MUTUAL DISAMBIGUATION OF RECOGNITION ERRORS IN A MULTIMODAL NAVIGATIONAL AGENT

by

Amy Chu

B.S., Computer Science and Engineering
Massachusetts Institute of Technology, 2002

Submitted to the Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degree of

Master of Engineering in Electrical Engineering and Computer Science
at the
Massachusetts Institute of Technology

May 2003

Signature of Author ______________ ______________
Department of Electrical Engineering and Computer Science
May 21, 2003

Certified by ______________ ______________
Dr. Henry Lieberman
Research Scientist, MIT Media Lab
Thesis Supervisor

Accepted by _________ ______________
Arthur C. Smith
Chairman, Department Committee on Graduate Theses
Department of Electrical Engineering and Computer Science

MUTUAL DISAMBIGUATION OF RECOGNITION ERRORS IN A MULTIMODAL NAVIGATIONAL AGENT

by
Amy Chu

Submitted to the Department of Electrical Engineering and Computer Science on

May 21, 2003

in Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science

ABSTRACT:

Recognition-based technology has made substantial advances in the past few years because of enhanced algorithms and faster processing speeds. However, current recognition systems are still not reliable enough to be integrated into user interface designs. A possible solution to this problem is to combine results from existing recognition systems and mutually disambiguate the unreliable sections. In addition, the results of one recognition system can be used to prepare the other recognition system. We experimented with an approach that uses a software agent to integrate off-the-shelf speech and character recognition applications via scripting languages. We describe how a software agent called MARCO, Multimodal Agent for Route Construction, utilizes multiple recognition systems to assist users in giving directions for urban navigation.

Thesis Supervisor: Dr. Henry Lieberman
Title: Research Scientist, MIT Media Laboratory

Contents

1 Introduction

For years, user interface designers have refrained from using recognition-based systems because technologies such as speech recognition and optical character recognition (OCR) have not achieved standalone reliability. High error rates and lack of graceful error handling are the major barriers to widespread use of recognition technology. More reliable recognition systems can be designed, however, through the use of multimodal recognition architectures that combine partial input from several existing recognition technologies. These commercial recognition applications are used in their original, unaltered state as opposed to creating new recognition algorithms or directly interfacing with the API's of recognition systems.

Even if we had perfect recognition, the mundane interface details of organizing the information—getting the raw data in and out of the recognition programs and putting it in a coherent, usable form—would be daunting for many everyday applications. This thesis describes how we used a software agent called, MARCO, Multimodal Agent for Route Construction, to integrate input from speech recognition, OCR, and user task context. The agent assists users in giving urban navigational directions through digital photographs and verbal dictation and displays the results as a series of route descriptions. By integrating the results of voice recognition and OCR, MARCO is able to increase the recognition accuracy for the names of landmarks. The challenge was to make a system that collects the audio and image data, relates it to any other data sources that might be available (such as GPS), and compiles a set of coherent, readable directions to be transmitted to another user.

Before we get into the nuts and bolts of the MARCO system, we will briefly discuss the goals of this thesis. The next two sections outline the motivation and the research objectives of this thesis.

1.1 Motivation

In CHI 2002 (Lieberman, 2002), Lieberman described the motivation for a multimodal navigational assistant. While image and audio input is attractive for this task and the capture of such input is now feasible on handheld devices, two barriers stand in the way—the accuracy of recognition and the integration and organization of the output for this task. Lieberman set out to determine if, first, there was enough redundancy of information in the task to provide a basis for disambiguation, and second, if existing recognition algorithms were able to achieve sufficient recognition. Preliminary results were positive, showing 75% combined landmark accuracy—the percentage of landmark names and phrases necessary for the directions that were recognized by either the speech recognizer or the OCR (Lieberman, 2002). This thesis builds upon the findings from the system described in Lieberman's paper (Lieberman, 2002) to show how verbal dictation with digital photography can help urban navigation. An improved feasibility test shows 94% combined landmark accuracy, which will be discussed in more detail later in the thesis.

1.2 Research Objectives

This thesis discusses three distinct topics: creating a system for generating directions to a location in a city, using a scripting language to "stitch" together off-the-shelf applications, and combining different modalities from various recognition systems with user-context to provide more robust results and services. Though this thesis has three focuses, the research objectives of this thesis are to investigate the general issues of combining OCR and speech recognition and whether: 1) the combination derives more accurate results, 2) the combination saves the user time during the presentation step, and 3) existing recognition technology can reasonably recognize the images and audio captured by current handheld computers. We discuss pragmatic and software-related issues in multimodal system implementation, using the navigational assistant as a preliminary feasibility exploration.

Before we delve into the thesis, we will go over a brief roadmap of the rest of the thesis. The next chapter will look at the first topic of this thesis by describing the intended use of the MARCO system and stepping through a suggested user scenario. Chapter 3 investigates whether current speech and character recognition technologies are sufficient for the task of generating urban directions. Chapter 4 goes through the overall workflow of MARCO, as well as the specific methods and strategies used to collect and process the image and audio information. Chapter 5 tackles the second topic of this thesis and discusses how MARCO combines the speech and character recognition results. Then Chapter 6 explores the third topic by describing how MARCO uses a scripting language to "stitch" together off-the-shelf applications. Chapter 7 evaluates the usability and effectiveness of MARCO. Chapter 8 details the current research on extending MARCO to the handheld platform, while Chapter 9 surveys related prototypes and similar research in multimodal recognition systems. Chapters 10 and 11 wrap up by summarizing the contributions of this thesis and laying out future work and applications.

2 An Assistant for Urban Navigation

When issuing an invitation to an event, the event organizer often includes directions that will enable the invitee to find the way to the event's location. Services such as Mapquest, shown in Fig. 1, can automatically generate street maps given a starting point and a destination, but these methods have their limitations. These map services cannot take advantage of visible landmarks ("Turn right at the Petco store."), or deal with routes that are partly walking, partly public transit ("Change for the Blue Line at Government Center").

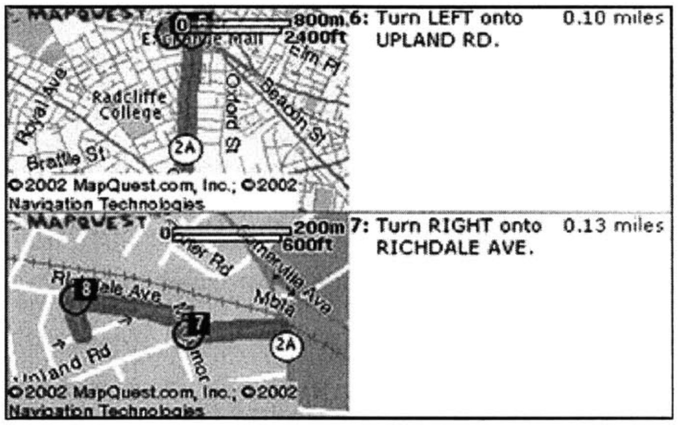


Fig. 1. Example Route from Mapquest.

Locating devices such as digital tags and global positioning systems (GPS) also have their limitations. Digital tags are still not prevalent and unless there is a tag in every street sign, we cannot use tags effectively in urban navigation. Similarly, GPS is only available outside and is not accessible inside a mall or an office building. Another problem to consider is the constant evolution of our environment. If the street signs change, it would be better to take recent photographs of the existing signs instead of relying on Mapquest, digital tags, or GPS, which may not be updated.

Instead of using web-based services or digital tags, we propose that the user will have a personal digital assistant (PDA) and a digital camera and demonstrate the route that he would like his guests to follow, physically traveling between the starting point and the destination. This approach to navigation is especially well suited to urban settings.

First, we can cater the directions to what a typical commuter would follow in an urban city. We can create routes that are partly pedestrian and partly public transportation. Since an urban commuter is mostly likely traveling on foot, we can take advantage of one-way streets (i.e., the user could walk down a one-way street in the opposite direction). We can also take shortcuts that are not confined to traditional roads.

Second, we since we are taking pictures along the route, we can utilize any information that is provided in the photograph that can be passed onto the recipient of the directions. Visible landmarks like stores, restaurants, sculptures, and other objects can be captured in a photograph. When a user later tries to follow the directions, he can refer to the image to verify that he is going in the right direction. Also, some cities, especially Boston, have small or hidden street signs and numerous one-way streets. Capturing these signs in a photograph can help direct attention to the correct locations.

These features make MARCO ideal for urban navigation. In the next section, we further demonstrate how MARCO aids urban navigation by providing a comprehensive example of a user's interactions with MARCO and illustrating the full-range of the system's capabilities.

2.1 Suggested User Scenario

MARCO assists the user in giving directions by annotating photographs of key landmarks with the user's narration of each direction step. After the user is finished walking through the desired route, MARCO organizes the information into a cohesive set of direction steps and displays them in a web page.

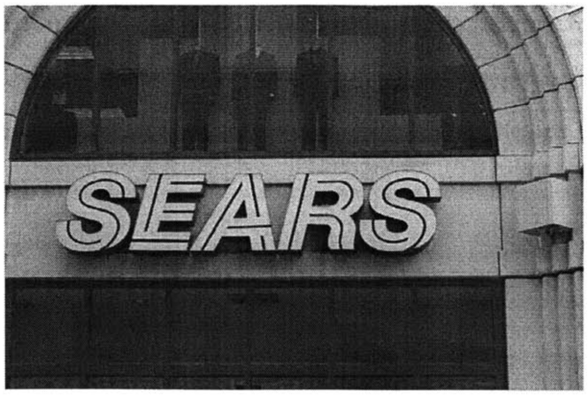
We present an example here to illustrate the system's core functionality, the mutual disambiguation of the speech recognition and the OCR, and also the user-to-agent interactions.

The User's Situation

Our hypothetical user, Tom, is the owner of a music store in the CambridgeSide Galleria mall. Tom, aware of the wealth of college students in the Boston and Cambridge area, wants to target individual colleges with students majoring in music. One such college is the Berklee College of Music near the Hynes Convention Center subway station. He wants to create flyers and emails that contain directions from Berklee to the mall. Since most college students do not own cars in Boston, Tom decides to create a set of directions that utilize both public transportation and pedestrian routes between Berklee and the CambridgeSide Galleria.

Interactions with MARCO

Tom picks up his PDA, with a digital camera and a microphone built-in, and starts MARCO. He begins to demonstrate the route that he would like the potential customers to follow. Starting at the Hynes Convention Center subway station, Tom physically travels along the expected route that students would take to get to the mall. Along the way, Tom takes digital pictures of key landmarks (street signs, store names, etc.) and dictates directions for each step. For example, Tom walks by a Sears department store entrance, takes a picture of the store sign, and says into the audio recorder of the PDA, "Walk past the Sears store entrance on your left." The photograph of the landmark and the associated user dictation are illustrated in Fig. 2:



"Walk past the Sears store entrance on your left."

Fig. 2. Example Landmark Photograph and User Dictation.

Once Tom finishes walking through his desired route, MARCO will automatically run the images and the dictation through the OCR and the voice recognition system and present the combined results for each photograph-dictation pair to the user. For the photograph and dictation provided in Fig. 2, MARCO might produce the following results:

Voice Recognition:	Walk past the seers store entrance on your left
Inverted OCR:	SEARS
Non-Inverted OCR:	AD
Combined Recognition:	Walk past the SEARS store entrance on your left

Tom can then click "OK" if the suggested combined recognition is correct, as in this case. He also has the option to correct the suggestion directly before clicking the "OK" button. Let's say, for example, that the OCR did not recognize the image of the store sign correctly. Instead, the inverted character recognition result was "EAD." Then MARCO's suggested combined recognition would be "Walk past the EAD store entrance on your left." When presented with the suggested combination, Tom can change "EAD" to "Sears" and click "OK."

MARCO runs through a similar procedure for each photograph-dictation pair, each time prompting Tom for his verification. When all of the photograph-dictation pairs have been processed, MARCO presents the entire route in a web page.

The complete route from the Hynes Convention Center subway station to the CambridgeSide Galleria mall is illustrated in Appendix A. The annotated images are an example of the detailed directions the MARCO agent produces for Tom. Since this is a route that requires both public transportation and walking, services like Mapquest would not be able to provide sufficient details. In addition, Mapquest only provides street connectivity and 2D maps, whereas MARCO creates a 3D view of the actual route. The corresponding Mapquest directions from the Hynes Convention Center subway stop to the CambridgeSide Galleria mall are listed in Appendix B for comparison with the MARCO directions.

