These edges connect every gesture node to every history node. The semantic type relations of these new edges can be easily inferred using the already known object identities. Based on our assumption that the objects from the history ARG and the objects from the gesture ARG cannot be referred simultaneously by one referring expression, the temporal relations of the new edges are set as "*non-concurrent*". If the size of the secondary object list is small, all objects on this list are added to the referent graph in the following manner. First, for each object in the list, a node is created and known as a *secondary node*. Then, edges are added to connect each secondary node to every node already in the referent graph. We set their semantic type based on their identities, and set their temporal relations as "*non-concurrent*".

Given a referring graph and a referent graph, reference resolution is a graph-matching problem, which aims to find the best match between the referent graph and the referring graph, while optimizing the satisfaction of various temporal, semantic, and contextual constraints.

5.2 Graph-matching Algorithm

First let us define notations.

- The referent ARG: $G_r = \langle\{a_x\}, \{r_{xy}\}\rangle$, where $\{a_x\}$ is the node list and $\{r_{xy}\}$ is the edge list. The edge r_{xy} connects nodes a_x and a_y. The nodes of G_r are called referent nodes.
- The referring ARG: $G_s = \langle\{\alpha_m\}, \{\gamma_{mn}\}\rangle$, where $\{\alpha_m\}$ is the node list and $\{\gamma_{mn}\}$ is the edge list. The edge γ_{mn} connects nodes α_m and α_n. The nodes of G_s are called referring nodes.

Our approach finds the best match between the referent graph and the referring graph by maximizing the following term with respect to $P(a_x,\alpha_m)$:

$$Q(G_r, G_s) = \sum_x \sum_m P(a_x, \alpha_m)\zeta(a_x, \alpha_m) + \sum_x \sum_y \sum_m \sum_n P(a_x, \alpha_m)P(a_y, \alpha_n)\psi(r_{xy}, \gamma_{mn}) \quad (1)$$

Here $P(a_x,\alpha_m)$ is the matching probability between a referent node a_x and a referring node α_m. Here $\Sigma_x P(a_x,\alpha_m) = 1$, if the speech referring node α_m refers to a single object.

The term $Q(G_c,G_s)$ measures the degree of the overall match between the referent graph and the referring graph. This term not only considers the similarities between the nodes in both graphs, but also considers the similarities between the edges in both graphs. The function $\zeta(a_x,\alpha_m)$ measures the similarity between a referent node a_x and a referring node α_m. $\zeta(a_x,\alpha_m)$ is defined based on the node properties of both a_x and α_m. The function $\psi(r_{xy},\gamma_{mn})$ measures the similarity between the edges r_{xy} and γ_{mn}. The domain knowledge is incorporated in the design of $\zeta(a_x,\alpha_m)$ and $\psi(r_{xy},\gamma_{mn})$. Currently these functions are empirically determined through a series of regression tests. In the future, these functions may be automatically learned from our experimental data.

We have adopted the graduated assignment algorithm [5] to maximize $Q(G_r,G_s)$ in (1). Our algorithm initializes $P(a_x,\alpha_m)$ using the selection probability if a_x comes from a gesture ARG and a pre-defined probability if a_x comes from a history ARG. It then iteratively updates the values of $P(a_x,\alpha_m)$ until the algorithm converges. When the algorithm converges, $P(a_x,\alpha_m)$ is the matching probability between a referent node a_x and a referring node α_m. Our algorithm will always find a most probable match between a referent node and a referring node. Based on the value of $P(a_x,\alpha_m)$, our system decides whether a referent is found for a given referring expression. Currently, if $P(a_x, \alpha_m)$ is greater than a threshold (e.g., 0.8), our system considers that referent a_x is found for the referring expression α_m. On the other hand, there is an ambiguity if there are two or more nodes matching α_m and α_m is supposed to refer to a single object. In this case, since there is no sufficient evidence from either inputs or the context for our algorithm to resolve these ambiguities, our system will ask the user to further clarify the object of his/her interest.

Note that the number of referent nodes may not match the number of the referring nodes. In other words, we may not

	G1: No gesture	**G2**: One gesture	**G3**: Multiple gestures	Total numbers
S1: No referring expression	2	1	0	3
S2: One referring expression	12	117	2	131
S3: Multiple referring expression	1	6	15	22
Total numbers	15	125	17	156

Table 2. Referring patterns from the user study.

find a match for a referent node or a referring node. To deal with this problem, a *null node*, which is a place holder and does not physically exist, is added to both a referent graph and a referring graph before the matching starts. Null nodes are the slack variables in the graduated assignment algorithm and provide matching destinations for unmatched referent or referring nodes.

6. SYSTEM IMPLEMENTATION

We have implemented the GUI and the gesture recognition component in C++ using Microsoft Visual C++ 6.0. The grammar-based parser for understanding speech utterances was implemented in Java, so is our graph-matching algorithm. Different components run as synchronized threads and communicate with each other using the TCP/IP protocol. The system runs in real-time under Windows 2000 with 1.1 GHz CPU and 512 MB of RAM..

