Multimedia Systems
DOI 10.1007/s00530-006-0052-y

REGULAR PAPER

An interface for mutual disambiguation of recognition errors in a multimodal navigational assistant

Henry Lieberman · Amy Chu

Abstract Users often have tasks that can be accomplished with the aid of multiple media – for example with text, sound and pictures. For example, communicating an urban navigation route can be expressed with pictures and text. Today's mobile devices have multimedia capabilities; cell phones have cameras, displays, sound output, and (soon) speech recognition. Potentially, these multimedia capabilities can be used for multimedia-intensive tasks, but two things stand in the way. First, recognition of visual input and speech recognition still remain unreliable. Second, the mechanics of integrating multiple media and recognition systems remains daunting for users. We address both these issues in a system, MARCO, multimodal agent for route construction. MARCO collects route information by taking pictures of landmarks, accompanied by verbal directions. We combine results from off-the-shelf speech recognition and optical character recognition to achieve better recognition of route landmarks than either recognition system alone. MARCO automatically produces an illustrated, step-by-step guide to the route.

Keywords Context-aware applications · Human–computer interaction · Speech recognition · Optical character recognition · Scripting · Multimodal input

H. Lieberman (✉) · A. Chu
MIT Media Laboratory, 20 Ames Street, Cambridge, MA 02139-4307, USA
e-mail: lieber@media.mit.edu

A. Chu
e-mail: amyc@media.mit.edu

1 Introduction

Imagine you are planning a party at your house. Since many of the guests will be unfamiliar with where your house is located, you decide to provide a map. Since you live near downtown Boston, where many streets are poorly marked, and many of the guests will be walking or taking public transportation, a Web map such as those produced by Mapquest may not be that helpful. You would like to make step-by-step directions, illustrated with pictures so that the guest can see they are on the right track at every step.

Starting at a well-known city landmark, you take your cell phone and walk the route yourself. At each important turn, sign, or landmark, you snap a picture, and speak directions into the phone, "Turn left at this corner, where you see the furniture store", or "Here's the number 87 bus stop".

Back home, you start MARCO (multimodal agent for route construction) on your computer. It uploads the data from your phone, and outputs a Web page that has pictures of every step, captioned with the directions for that step. It also produces a link that the recipient can use to download a MARCO route to their phone, to read step-by-step as they follow the route. As a result of using this software, all the guests arrive on time for your party (well, maybe we won't be able to really assure that!).

The above scenario is one that we intend to support using the MARCO system described in this paper. MARCO combines speech recognition, visual recognition, and user task context, and integrates the graphics and text to construct an illustrated route description.

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. Again, our approach is not to construct these functions from scratch, but to utilize existing application capabilities already on the machine, and connect them using scripting to minimize manual intervention by the user.

The agent assists users in giving urban navigational directions through digital photos and verbal dictation, and displays the results as a series of route descriptions. The primary target user for MARCO is a nontechnical user preparing a route description, but we also envision a companion application to be used by those following the route.

Though the paper covers three topics – (1) a system for creating directions to a location in a city; (2) combining different modalities from various recognition systems with user-context to provide more robust results and services; and (3) using a scripting language to "stitch" together off-the-shelf applications – we are more interested in the general issues of combining OCR and speech recognition. This paper discusses pragmatic and software-related issues in multimodal system implementation, using the navigational assistant as an exploratory example.

Related work

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

Mutual disambiguation of multimodal systems

Sharon Oviatt's [15] study of the multimodal systems is a good modern survey of the mutual disambiguation of multimodal systems. 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 [16]. This "mutual compensation" of recognition errors results in a system that is more reliable than the individual systems [1]. A flexible multimodal interface also allows users to interact effectively across multiple tasks and environments [13], especially mobile environments [14].

Conveying routes through descriptions and depictions

Barbara Tversky and Paul Lee [9] studied the use of pictorial and verbal tools for conveying routes. 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").