Features of MARCO

We give a brief review of the features illustrated by the above scenario. First, MARCO processed the recognition of the images and the narration collected by the user. The system automatically combined the speech recognition and the character recognition of the given image and narration, and displayed the information to the user for verification. If Tom found the suggested combination to be correct, then he committed the results; however, when the suggestions were incorrect, Tom typed in the correct result without much hassle. After Tom verified the combined results, MARCO presented the entire direction set, complete with the original photographs, in a web page.

While this suggested navigational assistant gives the map users additional context via photographs of the actual route, it is not as flexible as route-creation systems like Mapquest that dynamically generate a route between any two points. Though an event organizer's destination is fixed, it is rare that guests have the same starting location. However, the main objective of creating this prototype is not to design a map generator that will compete with Mapquest, but rather to demonstrate that combining OCR and speech recognition technology can improve the overall accuracy of recognition.

Before we proceed to examine the capabilities of current recognition technologies, we examine the various approaches to the recognition problem that we considered before finally settling on the current MARCO system.

2.2 Unusual Approaches to the Recognition Problem

Unimodal Recognition v. Multimodal Recognition

There are multiple ways to approach the high error rates of present recognition technologies. One solution would be to continue perfecting current recognition procedures or to design our own recognition algorithms. Existing recognition procedures can be refined through improved recognition rates or through the use of error correction techniques. Even though we do expect recognition algorithms to improve, we suspect that it is not necessary to wait for improved recognition to achieve usable results for this task. Multimodal recognition systems have an advantage in that they are able to use the additional input as error checkers or as dictionary preppers. As you will see later in the paper, we try to improve the OCR by altering the images and employing patterns to correct common errors, to eliminate garbage values, and to identify extraneous words that commonly appear on signs.

New Technology v. Existing Technology

When designing our multimodal recognition system, we chose to take advantage of existing, off-the-shelf recognition technology. These commercial technologies have already perfected their recognition processes and are leaders in their respective industries. Integrating these technologies can be difficult because they were not designed to work with other recognition technologies and each system has its own idiosyncrasies and weaknesses. Though these technologies have their limitations, we can avoid starting from scratch and testing each individual system.

Interfacing with APIs v. Stitching Together Applications

Another unusual aspect of our approach is the use of a software agent to stitch together the off-the-shelf software. If we wish to enhance recognition technologies by combining them with each other, a possible solution would be to design a program that interfaces with the APIs of the two recognition systems. This method, however, requires in depth knowledge of the structure and architecture of each specific recognition application and does not allow for easy transitions to alternative applications in the future.

In lieu of interfacing with the APIs of the recognition systems, we chose to utilize an agent to manipulate the commercial applications. A software agent is able to interact with the applications much like a human user would (Lieberman, 1998). The agent can mimic the user by selecting menus, clicking buttons, and typing words. Nonprofessional users tend not to explore the mechanisms of individual recognition systems. They prefer to treat programs like abstract black boxes, inputting information and receiving the output without actually viewing the process. A software agent serves as an interface between the users and the applications.

The drawback is that the commercial applications used by the MARCO software agent were not designed for this purpose; thus, there are several limitations to the system that must be solved through alternative means. The character recognition system (Caere OmniPage) was designed to recognize scanned documents, not color digital photographs. The speech recognition system (IBM ViaVoice), a speaker-dependent, large vocabulary recognition system, was designed to understand dictated documents and application commands. The user must train any additional vocabulary in his specific voice. Despite these drawbacks, however, integrating the commercial applications together at a higher level provides for easier debugging and a more intuitive process. Therefore, we chose to stitch together the current technologies at the appl ication level.

In the following chapter, we examine the capabilities of current recognition technologies and determine whether the state of current systems is up to the tasks laid out in the user scenario.

3 Is Present Speech Recognition and OCR Up to the Task?

To determine the feasibility our urban navigational assistant, we need to answer two questions: Is there enough information in typical route data? And are the recognition algorithms good enough? Lieberman performed a preliminary experiment in (Lieberman, 2002) to answer these questions by collecting data for some sample routes. To supplement those feasibility tests, we performed new tests with improved data capture methods and a different set of routes. The routes involved a combination of walking and public transportation. The preliminary simulations allowed us to analyze the potential for mutual disambiguation of speech and character recognition in the urban navigation scenario. The tests also allowed us to mine valuable data and sample routes that we could use during our evaluation and usability tests. In the next section, we explain the methodology of the feasibility tests and describe the obstacles we encountered during the tests.

3.1 Methodology and Challenges

We interpreted the data gathered during the feasibility tests in two phases. First, we ran the narration through a commercial speech recognizer, IBM ViaVoice, a large vocabulary speaker-dependent speech recognizer. The narration was dictated in real time through a desktop microphone and the results of the speech recognition were displayed in a word processing document. The speech recognizer was trained with only a minimal training session, the dictation of a sample story, before the data collection began. Second, we ran the images through a commercial OCR program, Caere OmniPage, in order to recognize the written information in street signs, store entrances, etc.

A challenge in interpreting the data was that the recognition programs were not intended for translating signs or directions. ViaVoice is for dictated documents in an office setting; OmniPage is intended for the interpretation of text from scanned paper documents. We had, for example, to process the images to make them acceptable to OmniPage by inverting them to enable the recognition of light-on-dark sign text. Realistically, the camera angle is usually skewed and the user does not ensure that the sign is straight. Thus, we must reorient the photograph to ensure maximum recognition. In addition, most of the data was "dirty" from excess noise and improper lighting in the photographs. We had to "clean" the images by tweaking them in Adobe Photoshop. We adjusted the resolution, size, orientation, contrast, etc.

We describe the results from our feasibility tests in the following section.

3.2 Feasibility Results

As might be expected, the reliability of recognition for both the speech recognizer and the OCR was pretty poor. The speech recognizer processed proper names such as street names incorrectly. This could be improved by preloading the speech recognizer with a dictionary of street names or subway station names for the city in question, as listed in map services and yellow pages of sites like Yahoo.

The OCR often missed information on signs, especially when the sign was tilted with respect to the plane of the camera, (as in cross streets on corners). Some of this could be alleviated by more careful image capture. We were also able to rectify some of these problems simply by rotating the images in both directions and trying again.

Table 1 shows, respectively, some images, the actual narration, its interpretation by the speech recognizer, and the results of the OCR. Even though some results are amusing, it is remarkable how the recognizers identified most essential information.

Table 1: Example Speech Recognition and OCR Output.

Narration	Start out at the corner of Sciarappa and Charles	You'll see the Lechmere subway station on your left	Change at Government Center for the Blue Line
Voice reco	Start out at the corner of sheer rock band and Charles	You'll see the leech year subway station on your lap	Change at government center for the blue logic
OCR	SCIARAPPA ST	LECHMERE	BLUE LINE TO AIRPORT & ALL TRAINS

The improved feasibility studies show greater than 94% landmark recognition accuracy when utilizing multiple input sources. We analyzed 10 routes and 92 direction steps. In the numbers below, we present the raw word-level accuracy rates and the landmark accuracy rates of the speech recognition and the OCR. The combined landmark accuracy is the percentage of landmarks recognized by either the speech recognizer or OCR.

Speech reco word-level accuracy:	82.5%
Speech reco landmark accuracy:	64.7%
OCR word-level accuracy:	48.5%
OCR landmark accuracy:	46.1%
Combined landmark accuracy:	***94.7%***

First, note that the landmark accuracy is worse than the word-level accuracy for the speech recognition. This is because landmarks tend to be proper names easily missed by the speech recognition. This could be considerably improved if a street directory were used to train the speech recognizer. Second, it is encouraging that the combined landmark accuracy is significantly higher than the speech landmark accuracy. This is because the landmarks most likely to be misinterpreted by the speech recognition are often those that would be most likely to appear on street signs. Third, it is not necessary to achieve 100% landmark accuracy to make a useful system, because the user can be queried about possible ambiguities, and a reasonably high landmark recognition rate would still save labor for the user. Finally, the 94.7% landmark accuracy rate is an upper bound. When calculating the combined accuracy rate, we disambiguated the results by hand, choosing the best results from each of the recognition outputs. Hand simulations, using techniques described in the next chapter, lead us to expect that we can achieve disambiguation results approaching this upper bound.

4 The Nuts and Bolts of MARCO

When designing a software agent to aid a user in completing a task, it is best to start off by simulating the steps required to complete the task. We physically walked through a typical urban route, taking pictures and narrating directions along the way. During the simulations, we noted any difficulties with capturing the input and any deficiencies with the input devices.

Then we fed the gathered information (the images and the audio) into various recognition and third-party applications, noting the necessary operations and parameters. Since every application has its shortcomings, sometimes we had to 'tweak" the input and application settings or reverse-engineer the process to get our desired results. This testing of various applications allowed us to determine the capabilities of each program and the effectiveness of each interface. During this preliminary process, we noted the pros and cons of the chosen applications and devices. Details of these assessments are given in the following sections.

4.1 General Workflow

After performing the user and application simulations, it was apparent that it might not be possible for us to build a working commercial system within a reasonable time frame given our current set of technologies. Therefore, the aims of the preliminary system described in this thesis are to demonstrate the feasibility of creating a more robust multimodal recognition system in the future. This higher-level goal is apparent in the following workflow outline of the user tasks and computer procedures:

image capture

|

image alteration

|

image OCR

|

narration capture

|

narration speech recognition

|

results integration and presentation

While performing each task, we also encountered many inessential limitations of the commercial technologies that we used. All applications have many idiosyncratic requirements (e.g., the input files must be configured in a certain way). Part of stitching together the applications required us to 'paper over" and simulate over these limit ations because we are not re-programming the applications. By manipulating image and audio files and enlisting the help of third-party applications, we used the software agent to implement the first five steps of the workflow for the user. We also developed heuristics for automatic data selection in the results integration step.

The first five workflow steps are described in the ensuing sections. The last step, results integration and presentation, is the most complicated step and has an entire chapter dedicated to it. For each of the workflow steps, we describe some of the 'tweaks" and wor karounds we needed to perform to accomplish our workflow requirements. Keep in mind that the exact details of these steps are not important. The details might change with another choice of application (or even with another version of the same application). But we present them here because they illustrate typical maneuvers necessary to adapt a general-purpose application for a specific user scenario. No application developer can anticipate all possible uses of his application. These kinds of tricks will often be necessary, and we want to give the reader a feel for the process involved.

4.2 Image Capture

The first user task is image capture. The user must walk along the proposed route, taking digital photographs of key landmarks along the way. These images must then be loaded into the computer as JPEG files before we can manipulate them and run them through recognition software. Results from the feasibility tests showed that problems commonly associated with taking random digital photographs are rotated images, skewed camera angles (i.e., the object is not at the same horizontal level as the camera lens), insufficient lighting, glares from camera flash, dirty signs, objects blocking the field of view, light-colored words on a dark background, and excess noise from unzoomed photographs. Fig. 3 is a photograph that exhibits most of the problems associated with taking random digital photographs. Most of these problems occur independently of the user, so instead of asking the user to retake a photograph, MARCO automatically manipulates a given photograph based on the common problems observed in the image capturing process.

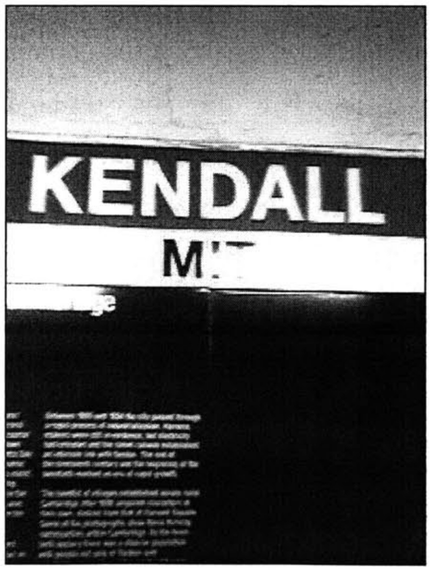


Fig. 3. Photograph Exhibiting Common Image Capture Problems.