7. EVALUATION

To evaluate our approach, we have conducted a preliminary user study. Eight subjects participated in our study. The subjects were asked to interact with the system using both speech and gestures to accomplish three tasks. The tasks were designed so that the subjects could explore different house or town information. For example, one task was finding the least expensive house in the most populated town. The voice was trained for each subject to minimize recognition errors.

Table 2 contains a summary of the referring patterns observed in this study. The columns indicate whether there is no gesture, one gesture, or multiple gestures in one input. The rows indicate whether there is no referring expression, one referring expression, or multiple referring expressions in one speech utterance. As shown in the table, the majority of observed user references involve only one referring expression and one gesture (e.g., [S2, G2] in Table 2). This is consistent with earlier findings [12]. Previous approaches work well with these simple references. However, in this study, we also found that 14.1% of the inputs were complex, consisting of multiple referring expressions from the speech utterances and multiple gestures (S3 in Table 2). Furthermore, 12.2% of gesture inputs were ambiguous where users did not precisely indicate the objects of their interest (the ambiguous data is not shown in Table 2). These behaviors are not observed in [12] partly due to the different interface design and the increased complexity of tasks.

	Recognized correctly	Recognized Incorrectly	Total
Identified correctly	83	26	109
Identified incorrectly	11	36	47
Total	94	62	156

Table 3: Overall evaluation results. The columns indicate the number of referring expressions that were correctly or incorrectly recognized by the speech recognizer. The rows indicate the number of referring expressions whose referents are correctly or incorrectly identified by the graph-matching algorithm.

Table 3 summarizes the performance of our approach in this study. Totally 109 out of 156 referring expressions are correctly resolved (about 70%). This overall performance is largely influenced by the poor speech recognition rate. Out of 156 referring expressions, only 94 are correctly recognized (60%). However, the results have confirmed the earlier findings that fusing inputs from multiple modalities together can compensate for the recognition errors through mutual disambiguation [18]. Among 62 referring expressions that are incorrectly recognized, 26 of them are correctly assigned referents by our algorithm.

Among the referring expressions that are correctly recognized, the accuracy of our approach is 88.3% (i.e., 83/94). The errors are mainly from the following sources: 1) vocabularies used by the users are not covered by our grammars (8 out of 11 cases). For example, “area” is not in our vocabulary. Thus any additional constraints expressed related to “area” is not captured. Therefore, our system cannot identify whether a house or a town is the referent when the user utters “this area”. 2) The speech and gesture inputs are unsynchronized within our turn-determination time threshold (i.e., 2 seconds). In one case, the gesture input occurred more than 2 seconds before the speech input. 3) In two other cases, spatial relations (as in “the house just close to the red one”) and superlatives (as in “the most expensive house”) were used but not interpreted correctly.

In this study, we focus on investigating ambiguous gesture inputs (similar to those handled by the unification-based approach [9]). No ambiguities from speech inputs were examined. Out of 19 ambiguous gestures, the references in 11 cases were correctly disambiguated. In seven cases, since there was no information from either the speech input or the conversation context to disambiguate the object of interest, the system asked the user for further clarification. Furthermore, when handling complex inputs, once the spoken expressions were correctly recognized, our approach achieved an accuracy rate of 92.9%.

8. CONCLUSION AND FUTURE WORK

This paper describes a novel technique that uses a probabilistic graph-matching algorithm to systematically resolve a wide variety of references during a human-computer multimodal conversation. Our approach incorporates temporal, semantic, and contextual constraints together in one framework and derives the most probable interpretation that best satisfies all these constraints. Preliminary user study results have shown that our approach can successfully resolve referring expressions, no matter simple or complex, precise or ambiguous, regardless of their referential forms.

Since our current implementation only considers ambiguities from gesture inputs, we plan to extend it to handle ambiguous speech utterances in the near future. Other plans for future work include the identification of speech disfluencies and gesture disfluencies using multimodal information. In our study, we have observed speech disfluencies such as speech repair and gesture disfluencies such as repetition (i.e., repetitively point to an object). Correctly identifying all these disfluencies will further improve reference resolution in a practical human-machine conversation.

1. ACKNOWLEDGEMENT

The authors would like to thank Keith Houck and the other colleagues in the Intelligent Multimedia Interaction Group at IBM T. J. Watson Research Center for their support. The authors would also like to thank Candy Sidner at MERL and the anonymous reviewers for their valuable comments on previous drafts of this paper.