Voice recognition and OCR on handheld computers

There has been substantial research in the area of information annotation devices. One such device is Ismail Haritaoglu's InfoScope [6], an automatic sign translation device for foreign travelers. 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.

Current multimodal map applications

A slew of multimodal systems featuring voice input and gesture input have been designed in the past few years. Multimodal Maps is a map-based application for travel planning that combines handwriting, gesture, and speech input [4]. AT&T's MATCH provides a multimodal interface for mobile information access with spoken and graphical interaction [7]. The MIT AI Laboratory has conducted significant research in combining verbal input and sketching [3]. Other work done on integration voice recognition with other modes of input includes systems like HearThere [5] and MUST [2].

Feasibility analysis for multimodal recognition in navigation

In CHI 2002 [11], Lieberman described a feasibility study for a multimodal navigational assistant. 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. An improved feasibility test shows 94% combined landmark accuracy, which will be discussed in more detail later in the paper.

2 Motivation

2.1 Unusual approaches to the recognition problem

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 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 suspected 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.

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.

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 [10]. The agent can mimic the user – selecting menus, clicking buttons, and typing words. Nonprofessional users do not tend 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.

The character recognition system (Caere OmniPage) was designed to recognize scanned documents, not color digital photos. The speech recognition system (IBM ViaVoice), a speaker-dependent, large vocabulary recognition system, was designed to understand dictated documents and application commands. Any additional vocabulary must be trained by the user in his or her specific voice.

Using a scripting language and user interface agents to stitch together application, without relying on an API also raises many UI issues: cursor control is taken away from the users at arbitrary times, buttons are clicked automatically like a self-playing piano, and the users must be careful not to interfere with the application for fear of creating a runtime error. However, integrating the commercial applications together at a higher

level provides for easier debugging and a more intuitive process.

2.2 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").

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, the GPS is only available outside and would not be 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. Nevertheless, the GPS information recorded by the capture device in real time would be useful in several ways. It could be used to synchronize the pictures with a conventional (e.g. Mapquest-style) map, or be integrated with a system used by a route follower, so he or she could ask, in effect, "Am I in the right place?".

Instead of using web-based services or digital tags, we imagine that the user will have a camera-equipped phone, or other PDA capable of recording still images and audio, and demonstrate the route that he would like his guests to follow, physically traveling between the starting point and the destination. Along the way, the user will take digital pictures of key landmarks (street signs, store names, etc.) and dictate directions for each step. For example, the user walks by a Sears department store entrance, takes a picture of the store sign, and says into the audio recorder of the digital camera, "Walk past the Sears store entrance on your left."

While this suggested navigational assistant gives the map users additional context via photos 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 will be fixed, it is rare that guests will 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

Fig. 1 Example route from Mapquest

that combining OCR and speech recognition technology can improve the overall accuracy of recognition.

3 Improved feasibility results

To supplement the feasibility tests conducted by Lieberman [11], we performed new tests with a different set of routes and improved data capture methods. The improved feasibility studies show greater than 94% landmark recognition accuracy when utilizing multiple input sources. We analyzed 9 routes and 92 direction steps. In the numbers below, we present the raw word-level accuracy rates of the speech recognition and the landmark accuracy. The combined landmark accuracy is the percentage of landmarks recognized by either the speech 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%

When calculating the combined accuracy rate, we disambiguated the results by hand, choosing the best results from each of the recognition outputs. Thus, the 94.7% landmark accuracy rate is an upper bound. Hand simulations, using techniques described subsequently, lead us to expect that we will be able to 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 obtain 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 paper are to demonstrate the feasibility of creating a more robust multimodal recognition system in the future. This higher-level goal is apparent in the workflow outline of the user tasks and computer procedures, shown in Fig. 2.

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 limitations because we are not re-programming the applications. By manipulating the 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. Currently, we are developing heuristics for automatic data selection in the results integration step. Details of the heuristics are described later in the paper.