4.3 Image Alteration

Since OmniPage was designed to recognize scanned paper documents, the technology expects files with specific properties (e.g., 200 dpi resolution, 'PICT" and 'TIFF" file formats, and black words on a white background). Therefore, each of the problems (rotation, skewed angles, glares, etc.) encountered during the previous step contributes to a high OCR error rate. To overcome the limitations of OmniPage and obtain higher recognition rates, we have to alter the images prior to running them through the OCR. In order to ensure that the user does not manually intervene during the image alteration process, we used Adobe Photoshop—a commercial photograph design and production tool—to rectify most of these image imperfections and to simulate the appearance of a scanned paper document. After the user submits a given image, MARCO changes the photograph in three ways: cropping, reorientation, and paper simulation.

Cropping

First, MARCO asks the user to crop the initial photograph to eliminate any excess noise that may be picked up by the OCR. Fig. 4 shows an example of a photograph that needs to be cropped to produce more reliable recognition results. The dotted box in the figure indicates the ideal place to crop the image, since the user probably wants MARCO to recognize 'CAMBRIDGE ST" and not the 'DO NOT ENTER" or the 'NO TURN ON RED" signs. MARCO keeps a copy of both the cropped image and the uncropped image so that the final set of directions contain the original, unaltered image.

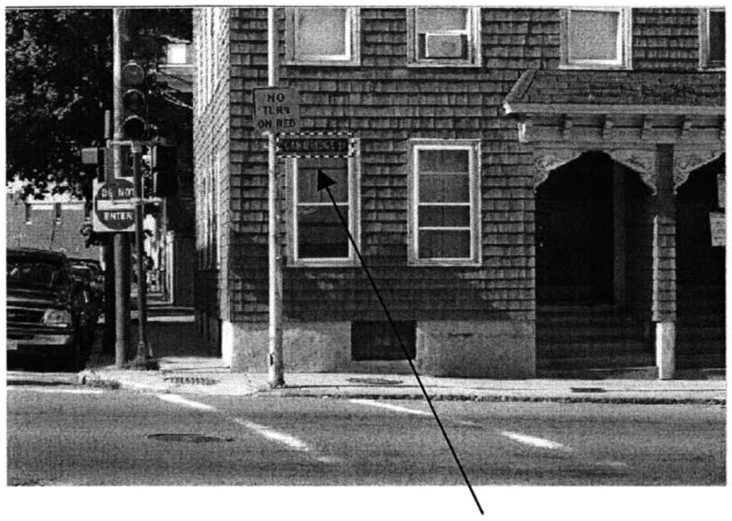


Fig. 4. Photograph Containing Excess Noise that Needs to Be Cropped.

Reorientation

Second, MARCO reorients the position or angle of a photograph. Photoshop does not have an automatic image alignment function. However, the OCR application can automatically rotate an image if the words are crooked. Though Omnipage can automatically align images during the recognition process, rotation of images is limited because the application is designed for scanned documents and it expects that scanned documents be rotated by either a few degrees or by increments of 90 degrees. Therefore, if the image pixel resolution is too low or if the words are light on a dark background, the image may be rotated 90 degrees, resulting in an empty result set or a set of garbage values. We used a Kodak DC 290 digital camera with a 720 x 480 pixel resolution. We found this pixel resolution to be sufficient for most sign character recognition.

In theory, we could use Photoshop's 'Rotate by d egrees" function to rotate the image clockwise by a certain increment. If we rotate the image by several increments and aggregate the results, then the results are likely to contain the optimal result. For the purposes of this thesis, however, we chose to allow OmniPage to automatically align the images.

Paper Simulation

Third, MARCO tries to simulate the appearance of a scanned office document through a series of additional modifications that include setting the resolution to 200 dpi, saving the image as a PICT file, inverting the image so that light-colored words on a dark background become dark-colored words on a light background, switching to grayscale, resizing the file, and ramping up the contrast or brightness level. Fig. 5 shows an example of an image before and after MARCO runs it through paper simulation.

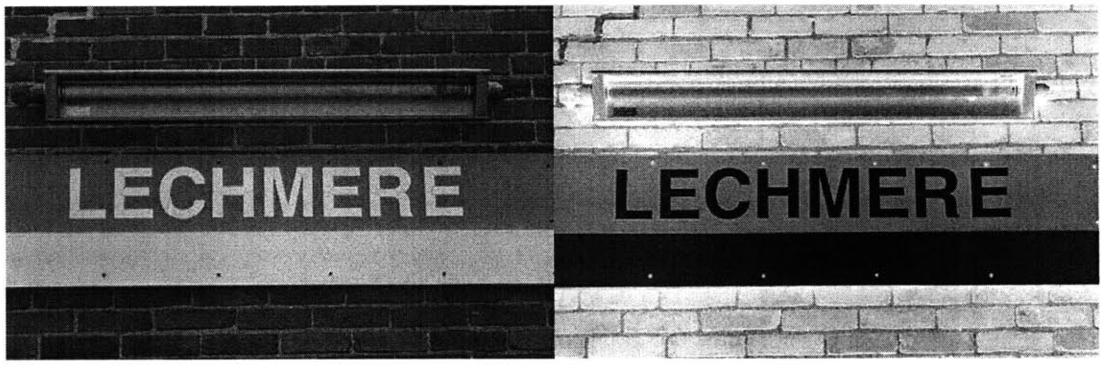


Fig. 5. Image Before and After MARCO Runs It through Paper Simulation.

The large number of alterations makes successive calls to the functions both tedious and time-consuming. In lieu of performing each of these functions individually, we decided to use Photoshop Actions to package up all of the adjustments. A Photoshop Action is a series of commands that you play back on a single file or a batch of files. For example, you can create an action that applies an Image Size command to change an image to a specific size in pixels, followed by an Unsharp Mask filter that resharpens the detail, and a Save command that saves the file in the desired format. With Photoshop Actions, we can record all of the required functions into a single action that can be performed on multiple files in the future. Photoshop Actions can record most, but not all, Photoshop commands, including commands that are not directly accessible through Applescript.

4.4 Image OCR

Once the necessary alterations have been made to the images, we can proceed to run the images through the OCR application. Several issues that needed to be considered during the image recognition process include scanned versus digital images, text selection, spelling and sanity checks, and inversion versus non-inversion.

Scanned v. Digital Images

Since OmniPage was designed to recognize scanned documents, we tried scanning some of our images to see if we would get more reliable results. The scanned photographs did not, however, produce better results than digital photographs. In fact, depending on the quality of the scanner, the results of scanned photographs may be worse.

Text Selection

The OCR produces better results if the image is cropped or if only the desired area to be recognized has been selected. This improvement is usually performed prior to the character recognition if the user correctly cropped the original photograph. In the event that the user did not sufficiently crop the image, OmniPage has an "auto select text" feature that attempts to select all text that appears in the image. With this feature, we can filter out the excess words recognized from the image. When researching common ways to give directions, we found that there are certain words that commonly appear on or near street, store, and restaurant signs. Some examples include "open," "close," "sale," "warning," "enter," "walk," "crossing," "stop," "speed," "parking," "right," "left," "merge," "law," "construction," "ahead," etc. If any of these words appear in the OCR result set, we can filter them out. There is the possibility, of course, that those words are part of the landmark that the user wants to recognize. However, most times these words were picked up from nearby traffic signs or sale signs.

Spelling and Sanity Checks

There is a spell check available while running OmniPage, but our system does not utilize it because we are aiming for minimal user interaction. While spelling check might not be worthwhile due to the frequency of proper names in street signs, we included "sanity checks" that might prevent such OCR-specific errors as the misrecognition of a "B" as an "8", or constraints that a word be plausibly pronounc eable. Table 2 contains several common errors in OCR and their respective corrections:

Table 2: Common OCR Errors and Their Possible Corrections.

Common Error	Possible Correction
<nothing>	image is inverted
"or "	H
(or)	l
[	C
0	O
2	Z
4	H
5	S
6	b
7	T
8	B
9	q

In addition to automatically correcting common OCR errors, we can eliminate parts of a result set if it contains too many garbage values. For example, if the OCR results contain punctuation marks or numbers, we can assume that some characters were not recognized correctly. Another example is if the OCR results contain more than three consonants in a row. If there are a series of punctuation marks, numbers, or consonants in a particular word of the results, then we can eliminate the entire word since it would not be helpful. Sometimes, the entire result set may be filled with garbage values. In this case, we can invert the image and examine those results before defaulting to the speech recognition results. The inversion process is described in the following section.

Inversion v. Non-Inversion

Since some signs have dark words on a light background and other signs have light words on a dark background, we run each image through the OCR twice (on the inverted image and on the non-inverted image). Generally, one of the two OCR results is empty, and we need only look at the other set of OCR results. On the off chance that neither set of results is empty, as is the case for an image such as the one shown in Fig. 6, we look at both sets of results and try to eliminate one or more of the options based on the "sanity checks" described previously.

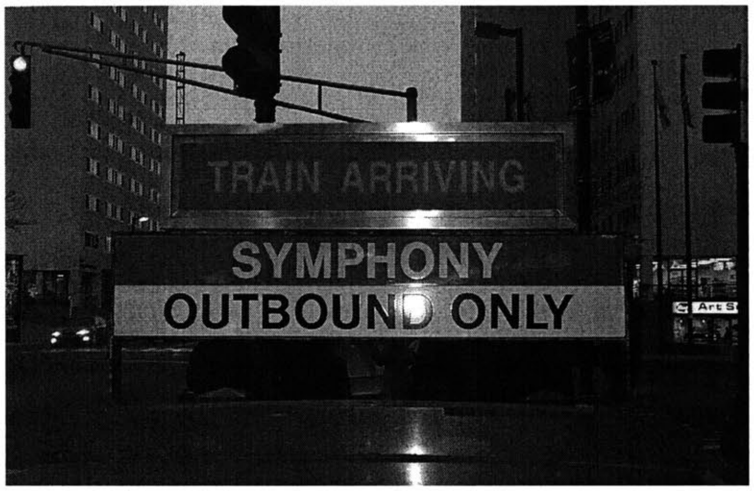


Fig. 6. Photograph Containing Both Light Words on a Dark Background and Dark Words on a Light Background.

4.5 Narration Capture

During the feasibility tests, narration of route directions was recorded in real-time through the built-in microphone in the digital camera. The sound clips recorded a lot of static and extraneous noises often muffled the user's voice, so the speech recogn ition was not able to pick up any coherent sentences. We were also limited by low-quality 8-bit audio recorded by our camera. We expect these limitations to be overcome with better quality sound devices and noise cancellation methods in the future. Presently, we make up for the low quality of the built-in microphone by redictating the narration directly into the speech recognition system after the images have been recognized.

4.6 Narration Speech Recognition

Before each image is processed through the OCR, we first dictate the narration into the ViaVoice recognition system. As we dictate, the system displays the results in a text document.

ViaVoice has an "Analyze My Voice" function that teaches the system your voice so your speech can be recognized with greater accuracy. The function measures the noise level of the room, measures the volume of the microphone, and processes the pronunciation and speech patterns after the user dictates a short passage for 10-15 minutes. Since the system has a limited dictionary, it does not recognize rare or proper nouns such as street names or store names. Instead, the user must train each new word by typing the word and recording the pronunciation.

Now we will describe the heuristics used to automatically select data in the results integration step.

5 Mutual Disambiguation of Recognition Results

The most complicated step of the MARCO workflow is the integration of the two sets of recognition results. Now that we have the results of the speech recognition and optical character recognition, we must decide which words to use from each system. There are two approaches to integrating the results: 1) take the results of the OCR and feed it into the speech recognition technology, or 2) look at the speech results and determine which part agrees or conflicts with the OCR results. We explored both approaches, before settling on the second method of disambiguation.

5.1 Feed OCR Results to Speech Recognizer

The first approach is to take the results from the OCR and feed them into the speech recognition system. By feeding the results into the speech recognition dictionary, we can prep its vocabulary with the words, mostly proper nouns, which are likely to show up during the user's narration. The same method cannot be applied in the other direction, however, because the OCR results are usually a subset of the speech results. While the speech recognizer captures most of the user's sentence, the OCR generally captures the proper nouns, the objects of the sentence. A major problem with this approach is that ViaVoice only permits additional vocabulary that is supplemented with a recording of the word's pronunciation. This would require additional user interaction, which may contain errors in it.