2. REFERENCES

1. Bolt, R.A. Put that there: Voice and Gesture at the Graphics Interface. *Computer Graphics*, 1980, 14(3): 262-270.
2. Cassell, J., Bickmore, T., Billinghurst, M., Campbell, L., Chang, K., Vilhjalmsson, H. and Yan, H. Embodiment in Conversational Interfaces: Rea. In *Proceedings of the CHI'99 Conference*, 1999, pp. 520-527. Pittsburgh, PA.
3. Chai, J., Pan, S., Zhou, M., and Houck, K. Context-based Multimodal Interpretation in Conversational Systems. *Fourth International Conference on Multimodal Interfaces*, 2002.
4. Cohen, P., Johnston, M., McGee, D., Oviatt, S., Pittman, J., Smith, I., Chen, L., and Clow, J. Quickset: Multimodal Interaction for Distributed Applications. *Proceedings of ACM Multimedia*, 1996. pp. 31– 40.
5. Gold, S. and Rangarajan, A. A graduated assignment algorithm for graph-matching. *IEEE Trans. Pattern Analysis and Machine Intelligence*, *vol*. 18, *no*. 4, 1996.
6. Grosz, B. J. and Sidner, C. Attention, intention, and the structure of discourse. *Computational Linguistics*, 12(3):175-204. 1986.
7. Gustafson, J., Bell, L., Beskow, J., Boye J., Carlson, R., Edlund, J., Granstrom, B., House D., and Wiren, M. AdApt – a Multimodal Conversational Dialogue System in an Apartment Domain. *Proceedings of 6th International Conference on Spoken Language Processing (ICSLP),* 2000.
8. Huls, C., Bos, E., and Classen, W. 1995. Automatic Referent Resolution of Deictic and Anaphoric Expressions. *Computational Linguistics,* 21(1):59-79.
9. Johnston, M, Cohen, P., McGee, D., Oviatt, S., Pittman, J. and Smith, I. Unification-based Multimodal Integration, *Proceedings of ACL'97*, 1997.
10. Johnston, M. Unification-based Multimodal parsing, *Proceedings of COLING-ACL'98*, 1998.
11. Johnston, M. and Bangalore, S. Finite-state multimodal parsing and understanding. *Proc. COLING'00*. 2000.
12. Kehler, A. Cognitive Status and Form of Reference in Multimodal Human-Computer Interaction, *Proceedings of AAAI'01*, 2000, *pp. 685-689.*
13. Koons, D. B., Sparrell, C. J. and Thorisson, K. R. Integrating Simultaneous Input from Speech, Gaze, and Hand Gestures. In *Intelligent Multimedia Interfaces*, M. Maybury, Ed. MIT Press: Menlo Park, CA, 1993.
14. Neal, J. G., and Shapiro, S. C. Intelligent Multimedia Interface Technology. In *Intelligent User Interfaces*, J. Sullivan & S. Tyler, Eds. ACM: New York, 1991.
15. Neal, J. G., Thielman, C. Y., Dobes, Z. Haller, S. M., and Shapiro, S. C. Natural Language with Integrated Deictic and Graphic Gestures. *Intelligent User Interfaces, M. Maybury and W. Wahlster (eds.)*, 38-51,, 1998.
16. Oviatt, S. L. Multimodal interfaces for dynamic interactive maps. In *Proceedings of Conference on Human Factors in Computing Systems: CHI '96*, 1996, pp. 95-102.
17. Oviatt, S., DeAngeli, A., and Kuhn, K., Integration and Synchronization of Input Modes during Multimodal Human-Computer Interaction, In *Proceedings of Conference on Human Factors in Computing Systems: CHI '97*, 1997,
18. Oviatt, S. L. Mutual Disambiguation of Recognition Errors in a Multimodal Architecture. In *Proceedings of Conference on Human Factors in Computing Systems: CHI '99*.
19. Oviatt, S.L., Multimodal System Processing in Mobile Environments. In *Proceedings of the Thirteenth Annual ACM Symposium on User Interface Software Technology (UIST'2000)*, 21-30. New York: ACM Press.
20. Stent, A., J. Dowding, J. M. Gawron, E. O. Bratt, and R. Moore, The commandtalk spoken dialog system. Proc. ACL'99, 1999, pages 183–190.
21. Stock, Oliviero, ALFRESCO: Enjoying the combination of natural language processing and hypermedia for information exploration. *Intelligent Multimedia Interfaces, M. Maybury (ed.)*, 1993, pp. 197-224.
22. Tsai, W.H. and Fu, K.S. Error-correcting isomorphism of attributed relational graphs for pattern analysis. *IEEE Trans. Sys., Man and Cyb.*, vol. 9, 1979, pp. 757–768.
23. Wahlster, W., User and Discourse Models for Multimodal Communication. *Intelligent User Interfaces, M. Maybury and W. Wahlster (eds.)*, 1998, pp 359-370.
24. Zancanaro, M., Stock, O., and Strapparava, C. 1997. Multimodal Interaction for Information Access: Exploiting Cohesion. *Computational Intelligence* 13(7):439-464.
25. Zhou, M. X. and Pan, S. Automated authoring of coherent multimedia discourse for conversation systems. Proc. ACM MM'01, pages 555–559, 2001.