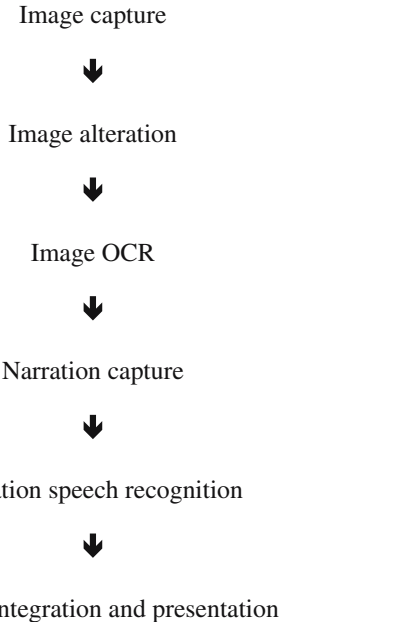

Fig. 2 Workflow outline of the user tasks and computer procedures

For each of the workflow steps, we describe some of the "tweaks" and workarounds 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 their 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 photos 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 photos 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 photos.

In addition, inexpensive digital cameras are too slow to take action, or moving, shots; photos 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. Photos 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.

4.3 Image alteration

Since OmniPage was designed to recognize scanned paper documents, it 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. We used Adobe Photoshop, a commercial photo design and production tool, to rectify most of these image imperfections. Thus, the user does not have to manually intervene during the image alteration process because MARCO automatically changes the images submitted by the user.

Photoshop does not have an automatic image alignment function. The OCR application, OmniPage, can automatically align images during the recognition process; however, rotation of images is limited because OmniPage is designed for scanned documents and it is expected that scanned documents be rotated by either a few degrees or by increments of 90 degrees. In theory, we could use Photoshop's "Rotate by degrees" 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 paper, however, we chose to allow OmniPage to automatically align the images.

To simulate the appearance of a scanned office document, other modifications to the images 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, cropping excess noise, resizing the file, and ramping up the contrast or brightness level. 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 OmniPage. OmniPage can recognize various fonts, but it cannot recognize all types of fonts. There is a spell check available while running the OCR, but our system does not utilize it because we are aiming for minimal user interaction. While spell check might not be worthwhile due to the frequency of proper names in street signs, we will include "sanity checks" that might prevent such OCR-specific errors as the misrecognition of a "S" as an "8", or constraints that a word be plausibly pronounceable. The OCR produces better results if the image is cropped or if only the desired area to be recognized has been selected. Alternatively, OmniPage has an "auto select text" feature, but it does not always select the correct portion to be recognized. 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 photos did not, however, produce better results than digital photos. In fact, depending on the quality of the scanner, the results of scanned photos may be worse.

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 recognition was not able to pick up many coherent sentences. We were also limited by low-quality 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 had been recognized. The narration elements were manually synchronized with the images for this experiment. We used a headset microphone, which produced better recognition results than the digital camera's microphone; however, we were unable to simulate the original environment (i.e. a noisy street, cars honking in the background, or a person coughing nearby), which explains the higher recognition rate.

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 will display 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 level of room noise, the volume of the microphone, and processes the pronunciation and speech patterns after the user dictates a short passage for 10–15 min . Since the system has a limited dictionary, it may not recognize rare or proper nouns such as street names or store names. Instead, the user must typically train each new word by typing the word and recording the pronunciation.

A considerable improvement in the raw speech recognition rate would be to employ named entity recognition from a database of geographic locations for the region in which the device was being used, or from some other large corpus, as do some speech recognition systems such as BBN Byblos. For the purposes of this experiment, however, we accepted the failure of the speech system to recognize many proper names, to see if we could disambiguate them by reconciling them with the visual input.

4.7 Results integration and presentation

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, 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.

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 that 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.

Therefore, we decided to look at 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 often 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.

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 the Lechmere subway station 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," and so on.