5.2 Selectively Replace Speech Results with OCR Results

Therefore, we decided to explore the second approach: look at the speech results first, determine which part of the results is the speech recognizer's interpretation of the landmark, and see if that interpretation agrees with the OCR's interpretation. Since landmarks most likely to be mistaken by the speech recognition are those that would appear on street signs, the agent would know that the results from the OCR might be more accurate than the results from the speech recognizer.

Sentence Patterns of Route Directions

Before we implemented the second method, we analyzed the sentence patterns of route directions. To facilitate the analysis, we examined the speech acts of common route directions. Similar directions can be grouped together based on word connotation and user context. For example, there are multiple ways to tell a person that they will pass Lechmere on the right:

'On your right, you will see *Lechmere*,"
"*Lechmere* will be on your right,"
'Staying to the left of *Lechmere*,"
'Pass to the left of *Lechmere*,"
'With *Lechmere* on your right,"
'Continue down the street, passing *Lechmere* on your right," and so on.

We compiled hundreds of these speech acts by analyzing examples of route directions, taken from the Internet or from people's emails of directions to parties. They can be grouped into about a dozen different categories including where to start from, where to exit, where a landmark is in relation to another landmark, where to go straight, when you are facing something, when to transfer or get off at a stop, where to turn left or right, etc. We collected over 500 examples and compiled a comprehensive list of sentence patterns. Utilizing these groups of sentence patterns, we wrote Lisp programs to alert the agent when a landmark is expected in a sentence.

Landmark Extraction from Speech Results

By comparing the speech results obtained by MARCO to the database of common direction patterns described in the previous section, we were able to extract the landmark noun or nouns from the speech results. We used a simple LISP program to pattern match the results to the list of direction patterns. The Lisp programming language is particularly efficient in the area of segment pattern matching. Segment matching differs from regular pattern matching in that the pattern can be matched to zero or more items. Implementing segment pattern matching allows us to anticipate one-word landmarks as well as landmarks containing two or more words.

Let's say, for example, that the user takes a picture of the Lechmere subway station sign and dictates, 'Get off at Lechmere station." Since the speech recognize r does not have 'Lechmere" in its vocabulary database, the recognizer interprets the user's dictation as 'Get off at leach mirror station." Luckily, one of the direction patterns that we gathered from analyzing examples of route directions is the pattern, 'Get off at <landmark> station." In LISP code, this pattern is translated as 'Get off at (?* x) station." Given a set of speech recognition results, the LISP program will run through a list of patterns, translated in a similar fashion, and compare the results with each pattern until it finds a match. When we compare 'Get off at leach mirror station" with 'Get off at (?* x) station," we can assign 'leach mirror" to the value x, thus successfully extracting the landmark of the sentence.

Natural Language Processing Alternatives

The language-processing community has two philosophies to understanding speech. One philosophy, which our simple pattern matching approach is based on, holds that statistical analysis—matching words or phrases against a historical database of speech examples—is the most efficient tool for guessing a sentence's meaning.

The other philosophy tries to analyze a sentence through its grammatical structure. An alternative way to extract the landmark of a sentence may be to parse the speech results with a grammar-based natural language system, such as a part-of-speech tagger (Brill, 1995) or a syntactic parser (Sleator & Temperley, 1993). A grammar-based system would break a sentence into its syntactic components (e.g., subject, verb, and object) and arrange these components into treelike diagrams that represent a sentence's semantic content (e.g., who did what to whom) in order to isolate the object of the sentence. Since the landmark of a direction sentence is generally the object of the sentence, we can extract the landmark by identifying the object of the sentence.

For this thesis, however, we decided to stick to the first language-processing philosophy of statistical analysis. Instead of employing a grammar-based system, we match the speech results against our own database of common direction patterns.

Comparison of OCR and Speech Results

Once we extracted the landmark from each speech result, we compared the landmark to the OCR results. Currently, the comparison is a straight matching of the word or words. Since landmarks tend to be proper nouns with uncommon pronunciations that are unlikely to be contained in the speech recognizer's vocabulary database, the speech recognizer is likely to misinterpret the user's dictation of the la ndmark. On the other hand, since the user is likely to capture the landmark on the camera, the OCR results will probably contain the correct spelling of the landmark. Therefore, we can simply replace the landmark from the speech result with the OCR results.

In the example illustrated in the previous section, the LISP program successfully extracted the phrase "leach mirror" from the speech results. In that example, the user took a photograph similar to the image show in Fig. 7.

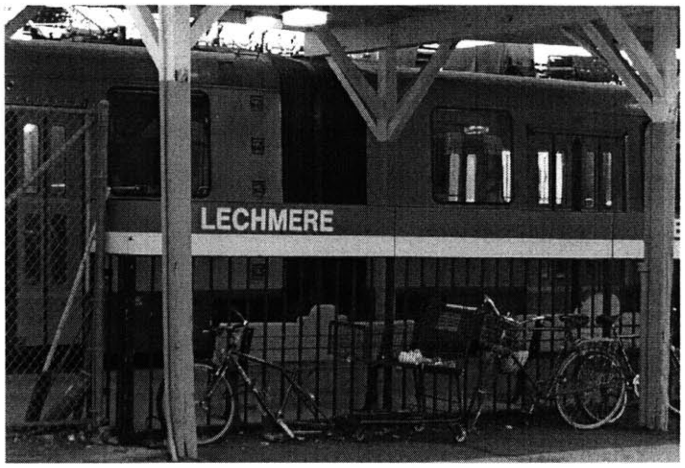


Fig. 7. Image taken of the Lechmere subway station sign.

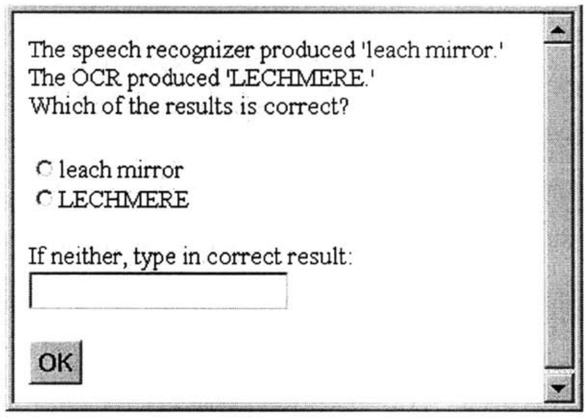
Whereas the speech recognizer translated 'Lechmere" into 'leach mirror," the OCR successfully recognized the characters on the subway sign as 'LECHMERE." Therefore, MARCO can simply substitute 'LECHMERE" in place of 'lea ch mirror" in the speech results. A summary of the results is shown below:

Narration	Get off at Lechmere station
Speech recognition	Get off at leach mirror station
OCR	LECHMERE
Combined recognition	Get off at LECHMERE station

This method of selectively replacing the landmark from the speech results with the OCR results, allows us to efficiently disambiguate the two recognition results without any complicated algorithms or heuristics. In the future, we can run both sets of syllables through speech synthesizers and analyze their waveforms to determine similarities. Although the speech recognizer is likely to misinterpret the words of the landmark, one thing that the recognizer generally interprets correctly is the number of syllables, so we can parse the landmark into syllables and compare those syllables to the syllables from the OCR results.

User Intervention Dialog Box

If the comparison is unsuccessful, we bring the results to the user and ask the user to select the correct solution. As shown in Fig. 8, if the speech recognition results contain the words "leach mirror" and the OCR results contain the word "Lechmere," we bring up a pop-up screen that says, "The speech recognizer produced 'leach mi rror' and the OCR produced 'Lechmere.' Which of th e results is correct?" The user can then select the buttons next to the correct result or type in a different word. Even if we must ask for user intervention many times, MARCO will still provide a significant reduction in user overhead for time.



The speech recognizer produced 'leach mirror.'
The OCR produced 'LECHMERE.'
Which of the results is correct?

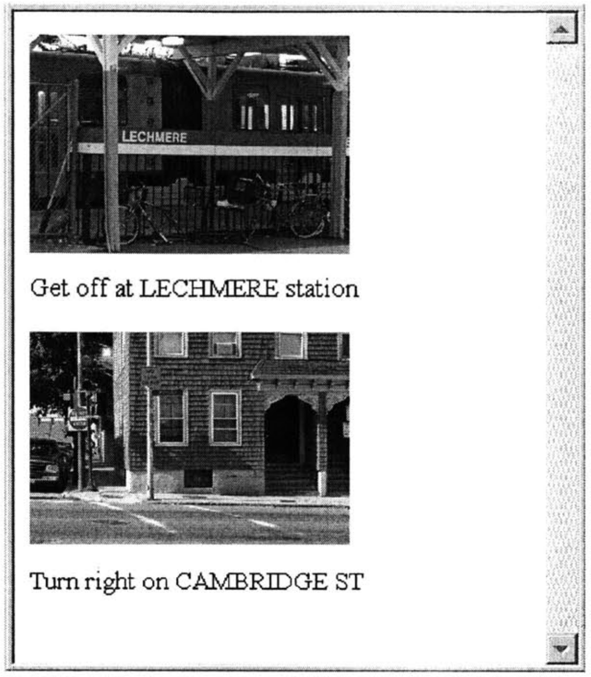
○ leach mirror
○ LECHMERE

If neither, type in correct result:

Fig. 8. Screen Asking the User to Disambiguate Results.

Presentation of Final Results

After successfully disambiguating the two recognition results of one image-dictation pair, MARCO stores the completed direction and continues to recognize and disambiguate the rest of the directions. Once it is finished disambiguating the rest of the directions, MARCO compiles the set of coherent, readable directions into a web page containing each image, followed by the disambiguated recognition results. Fig. 9 shows an example of the output produced by MARCO.



Get off at LECHMERE station

Turn right on CAMBRIDGE ST

Fig. 9. Example Output Produced by MARCO.

Now, the user can transmit the web page to other users or easily post the directions on the Internet. MARCO is also ready, at this point, to start generating a new set of route directions.

In the next chapter, we describe how we used a software agent to streamline the data capture, recognition processing, mutual disambiguation, and presentation into one, standalone application.

6 Stitching Together the Applications

As seen in chapters 4 and 5, MARCO uses several commercial software applications (Photoshop, ViaVoice, OmniPage) and a myriad of LISP programs to collect, combine, and present the recognition results of the audio and image files. In order to automate the entire recognition process and to provide a single, umbrella application for the user, MARCO needs a way to streamline the processes and tie together the various applications. This is where the concept of a software agent came into play.

6.1 MARCO's Software Agent

As described in section 2.2, instead of interfacing with the APIs of the individual recognition systems, we chose to utilize a software agent to interact with the commercial applications. A software agent is a piece of software, which acts to accomplish tasks on behalf of its user. Many agents are based on the idea that the user need only specify a high-level goal instead of issuing explicit instructions, leaving the "how" and "when" decisions to the agent. The MARCO software agent uses a scripting language to mimic a human user's interaction with the recognition systems and to integrate the applications together at a higher level.

Applescript: the Thread that Weaves Together the Processes

The scripting language that we used to stitch together the various applications was Applescript, a language used to control applications on the Macintosh OS. When saved as a droplet application and protected by thorough error checking, an Applescript program can run a series of tasks on multiple files within multiple folders. Given one or more folders of JPG files and user dictation through a microphone attached to the desktop, the MARCO agent is able to "clean" the image files, run the files through OCR, process the user dictation with speech recognition, and display the results of each recognition in a web page. It takes, on average, about three minutes for MARCO to collect the inputs and display the results for two images; however, we are constantly refining the automation and cutting down the processing time. The following Applescript code is a snippet of the program used to alter a JPEG file and run it through the OCR:

```
tell application "Photoshop"
open jpg_file
do script "transform"
end tell
...
tell application "OmniPage Pro 8.0"
...
load and OCR pct_file
end tell
```

The main problem encountered when combining two unrelated recognition systems is that not only do we have to deal with the problems of each individual system, but we also have to deal with the difficult interaction between the two systems. Applescript, while generally successful at interfacing with the various applications, did have a few obstacles. Such obstacles are also common in other scripting languages, e.g., Visual Basic and TCL. Applescript, much like the applications that it weaves together, is not designed to be used for the task of combining recognition technologies. Since Applescript is not designed to program software agents, there are some applications (e.g., the voice recognition technology IBM ViaVoice) that are not scriptable.

QuicKeys and Prefab Player: the Scotch Tape When the Thread Breaks

To overcome the limitations of Applescript in simulating user-computer interaction, we often have to resort to workarounds with the help of third party programs like QuicKeys and Prefab Player. Both programs provide a way to automate user interaction with the computer, by recording user input and computer keyboard or mouse movements.

QuicKeys is useful for displaying instruction dialogs and interfacing with secondary applets. QuicKeys cannot, however, interface with droplets since droplets are initiated by dropping files or folders on the icon.

Prefab Player allows the computer to mimic a user by navigating menus and typing commands. For example, in order to save the dictation from the speech recognizer, we used the following code:

```
tell application "IBM SpeakPad" to activate
tell application "PreFab Player"
do menu menu item "Save" of menu "File"
type "voice-output"
click button 1
end tell
```

After "sewing" and "taping" together the ap plications and programs for MARCO, we now have a massive collection of software that resembles a Frankenstein for the digital world. Designing MARCO in this fashion brings up many user interface issues. The next section covers some of the UI issues inherent in choosing to employ a software agent.

6.2 User Interface Issues

Using a scripting language and user interface (UI) agents to stitch together applications, without relying on an API, raises many UI issues: buttons are clicked and menus are opened automatically like a self-playing piano, cursor control is taken away from the users at arbitrary times, and the users must be careful not to interfere with the application for fear of creating a runtime error. There are five issues commonly identified when designing software agent user interfaces: understanding, trust, control, distraction, and personification (Maes & Wexelblat, 1997). The following sections deal with each of these issues in the context of the MARCO system.

Understanding

The first issue is understanding. How does the agent know what to do, and how does the user know what the agent is capable of doing?

Since we want the MARCO agent to operate the applications for the user, the user must be able to give the agent information pertaining to the desired result. We already designed the MARCO system with the goal of gathering, combining, and displaying recognition results for audio and image files. However, the user needs to feed the audio and the images to the agent, and the user must let the agent know when he is done dictating the directions so that the agent can proceed with the recognition process. The latter is accomplished via QuicKeys commands, which the user selects from a menu to indicate when he is finished dictating the directions. The former is accomplished at the start of MARCO, when the agent asks the user to provide a folder containing the photographs to be recognized.

Now that the agent knows what the user wants to do, the agent needs to tell the user what functions it is capable of performing. MARCO addresses this situation by showing "tips" when it is about to perform a particular function or when it requires some action or input from the user. These tips come in the form of modal QuicKeys dialog boxes, which the user must click away to indicate that he understands the request.

Thus, the MARCO agent and the user are able to understand each other through QuicKeys and information dialog boxes. These information "tips" and user directions help the agent and the user update each other.

Trust

The second UI issue is trust. To what degree can the user trust that the MARCO agent can perform its task relatively independently without repeated supervision? If the agent does not perform a task in a fashion similar to the user and the results of the task are unsatisfactory, then the user loses trust in the agent.

Our approach to this problem is to have the agent suggest what it would do and allow the user to approve or disapprove the suggestion before any action is taken. For example, after the agent uses the mutual disambiguation algorithm to combine the speech and OCR results, it displays a dialog box with the original recognition results and the suggested combination. The user is given the option to either accept the suggestion or override it and type in the correct combination.

Even though adding the extra step of asking for the user's approval may decrease the efficiency of the MARCO system, the extra effort provides a way for the agent to establish trust with the user. If the user sees in advance what the agent plans on doing, then he is more likely to trust that the agent will do the right thing. In addition, this extra step allows the user to catch any mistakes that were not corrected through the mutual disambiguation algorithm or through the numerous file enhancements described in chapter 4.

Control

Control is the biggest issue for the MARCO agent. Since the software agent serves as an intermediary between the user and the applications, the user interacts solely with the agent. The agent serves as the interface, translating the user's input to the programs. One of the advantages of using a software agent to communicate with the underlying applications is that the user can delegate the mundane procedures required to process the audio and image files to the agent. On the flip side, the user must relinquish control of the application and allow the agent to operate with some degree of autonomy.

This problem is inherent when using any piece of software. Users must give up some control; what is important is that control not be taken from users involuntarily. This is where the MARCO user interface needs improvement. Currently, keyboard and mouse control are taken away from the user at arbitrary times, random buttons and menus are clicked automatically, and the user must refrain from interfering with the application unless asked to by the agent. This approach can have problems because the user is unable to bypass the agent, causing the user to feel out of control.

In any complex task, there are inevitably points when key decisions are made or when alternatives must be analyzed. One way to help address the control issue is to allow the software agent to execute a task until one of these key points is reached. At these points, the MARCO agent uses update boxes to bring the user back into the loop. In the current MARCO system, the user can suspend the agent's process at these key junctures but not at any other time. In the future, however, a more sophisticated user interface would allow the user to suspend the agent's process at random times via 'Pause" and 'Resume" buttons.

Distraction

As shown in the previous three sections, the MARCO software agent handles the issues of understanding, trust, and control by providing informational messages or dialog boxes. As a consequence, distraction becomes an issue. Since the agent needs to initiate communication with the user, it is necessary to change the user interface visually to attract the user's attention, which ends up disrupting th e user's normal work pattern.

There are several ways to deal with the distraction problem. We can find a less obtrusive means of notifying the user, we can differentiate between important messages and less important messages, we can have the user control the level of interaction, or we can adapt the level of communication to the importance of the current operation (Maes & Wexelblat, 1997). These solutions are not applicable to MARCO since most of MARCO's communications with the user require a response fro m the user before the agent can proceed with its task. All of the messages are essentially important and there is no way to avoid distracting the user with them.

This may not be a problem for this application, however, because the user is unlikely to be engaged in other tasks simultaneously. Since the user is actively walking through the desired route, MARCO requires the user to annotate the captured images by dictating directions into the microphone. Thus, even though we risk distracting the user through pop-up windows, the user of the MARCO system is unlikely to have something to be distracted from.

Personification

The final UI issue is personification, or representing the agent as a separate entity to the user. Our current system does not use graphics or intentional language to give the agent an appearance of a recognizable entity. These tactics may be employed in the future to encourage the user to interact with the agent in a more delegative style; however, we do not feel that this is a big issue in our particular scenario.

Therefore, despite the five UI issues described above, MARCO saves the user from the mundane and tedious task of manipulating and transferring audio and image files from application to application. MARCO also reduces the overhead time and labor required if a user were to juggle the applications himself. These reductions are illustrated in the following chapter, which details the methods and results of our evaluation tests.

7 Evaluation

7.1 MARCO v. Alternative Map Services

Methodology

To evaluate the effectiveness of MARCO directions in relation to map service alternatives such as Mapquest, we conducted several usability and accuracy tests. The first set of usability tests were aimed at determining whether MARCO routes are more effective for users than pure street-connectivity routes such as the ones produced by Mapquest. The usability tests consisted of giving the users maps—created from either MARCO or Mapquest—to various locations around Cambridge, MA, and asking them to record the time it took them to walk according to the directions. First, we used MARCO to create two different routes around Cambridge. Then we mapped the same routes with Mapquest.

Results

The following numbers are the results of the evaluation tests. Each route is followed by the average number of minutes required to travel from the starting point to the destination:

MARCO Route 1	**16.25 minutes**
Mapquest Route 1	**47.5 minutes**
% Increase in duration	**192.3%**
MARCO Route 2	**14.75 minutes**
Mapquest Route 2	**17.5 minutes**

% Increase in duration	**18.6%**

First note that the MARCO directions significantly reduced travel time for Route 1. This reduction occurred because the Mapquest version assumed that the user was driving to the destination, while the MARCO version allowed the user to use public transportation. Thus Mapquest gave the user one incorrect direction, which led to a 192% increase in travel time. MARCO was able to take advantage of user context information and present directions that cater to a pedestrian.

The MARCO version of Route 2 also improved the travel time, though to a lesser extent. This is because it did not have the advantage of public transportation. Instead, the MARCO directions were able to use many landmark descriptions to facilitate navigation through various buildings and shortcuts. The results of Route 2 illustrate the ability of MARCO to utilize key landmarks, which are absent in maps generated by Mapquest.

The results of the evaluation tests were very encouraging. We showed that MARCO offers faster alternatives for map users when the user travels on foot and via public transportation. In most cases, there was a modest improvement. However, there may be times when travel time is significantly reduced.

7.2 Accuracy of Multimodal Disambiguation Algorithm

Methodology

The second test measures the accuracy of the multimodal disambiguation algorithm described in chapter 5. There are two components of this evaluation test. First, we calculate the percent coverage of the speech act templates, using the routes generated during the feasibility tests. Second, we calculate the percent accuracy of the recognition results after we combine them using the simple disambiguation algorithm.

Results

After analyzing examples of route directions, taken from the Internet or from people's emails of directions to parties, we were able to compile a total of 183 common direction patterns. When compared to 3 routes and 17 direction steps, a subset of the directions used during the feasibility tests described in chapter 3, we found that 13 direction steps (or 76.5% of the directions) in the feasibility tests were covered by the database of direction patterns.

When the simple mutual disambiguation algorithm was tested on the 17 direction steps, we obtained the following results:

Speech reco word-level accuracy:	76.3%
Speech reco landmark accuracy:	62.2%
OCR word-level accuracy:	70.6%
OCR landmark accuracy:	70.6%
Combined word-level accuracy:	87.3%
Combined landmark accuracy:	78.4%
% benefited by mutual disambiguation:	76.5%

You will notice that these percentages vary from the statistics collected during the feasibility tests in chapter 3. These variations occur because we only tested a subset of the original routes and we implemented many features of the MARCO system after we conducted the initial feasibility tests. One of the additional features is asking the user to crop the images prior to running them through the OCR, which led to equivalent word-level and landmark accuracies for the OCR. Even though these percentages vary from our earlier tests, the conclusions that can be drawn from these statistics are consistent with the hypotheses proposed after looking at the feasibility tests.

First, the landmark accuracy is still worse than the word-level accuracy for the speech recognition. As predicted, landmarks tend to be proper names easily missed by the speech recognition. Second, the combined landmark accuracy is significantly higher than the speech landmark accuracy, and the combined word-level accuracy is higher than the word-level accuracies for the speech recognition and the OCR. This is because the landmarks most likely to be misinterpreted by the speech recognition appeared on street signs and store entrances. Thus, we were able to substitute the landmarks in the speech results with the results from the character recognition. Third, as noted in chapter 3, it is not necessary to achieve 100% landmark accuracy in order for MARCO to be considered successful. After we combine the speech recognition results with the OCR results, MARCO queries the user about possible ambiguities. A reasonably high landmark recognition rate (78.4%) still saves labor for the user. This reduction in labor is explored in the third evaluation test, described in the next section. Finally, 76.5% of the direction steps followed one of the patterns in our list of common direction patterns. The few that did not fall under one of our patterns were particular to their situations and probably will not appear again in the future. However, we could implement an additional feature that allows MARCO to 'learn" new patterns with the user's help so that they can be used in future iterations of MARCO.

These results are very encouraging. We have shown that through our simple mutual disambiguation algorithm, we were able to successfully combine most direction steps. In every case, we were able to produce a high enough landmark accuracy so as to reduce the user's labor when they are asked to resolve ambiguities.

7.3 Reduction of User's Labor

Methodology

The third test examines whether MARCO reduces the overall labor of the user. Even though the 78.4% combined landmark recognition rate found during the algorithm accuracy test (Section 7.2) is not as high as the 94.7% upper-bound that we predicted during the recognition feasibility test (Section 3.2), the reasonably high landmark recognition rate still saves labor for the user. MARCO is able to mutually disambiguate 76.5% of the directions (the percentage of directions covered by the patterns used in our mutual disambiguation algorithm). Thus, the user only needs to manually disambiguate 23.5% of the cases. This means we can ask the user to verify all of the automatically combined results in one step by displaying those results in a single dialog box. The remaining 23.5% of the directions can be brought to the user for manual disambiguation. This simple act of combining 76.5% user prompts into a single prompt significantly reduces the number of interactions between the user and MARCO.

To test the reduction in user-computer interactions, we compared the number of user-computer interactions in MARCO to the number of user-computer interactions in Microsoft Word. (We found inserting image files and typing directions into a Word document to be the best alternative for generating a list of directions from a set of images and audio files.) The next section lists and totals the number of user-computer interactions required to generate a list of directions in MARCO and Microsoft Word.