Hundreds of these speech acts were compiled by analyzing examples of route directions, taken from the Internet or from people's emails of directions to parties, etc. 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. Since we annotated each example with "<landmark>" tags, we fed as many as 500 or 1,000 examples into an information extractor (IE), an algorithm that "learns" sentence patterns [8]. More sophisticated information extraction techniques, such as those employing more advanced named entity recognition, are also possible, but again, the information extraction phase of this process is not the main point of this work. Utilizing these groups of sentence patterns, we wrote Lisp programs to alert the agent when a landmark is expected in a sentence. 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. 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.

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. 3, 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 mirror' and the OCR produced 'Lechmere.' Which of the 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 system can also save the results of user disambiguation for use in the future (if one user prefers "Lechmere" over "Leach Mirror" it is likely that all users will do so), so that it can learn over time.

After disambiguating the two recognition results, a web page is automatically brought up, displaying each image, followed by the disambiguated recognition results (as shown in Fig. 4).

4.8 Stitching together the applications

To automate the recognition process, we used Applescript – a language used to control applications on the Macintosh OS – to stitch together a myriad of commercial applications (Photoshop, ViaVoice, OmniPage). 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 3 min 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:

```
-- Run OCR on the first photo from the
processing folder tell application "Photoshop"
   open jpg_file
   do script "transform"
end tell
set pct_file to
   alias (replace_chars((jpg_file as text),
        "jpg", "pct") as text)
tell application "OmniPage Pro 8.0"
```