Results

First, we analyzed the user-computer interactions required by Microsoft Word. If a user has a folder of digital images and a folder of audio files containing directions that correspond with the images, he could generate a list of image-annotated directions by inserting the images one-by-one into a Word document. Then, he could play the audio files and transcribe the recording into the document after the appropriate images. Inserting the images takes three user-computer interactions per image, while transcribing the audio files takes two user-computer interactions per file. Users often make mistakes while typing on the computer, however, so the user would have to run Microsoft Word's 'Spelling and Grammar" tool to catch his typing mi stakes. This amounts to at least two user-computer interactions. The worst-case scenario is when the user misspells every single word and has multiple grammar errors. Therefore, using Microsoft Word to generate the list of directions requires about 6 user-computer interactions per direction.

Next, we analyzed the user-computer interactions required by MARCO. The user must narrate the directions for each image file and notify MARCO when he is done dictating. This requires two user-computer interactions per file. After the user is done dictating directions for all of the files, MARCO processes the speech and character recognition for each file. If MARCO is unable to automatically disambiguate the recognition results (i.e., the speech recognition results do not match any of the patterns in the mutual disambiguation algorithm), then MARCO displays the results from the speech recognition and the OCR and prompts the user to manually disambiguate the results. Otherwise, any results that are automatically disambiguated are displayed together in a single dialog box, and the user can verify these results in one step. Since 76.5% of the directions are covered by the patterns used in our mutual disambiguation algorithm, the verification process requires one user-interaction per direction for 23.5% of the directions and one user-interaction for the remaining directions. Thus, using MARCO to generate the list of directions requires a little above two user-computer interactions per direction.

These results prove our hypothesis that MARCO reduces the number of user-computer interaction and, hence, the amount of user labor required to generate a list of directions from a set of digital images and audio files. If the user uses MARCO, he interacts with the computer one-third as much as he would if he uses Microsoft Word. Therefore, even though the combined landmark recognition rate found in Section 7.2 is not as high as the upper-bound predicted during Section 3.2, MARCO still saves labor for the user.

Now that we have evaluated the effectiveness of the current MARCO system on a desktop computer, the next chapter will explore the feasibility of porting the MARCO system to a handheld computer.

8 Can MARCO Be Extended to the Handheld Computer?

Our current application is centered around batch post processing on a desktop computer. For example, MARCO requires the user to redictate the narration direction into the speech recognition system after the images are recognized. In the future, however, we imagine the user would have a handheld computer containing a color digital camera. The user would only need to dictate the narration once, right after or before he takes a snapshot of the landmark.

MARCO would start by isolating the words the viewer wants to understand. The user selects the part of the image containing the words. Then the images and audio clips would either be processed immediately and serially on the handheld or be compressed and transmitted to a server wirelessly over the cellular network. Most of the work would be done by transferring the processing jobs to a mainframe, thus moving the heavy lifting to the server. The server would perform the complex image and voice recognition processing and send the results back to be displayed on the handheld.

The next three sections summarize the work we have done to integrate the MARCO system with a handheld Pocket PC device, the recognition accuracy of the images and the audio recordings captured by the handheld device, and the future directions for this extension of the MARCO system.

8.1 Completed Research

Handheld Device

To assess the feasibility of extending the MARCO system to a handheld computer, we experimented with audio and visual capture on a Toshiba E750 Pocket PC.

This particular model was chosen because its built-in wireless and microphone functionality enable it to take voice input and communicate with the desktop. The Toshiba also has a Compact Flash (CF) card slot that can be connected to a Lifeview FlyCAM-CF camera for video and photo capture abilities.

Existing Software Applications

The Toshiba Pocket PC 2002 operating system comes with three built-in or downloadable voice recording and image capture software programs. The first program is the built-in Voice Recorder program that operates in conjunction with the Notes application. This simple program saves the user's voice input as a wave (WAV) sound file that can be transmitted to other handheld computers and desktop computers.

The second program is Quicktest, a software bundled with the FlyCAM camera that facilitates photo capture up to resolutions of 640 x 480. The captured graphics files are stored as bitmap (BMP) files, which can be transmitted along with the audio recording to other computers. The Quicktest application is written in Embedded Visual C++ and has documentation outlining its functions.

The final program is Microsoft Portrait, a Microsoft Research application that enables voice and video conferencing between a Pocket PC and a host desktop computer. Utilizing the wireless capabilities of the Pocket PC, a connection is made with the IP address of the desktop computer. This software is still under development and there are several instances when the Pocket PC device had to be restarted to ensure smooth functionality of the Portrait application.

Software Integration Issues

We analyzed each software program to determine if the Toshiba E750 is capable of recording sound files, capturing digital photos, and wirelessly communicating with a desktop computer.

Using Embedded Visual Basic and the Windows Voice Recorder control, we successfully developed a voice recording application. This application performs the same functions as the built-in Voice Recorder program and it can be integrated with other programs written in Embedded Visual Basic. However, we were not able to create an Embedded Visual Basic program to control image capture and Quicktest is implemented in Embedded Visual C++.

Therefore, we decided to investigate the Microsoft Portrait application because it already bundles the voice recording and the photo capture functions into one application that can communicate wirelessly with a desktop computer. If we can harness the capabilities of the Portrait application, we would not need to program in an embedded language.

However, there are some immediate integration difficulties since Microsoft Portrait is currently only available on the Windows platform and the application does not facilitate automatic file retrieval or audio recording downloads. The photos taken through Microsoft Portrait were also of lower quality than those taken through Quicktest. In theory, we could alter the Portrait application to suit our purposes; however, the code for the application may be too complicated and too difficult to obtain.

Section 8.3 goes over future work in integrating the MARCO system on a handheld computer. In the meantime, we decided to test the quality of the digital images and voice recordings captured with the Toshiba handheld to determine if those inputs can be recognized by current recognition technology. The next section gives the results of our feasibility experiment.

8.2 Recognition Accuracy

To determine the quality of the images and audio files captured by the Toshiba, we performed a feasibility test similar to the one conducted in Chapter 3. We walk along a short route, dictating directions and taking pictures of key landmark signs along the path. We used Quicktest and Voice Recorder to capture the images and the audio recording, respectively. Then, we ran the files through MARCO's voice recognizer and character recognizer and calculated the error rates of the recognition.

Our feasibility study shows above 68% landmark recognition accuracy for the voice recognizer and 55% landmark recognition accuracy for the character recognizer.

We analyzed one route and six direction steps. In the numbers below, we present the raw word-level accuracy rates of the speech recognition and the landmark accuracy of both the speech recognition and the character recognition.

Speech reco word-level accuracy:	72.5%
Speech reco landmark accuracy:	68.4%
OCR landmark accuracy:	55%

These results are pretty consistent with the results we found during our initial feasibility tests. Though the OCR landmark percentage (55%) is not as high as the percentage found during the accuracy tests in Section 7.2 (70.6%), the speech recognition percentages are pretty high. One reason for the lower OCR landmark accuracy rate is the 640 x 480 pixel resolution of the Quicktest images, compared to the 720 x 480 pixel resolution produced by the Kodak DC 290 digital camera.

8.3 Scope for Future Work

The results of our handheld research demonstrate that there are many possibilities for further development of the MARCO system on the handheld computer. Although a fully functional application could not be developed in Embedded Visual Basic, Embedded Visual C++ has been shown to be the easiest solution for interfacing with the FlyCAM CF camera. The first stage would be to develop an application in Embedded Visual C++ to incorporate both the camera and voice recorder functions. This could be done by modifying the existing Quicktest code and then extended to take advantage of the wireless capabilities of the Pocket PC. An alternative would be to alter the Portrait application so that we can automatically accept image file transfers and download audio recording.

The original goal of this feasibility experiment was to develop an application, which integrated the functions of voice recording and photo capture into one user-friendly application. The files outputted from the application would then be transmitted wirelessly to a back-end server, which would do the processing-intensive recognition procedures in MARCO. Eventually, as the processing capabilities of handheld computers continue to improve, we could eliminate the back-end server and perform all of the recognition locally on the handheld computer. Ultimately, MARCO could be developed completely on the Pocket PC device. The 400Mhz XScale processor should be sufficient for both voice recognition and OCR technologies. Current software, such as IBM's Embedded ViaVoice Mobility Suite, can incorporate these technologies and should be investigated in the future.

Now that we have established the feasibility of extending the current MARCO system to a handheld computer, the next chapter will attempt to put into perspective the overall approach of MARCO with a discussion of other applications that have also attempted to incorporate multimodal input systems.

9 Related Works

This thesis is related to mutual disambiguation of multimodal systems, the study of methods of conveying routes, the advances in voice recognition and OCR on handheld computers, and current multimodal map applications. The urban navigation assistant application, and use of a combination of voice recognition and OCR in such an application by nonprofessional users, we believe to be unique.

9.1 Mutual Disambiguation of Multimodal Systems

Current powerful computers and better understanding of human-computer interaction have opened the doors to utilizing the flexibility and reliability of multimodal user interfaces (Raisamo, 1999). Each modality has its own strengths and weaknesses. The challenge is to identify when to use one modality over the other so that we can mutually disambiguate all of the results.

Sharon Oviatt's study of multimodal systems is a good modern survey of the mutual disambiguation of multimodal systems (Oviatt, 1999). Contrary to the common belief that a multimodal system incorporating two error-prone recognition technologies will compound errors and yield even greater unreliability, Oviatt concludes that a multimodal architecture fuses two or more input modes to permit the strengths of each mode to overcome weaknesses in the others (Oviatt, 2000). This "mutual compensation" of recognition errors results in a system that is more reliable than the individual systems (Abowd et. al., 2000). A flexible multimodal interface also allows users to interact effectively across multiple tasks and environments, especially mobile environments (Oviatt, 2000).

9.2 Conveying Routes through Descriptions and Depictions

Barbara Tversky and Paul Lee studied the use of pictorial and verbal tools for conveying routes (Lee & Tversky, 1999). They concluded that the existence of parallel depictions and descriptions for routes does not mean that both are equally effective in all situations. In many cases, a combination of the two is the most effective in portraying routes; these cases are able to simultaneously utilize the advantages of both methods. Descriptions are more appropriate for abstract information ('turn right'), whereas depictions are more appropriate for information that is directly or metaphorically visualizable ('big oak tree') .

9.3 Voice Recognition and OCR on Handheld Computers

There have been substantial projects in the research area that links the gathering and capturing of images with powerful, pocket-sized computers. These projects, in addition to information annotation devices, are looking at the intersection between powerful mobile computing devices and imaging systems. One such device is Ismail Haritaoglu's InfoScope, an automatic sign translation device for foreign travelers (Haritaoglu, 2001). The user carries around a color camera attached to a PDA with wireless modem connection. When the user encounters a sign with foreign text, he takes a snapshot of the sign. The picture is then displayed on the PDA, where the user selects a sub-portion of the image and sends the image to a server via wireless communication. The server does the compute-intensive image processing (segmentation and OCR) and translating, and sends the translated text back to the client where it is displayed in the same place where the original text was written.

While Haritaoglu and others were perfecting the combination of handheld computers and image OCR, researchers at IBM were working with Compaq to create one of the first commercially available handheld computers that is accessible via human speech. The successful deployment of Compaq's Pocket PC H3800 Series proved that current handheld computers have the processing power to handle voice recognition. In addition to processing power, handheld computer programmers must deal with the issue of performance degradation now that the speech recognition applications are mobile. Researchers at Oregon Graduate Institute (Oviatt, 2000) and Carnegie Mellon (Huerta, 2000) are attempting to address the degradation issue through multimodal input and speech coding algorithms.

9.4 Current Multimodal Map Applications

Designing software for handheld computers is difficult because of the limited computing power, memory, display size, and the choices of hardware and operating systems. Designing handheld applications that must be network-based, is even more daunting. Research done at the Center for Human-Computer Communication in the Oregon Graduate Institute, however, shows that some of the challenges can be addressed by designing these applications as multimodal, software agent systems (McGee & Cohen, 1998).