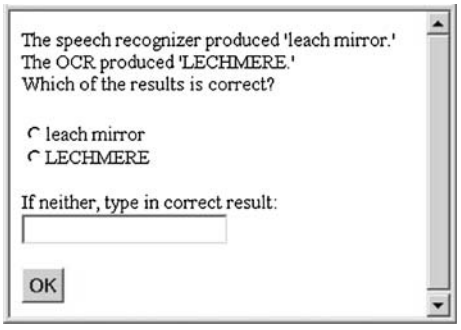

Fig. 3 Screen asking the user to disambiguate results

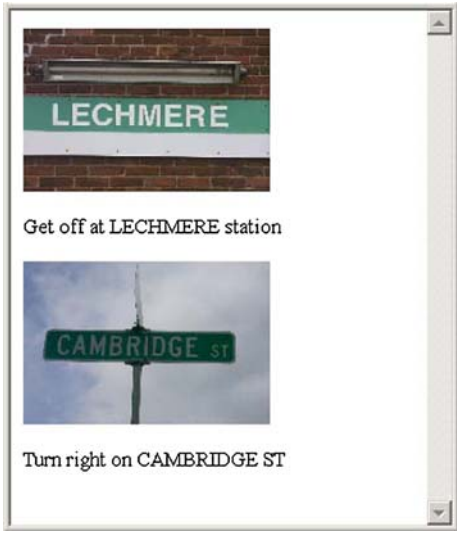

Fig. 4 Example output produced by MARCO

```
   set output file to
        ((temp_folder as string)
          & "ocr-output")
   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. Since Applescript is not designed to program software agents, there are some applications that are not scriptable, such as IBM ViaVoice. We often have to resort to workarounds with the help of third-party programs like QuicKeys to simulate user-computer interaction. 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.

Our current application is centered around batch postprocessing on a desktop or a laptop computer. Again, once mobile devices reach sufficient compute power, we expect to run most of the processing on the device itself in real time. Intermediate solutions based on communicating portions of the data from the mobile for postprocessing on a server are also possible.

5 Future applications

Further applications of this research include navigatin large buildings, helping the elderly, and extending the ARIA system [12].

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.

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. As particular routines, such as laundry or shopping, are so specific to each person, and need for specific kinds of reminders differs in 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.

The ARIA system is a software agent that proactively looks for opportunities for image annotation and retrieval in everyday work, like email messaging [12]. It reduces user overhead and leads to fewer missed opportunities for image use by searching through a database of annotated images. The current ARIA system does not utilize visual recognition of images or speech recognition. But it could put together OCR of images and verbal annotations made at image capture time as we do for MARCO, to provide an additional source of semi-automated image annotation.

6 Conclusion

This paper has presented MARCO, a Multimodal Assistant for Route Construction. MARCO shows that redundant sources of visual and speech information can be used to create route descriptions, even if there are flaws in the underlying recognition procedures. We also show how recognition applications intended for other uses can be "stitched together" to serve this application. We expect that this kind of "stitching" will become more common as time goes on, to avoid duplicating and re-implementing functionality already present on systems. Finally, we hope that applications like MARCO will help keep at least a few users from getting lost.

References

1. Abowd, G.D., Mankoff, J., Hudson, S.E.: OOPS: a toolkit supporting mediation techniques for resolving ambiguity in recognition-based interfaces Special Issue on Calligraphic Interfaces. Comput Graph. **24**(6), 819–834 (2000)
2. Almeida, L., Amdal, I., Beires, N., Boualem, M., Boves, L., den Os, E., Filoche, P., Gomes, R. , Eikeset Kudsen, J., Kvale, K., Rugelbak, J., Tallec, C., Warakagoda, N.: Implementing and evaluating a multimodal and multilingual touris guide. In: van Kuppevelt, J. et al. (eds.) Proceedings of the International CLASS Workshop on Natural, Intelligent and Effective Interaction in Multimodal Dialogue Systems, 28–29 June 2002, pp. 1–7. Copenhagen, Denmark
3. Alvarado, C., Randall, D.: Resolving ambiguities to create a natural sketch based interface. In: Proceedings of IJCAI-2001 (2001)
4. Cheyer, A., Julia, L.: Multimodal maps: an agent-based approach. In: Bunt, H., Beun, R.J., Borghuis, T. (eds.) Multimodal Human–Computer Communication. Lecture Notes in Artificial Intelligence, vol. 1374, pp. 111–121. Springer, Berlin Heidelberg New York
5. Donath, J., Karahalios, K., Rozier, J.: HearThere: an augmented reality system of linked audio. ICAD (2000) http://www.icad.org/websiteV2.0/Conferences/ICAD2000/ICAD2000.html

6. Haritaoglu, I.: InfoScope: link from real world to digital information space. In: Proceedings of Ubicomp, Atlanta, pp. 247–255 (2001)
7. Johnston, M., Srinivas B., Gunaranjan V.: MATCH: multimodal access to city help. In: Automatic Speech Recognition and Understanding Workshop, Madonna Di Campiglio, Trento, Italy (2001)
8. Kushmerick, N., Thomas, B.: Adaptive information extraction: core technologies for information agents. In: Intelligent Information Agents R&D in Europe: An Agent Link perspective. Springer, Berlin Heidelberg New York (in press, 2002)
9. Lee, P., Tversky, B.: Pictorial and verbal tools for conveying routes. In: Proceedings of COSIT, pp. 51–64 (1999)
10. Lieberman, H.: Integrating user interface agents with conventional applications. In: International Conference on Intelligent User Interfaces, San Francisco (1998)
11. Lieberman, H.: Out of many, one: reliable results from unreliable recognition. In: Proceedings of CHI '02, Minneapolis, MN. ACM Press, New York (2002)
12. Lieberman, H., Rosenzweig, E., Singh, P.A.: An agent for annotating and retrieving images. IEEE Comput. **34**(7), 57–61 (2001)
13. Oviatt, S.L.: Designing robust multimodal systems for universal access. In: Workshop on Universal Accessibility of Ubiquitous Computing, pp. 71–74. ACM Press, New York (2001)
14. Oviatt, S.L.: Multimodal system processing in mobile environments. In: Proceedings of UIST, pp. 21–30. ACM Press, New York (2000)
15. Oviatt, S.L. Mutual disambiguation of recognition errors in a multimodal architecture. In: Proceedings of CHI, pp. 576–583. ACM Press, New York (1999)
16. Oviatt, S.L.: Taming recognition errors with a multimodal interface. CACM **43**(9), 45–51 (2000)