A slew of multimodal handheld systems featuring voice input and gesture input have been designed in the past few years. EURESCOM has produced several pen-voice applications (Cheyer & Julia, 1999). Most noteworthy is the project MUST (Almeida et. al., 2002), which uses pen and speech input to help a tourist navigate around Paris. Philip Cohen and researchers at the Oregon Graduate Institute designed QuickSet—a pen/voice multimodal interface that aids the military in creating and navigating maps (Cohen et. al., 2000). AT&T's MATCH provides a multimodal interface for mobile information access with spoken and graphical interaction (AT&T Research Labs). Other multimodal speech and pen-based gesture interfaces have been designed by OGI, IBM, Boeing, NCR, and BBN (Oviatt & Cohen et. al., 2000).

Each of these map applications try to mimic what Sharon Oviatt terms as 'natural multimodal communication" (Oviatt & Cohen, 2000). Instead of asking the user to enter information via keyboard and mouse clicks, these applications collect the user input through more natural modes of communication. The success of these speech/gesture map applications reinforces the fact that multimodal systems create more flexible and efficient ways of gathering user input. However, this particular combination of inputs is not applicable to the task of generating new directions. While speech and gesture inputs are natural ways for a user to communicate where he wants to travel within an existing map, we explored the combination of speech and images inputs for the MARCO system.

This chapter has hopefully helped to place MARCO and the mutual disambiguation process into the context of other research efforts in these areas. In the next chapter, we bring the big picture back to MARCO to discuss the contributions made by this thesis work and to suggest future work needed.

10 Conclusions

10.1 Summary of Contributions

In this thesis, we investigated three broad topics—creating a system for generating directions to a location in a city; combining different modalities from various recognition systems with user-context to provide more robust results and services; and using a scripting language to "stitch" together off -the-shelf applications. The MARCO navigational assistant provided a preliminary application platform from which approaches to these properties were tested. What resulted was a software agent that integrates off-the-shelf speech and character recognition applications via scripting languages to assist users in giving directions for urban navigation.

Though we encountered certain deficiencies in each individual commercial recognition technology, we were able to 'tweak" the input and application settings and effectively demonstrate how piecing together partial results obtained from each mode of recognition can derive more reliable results. We successfully reduced landmark recognition errors through a relatively simple mutual disambiguation algorithm and showed that it is possible to provide reliable results through two unreliable recognition technologies by taking advantage of the user context and of mutual disambiguation. By automatically disambiguating a large percentage of the recognition results, MARCO significantly reduces the number of user-computer interactions and decreases the overall labor required by the user.

In this thesis, we also determined that current handheld computers are capable of retrieving images and audio for accurate character recognition and speech recognition. This conclusion ensures that the MARCO software can be ported to a handheld computer and still produce reliable recognition. Once the system is on a handheld, we would have a mobile device that allows users to store trip directions locally so that they can retrieve and rely on these directions during the trip. Information overlays on a PDA would be a lot more practical. We could also integrate information from a map or other database onto the PDA display.

10.2 Future Work

Additional Sources of Information

The existing system uses two sources of information to determine the correct set of directions. However, there is nothing stopping us from introducing a third or fourth source of information. Assuming that the mutual disambiguation theory is correct, introducing additional sources of information may even further improve the final result set. This third source of information can verify suggested results and serve as a tiebreaker when the other two sources disagree. Examples of additional sources of navigational information include directories and yellow pages or GPS locations.

Mapquest and Yahoo Yellow Pages and other similar services contain databases of Boston and Cambridge street names and store names. During the disambiguation process, we could try to match the voice result and the character result to streets or stores in the existing database. While matching the results, we could assign a confidence percentage to each word and choose the word with the highest percentage. If none of the confidence percentages are significantly high, then either the recognition was way off (i.e., the image was too distorted or the user narrated the directions in an especially noisy environment) or the phrase we are trying to recognize does not exist in the database. The benefit of keeping a database of common landmarks is that the system can 'learn" new phrases with each iteration. Since the user is given a chance to correct the suggested results, any landmarks that were not found in the database can be added to the database after the user verifies the spelling of the landmark. This assumes, of course, that the user checks the spelling of the landmark and that the user knows the correct spelling of the landmark.

We could also use GPS to obtain precise location data for the exact point where a photograph is taken. Unlike the street database scenario described in the previous paragraph, using GPS can significantly reduce the number of items in the search set by restricting the landmarks to ones that are within a few feet of the point where the user takes a photograph. From there, we could do the same sort of matching as in the street database case.

Smart Direction Guide

The current MARCO directions come with just line-by-line instructions about how to get from one location to another. What if MARCO was able to incorporate common sense reasoning about the destination or purpose of a trip? The process of reasoning about things using knowledge and skills that most people share (Minsky, 2003), allows us to make new statements that were not mentioned explicitly in the original statements. By using common sense reasoning, MARCO could serve as a smart direction guide and provide the user with additional information that might be helpful during his trip (Xian, 2002).

The annotated information supplied by the smart direction guide saves the user time and effort in searching for the information, when he might be busy driving or walking. What might be useful information to the user during his trip? The purpose of the trip could be inferred from the set of directions (in particular, the destination) for the trip. For instance, if a direction set describes traveling from home to the airport, then one can conjecture that the mobile device user is probably about to catch an airplane or about to pick someone up from the airport. Further, we can also infer that real-time information on flight arrivals and departures might be useful for the user. The user can be informed of delays and adjust his travel plans prior to leaving for the airport.

Another user scenario could be that a user is traveling from home to a friend's house. The user's friend is throwing a party, but the user does not have time to search for possible gifts to bring over. In this case, the smart direction guide could look at each component of the direction set and suggest some gift stores or flower shops on the way to the destination. Given access to the local yellow pages or to some similar database, the smart direction guide would be able to provide useful recommendations.

Driving Directions

Current digital cameras are too slow to take action or moving shots; photographs taken while the user is on a moving vehicle—such as a car, a bus, or a subway train—result in blurred lines. Optimal image capture occurs when the user is on foot, and photographs associated with public transportation must be taken before or after the user boards the vehicle. Therefore, the current MARCO system is limited to walking directions or directions utilizing public transportation. Driving directions may be implemented in the future with the aid of high-speed digital cameras.

Direction-Taking Navigational Assistant

The navigational assistant discussed in this thesis allows users to demonstrate, or "give" directions. A further step would be to transmit the fina l compiled directions to a guest's computing device in electronic form. These directions could then be used by a sister direction-following assistant that could provide help of the form 'Is this is street where I'm supposed to make a left?".

This sister direction-taking assistant is also more practical than the direction-giving assistant since many applications of our research are for users who wish to follow a set of directions. The following chapter describes these applications and give examples of how our research can be used in those scenarios.

11 Future Applications

11.1 Navigating Large Buildings

The average American mall rarely contains the kind of navigation cues common in urban streetscapes, such as street signs and addresses. There are no prominent cues or signs to direct the shopper to a particular store. At the mall entrance, interested shoppers are presented with a basic map of the mall's layout as well as numbers for the numerous stores. However, the stores themselves do not display corresponding numbers, and most shoppers do not notice the simple maps given at the front of the mall. It has been suggested that this lack of navigation cues is deliberate, to promote wandering and store browsing. But for the shopper that wishes to target a specific store or a particular item within the mall, this convoluted maze can cause much unnecessary frustration. Many consumers are forsaking mall shopping in favor of on-line ordering, in no small part because mall navigation is difficult when they have targeted needs. MARCO's sister direction-taking device would be especially helpful in this situation. The system can be easily modified to adapt to routes within a large building. A shopper would have a complete list of stores on the handheld and could then navigate the mall with the click of a button. The same system can be applied to office buildings or hospitals.

11.2 Helping the Elderly

The interest to seniors is that such a system would make it easy to produce customized, step-by-step illustrated reminder systems for way finding, household procedures, medical procedures, etc. More generally, it opens up a path for usable speech recognition and visual recognition to make many kinds of household and medical interfaces easier to use. Since particular routines such as laundry or shopping are so specific to each person, and the need for specific kinds of reminders differs for many individuals, caregivers must often produce custom-designed procedural reminders. Sometimes it is this inability to perform everyday routines that provides the driving motivation for putting elderly people in nursing homes. Additional technological support might enable some elders to continue their independence for longer periods.

12 References

Abowd, Gregory D., Mankoff, Jennifer, & Hudson, Scott E. (2000). *OOPS: A Toolkit Supporting Mediation Techniques for Resolving Ambiguity in Recognition-Based Interfaces.* Computers and Graphics (Elsevier). Special Issue on Calligraphic Interfaces. 24(6), December, 2000. pp. 819-834.

Almeida, L., Amdal, I., Beires, N., Boualem, M., Boves, L., Os, E., Filoche, P., Gomes, R., Knudsen, J., Kvale, K., Rugelbak, J., Tallec, C., & Warakagoda, N. (2002). *Implementing and Evaluating a Multimodal Tourist Guide.* In Proc. International CLASS workshop on natural, intelligent and effective interaction in Multimodal dialog system, Copenhagen, Denmark.

AT&T Research Labs. *Multimodal Interfaces for Mobile Information Access.* MultimodalAccessToCityHelp project.

Brill, E. (1995). *Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging.* Computational Linguistics, 21 (4), 543-565.

Cheyer, Adam, & Julia, Luc. (1999). *Designing, Developing and Evaluating Multimodal Applications.* In WS on Pen/Voice Interfaces (CHI 99), Pittsburgh.

Cheyer, Adam, & Julia, Luc. (1998). *Multimodal Maps: An Agent-based Approach.* In book Multimodal Human-Computer Communication, Lecture Notes in Artificial Intelligence #1374, Bunt/Beun/Borghuis (Eds.), Springer, pp 111-121.

Cohen, Philip, McGee, David, & Clow, Josh. (2000). *The Efficiency of Multimodal Interaction for a Map-Based Task,* in the Proceedings of the Language Technology Joint Conference (ANLP-NAACL2000), Association for Computational Linguistics Press: Seattle, WA, Apr. 29-May 4, 2000, pp. 331-338.

Haritaoglu, Ismail. (2001). *InfoScope: Link from Real World to Digital Information Space,* Ubicomp 2001: 247-255.

Huerta, Juan. (2000). *Speech Recognition in Mobile Environments,* PhD thesis, Carnegie Mellon University.

Lee, Paul, & Tversky, Barbara. (1999). *Pictorial and Verbal Tools for Conveying Routes*, COSIT 1999: 51-64.

Lieberman, Henry. (1998). *Integrating User Interface Agents with Conventional Applications*, International Conference on Intelligent User Interfaces, San Francisco.

Lieberman, Henry. (2002). *Out of Many, One: Reliable Results from Unreliable Recognition*, in Proceedings of CHI ' 02 (Minneapolis MN)ACM Press.

Maes, Pattie & Wexelblat, Alan. (1997). *Issues for Software Agent UI.* Unpublished manuscript by the Software Agents Group, MIT Media Lab.

McGee, David, & Cohen, Philip. (1998). *Exploring Handheld, Agent-based Multimodal Collaboration.* In the Proceedings of the Workshop on Handheld Collaboration at the Conference on Computer Supported Cooperative Work (CSCW' 98): Seattle, WA.

Minsky, Marvin. (2003). *The Emotion Machine.* Chapter 6, Draft of Part I. http://web.media.mit.edu/~minsky/E6/eb6.html

Oviatt, Sharon, Cohen, Philip, et al. (2000). *Designing the User Interface for Multimodal Speech and Pen-Based Gesture Applications: State-of-the-Art Systems and Future Research Directions.* Human-Computer Interactions. Vol. 15, pp. 263-322.

Oviatt, Sharon. (2000). *Multimodal System Processing in Mobile Environments.* UIST 2000: 21-30.

Oviatt, Sharon. (1999). *Mutual Disambiguation of Recognition Errors in a Multimodal Architecture.* CHI 1999: 576-583.

Oviatt, Sharon, & Cohen, Philip. (2000). *Perceptual User Interfaces: Multimodal Interfaces that Process What Comes Naturally*, Communications of the ACM, v.43 n.3, p.45-53.

Oviatt, Sharon. (2000). *Taming Recognition Errors with a Multimodal Interface.* CACM 43(9): 45-51.

Raisamo, Roope. (1999). *Multimodal Human-Computer Interaction: A Constructive and Empirical Study*, Ph.D. dissertation, University of Tampere, Tampere.

Sleator, Daniel, & Temperley, Davy. (1993). *Parsing English with a Link Grammar.* In Third International Workshop on Parsing Technologies.

Xian, Angela Xiang. (2002). *A Smart Direction Guide that Provides Additional Destination-dependent Information.* Massachusetts Institute of Technology.

13 Acknowledgements

I want to thank the following people, without whom this work would not have been possible:

I want to thank my advisor, Henry Lieberman, for his persistence in helping me obtain the necessary resources for my project. I thank him for his guidance, flexibility, and encouragement for the thesis as well as for life outside of schoolwork.

I want to thank Nicholas Kushmerick from University College Dublin, Marc Millier from Intel, Elizabeth Rosenzweig from Kodak, and Kevin Brooks from Motorola for their insight and suggestions for the direction of my project. I also want to thank Kodak for the use of a DC290 zoom camera.

I want to thank Timothy Lee for his comprehensive research into combining handhelds and recognition programs. His work ensures that this project will continue next year.

I want to thank Earl Wagner and Hugo Liu, who made coming into the dark Pond of the Media Lab fun and relaxing. I thank them for setting a high bar of excellence for me to aspire for.

I want to thank Mindy Chang, Robert Chang, Priscilla Cheung, Rusan Hsiao, Jennifer Louie, and Rayka Yokoo for braving the streets in Cambridge. Their careful measurements provided valuable statistics for the evaluation tests.

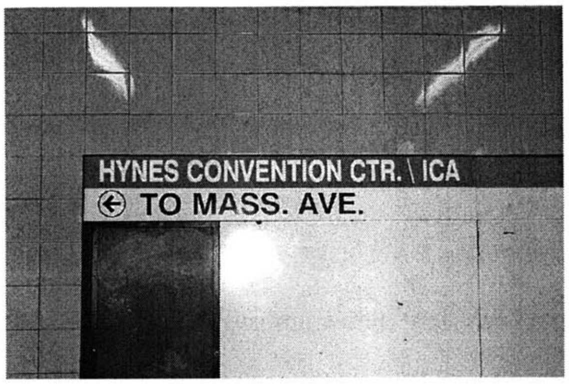
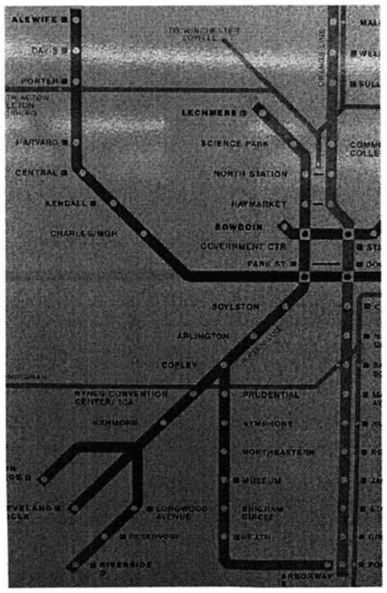
I want to thank Laurie Bittman, Vivian Cheung, James Choi, Rusan Hsiao, Liza Leung, Anya Ligai, Alisa Rhee, Amy Wong, and Cody Yiu for their prayers and emotional support during my year as a graduate student. Without their friendship, I would not have been able to publish this thesis on time.

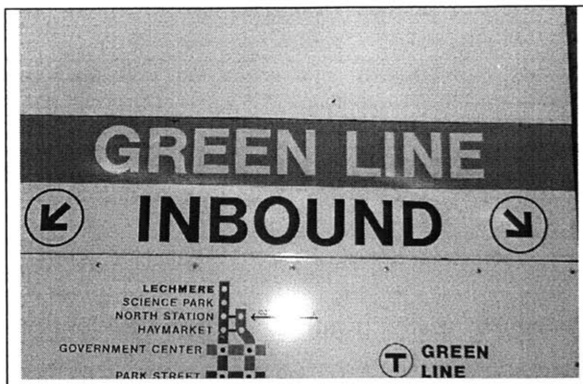
I want to thank my parents and my brother for their love and for their continued faith in me.

Finally, I want to thank God for providing me with the opportunity to contribute a little something to technology. Thank you for providing the will and the ability to finish this research, especially during the late nights and the sluggish days.

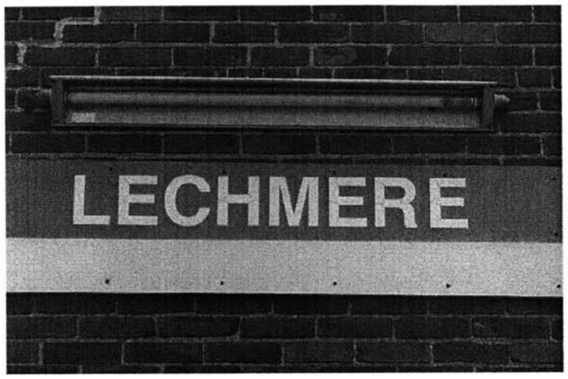
Appendix A: MARCO Scenario

The following list of annotated images is an example of the detailed directions the MARCO agent produces for a user who wishes to travel from the Hynes Convention Center subway stop to the CambridgeSide Galleria mall.

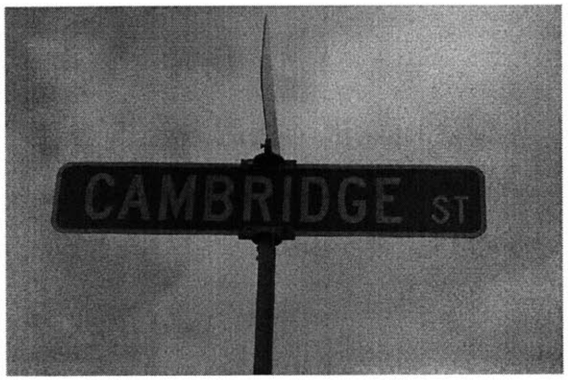
Image	**Directions**
	Start out at the Hynes Convention T stop on Massachusetts Ave.
 ALEWIFE DAVIS PORTER HARVARD CENTRAL KENDALL CHARLES/MGH LECHMERE SCIENCE PARK NORTH STATION HAYMARKET BOWDOIN GOVERNMENT CTR PARK ST BOYLSTON ARLINGTON COPLEY HYNES CONVENTION CENTER/ICA KENMORE PRUDENTIAL SYMPHONY NORTHEASTERN MUSEUM LONGWOOD AVENUE RESERVOIR RIVERSIDE BRIGHAM CIRCLE HEATH ARBORWAY	You will take the Green Line, inbound, until the Lechmere stop, at the end of the line.



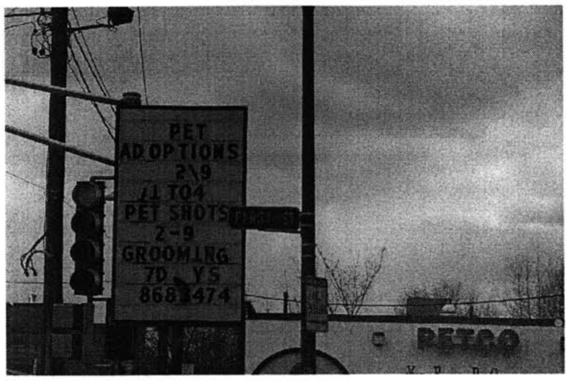
Walk down the stairs to the Green Line Inbound platform.



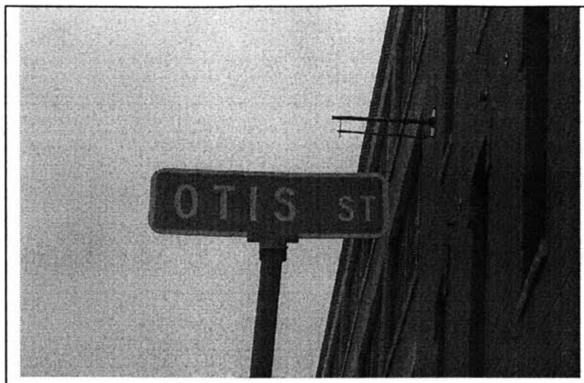
Get off at the Lechmere station.



Turn right onto Cambridge St.



Turn left onto First Street.

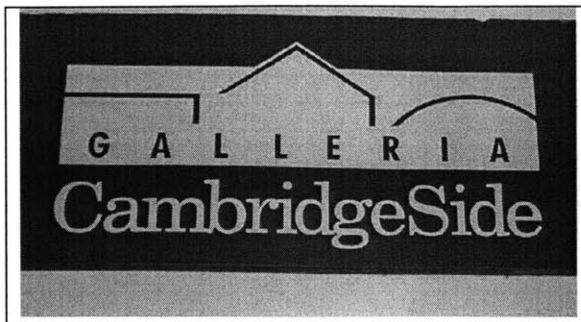


Cross Otis Street, toward a redbrick warehouse.

Walk past One Canal Park on your left.

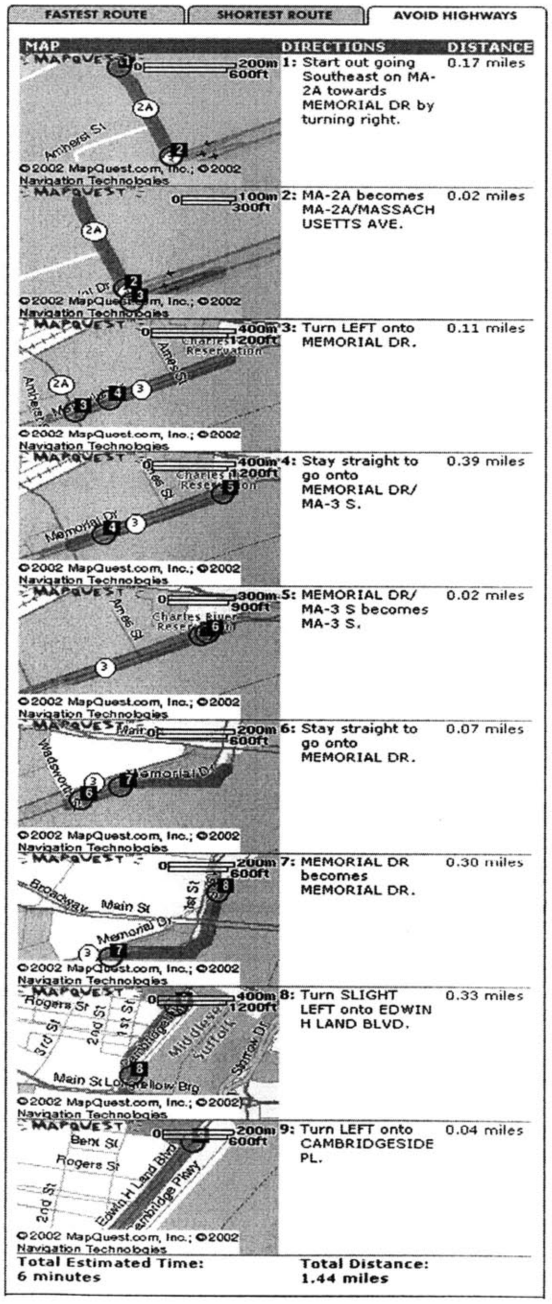
Walk past the Sears store entrance on your left.

Turn left onto CambridgeSide Parkway.



You are now at the CambridgeSide Galleria entrance.

Appendix B: Mapquest Scenario



FASTEST ROUTE | SHORTEST ROUTE | AVOID HIGHWAYS

MAP	DIRECTIONS	DISTANCE
	1: Start out going Southeast on MA-2A towards MEMORIAL DR by turning right.	0.17 miles
	2: MA-2A becomes MA-2A/MASSACHUSETTS AVE.	0.02 miles
	3: Turn LEFT onto MEMORIAL DR.	0.11 miles
	4: Stay straight to go onto MEMORIAL DR/ MA-3 S.	0.39 miles
	5: MEMORIAL DR/ MA-3 S becomes MA-3 S.	0.02 miles
	6: Stay straight to go onto MEMORIAL DR.	0.07 miles
	7: MEMORIAL DR becomes MEMORIAL DR.	0.30 miles
	8: Turn SLIGHT LEFT onto EDWIN H LAND BLVD.	0.33 miles
	9: Turn LEFT onto CAMBRIDGESIDE PL.	0.04 miles

Total Estimated Time: 6 minutes

Total Distance: 1.44 miles