A Multimodal Interface for Road Design

Alexander Blessing
University of Cambridge
acb71@cam.ac.uk

T. Metin Sezgin
Dept. of Engineering
Koç University
mtsezgin@ku.edu.tr

Relja Arandjelović
University of Cambridge
Dept. of Engineering
ra313@cam.ac.uk

Peter Robinson
Computer Laboratory
University of Cambridge
pr@cl.cam.ac.uk

ABSTRACT

Sketch and speech-based interfaces have been suggested for many design tasks. We have investigated the suitability of these modalities for road design, which makes heavy use of two dimensional visual representations. We have implemented a multimodal road design interface that combines sketching and speech. Our system allows designers to specify spatial and geometric aspects of their design using a pen-based interface that can recognize user input interactively in real-time. We recognize free-hand drawings of domain shapes, and provide seamless support for gesture-based interaction. We also support speech-based interaction for specifying aspects of the road design that are more naturally conveyed through speech. We evaluated the usability of our system for the task of designing driving courses for the STISIM driving simulator. Our initial evaluation with four users suggests that users find our multimodal interface to be superior to conventional methods.

Author Keywords

multimodal road design, sketch-based interfaces, speech-based interfaces

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: User Interfaces

INTRODUCTION

Over the last decade, there has been an increasing interest in the user interfaces research community to identify design tasks that can be improved by intelligent multimodal interfaces combining speech and sketching. Speech and sketch-based interfaces have the potential to make the design process more interactive, natural, efficient, and enjoyable. We have identified the *road design* process as a visually rich design activity that could use pen and speech inputs for specifying the geometric and spatial aspects of the design.

Road design is the iterative process of planning the layout, signs, signals, and the surrounding landscape that collectively constitute roads. Transportation Engineers may generate road designs as part of a planned road construction (*actual road design*), or they may design roads for studying various aspects (e.g., layout, speed management, signs) of an existing or a hypothetical road in a laboratory setting using driving simulators (*road design for simulators*).

Actual road design, and road design for simulators both require specifying the two dimensional layout of the road, the placement of signs, signals, elements of landscape (e.g., trees, shrubs), and buildings (see Fig. 1 as an example). These visual elements can potentially be specified by sketching. In addition, road design for simulators requires specifying the desired behaviour of the traffic during a simulation, which we believe can be done more effectively through speech.

We propose a multimodal interface for road design that allows users to use speech for things that are easier said, and sketching for things that are more effectively specified through drawing. Our design choices have been shaped by our interviews with domain experts. Therefore, we briefly discuss the list of requirements that we compiled after interviewing domain experts. Then we summarize the potentials for speech and sketch input, and describe the architecture of our multimodal road design software. We present evidence on the utility and usability of our system in the evaluation section. We conclude by a review of the related work and point out extensions that might be considered for future work.

DESIGN CONSIDERATIONS

We based our design on a set of requirements that we have identified after interviewing domain experts.

Interviews with domain experts

We interviewed two domain experts, a Transportation Engineer specializing in actual road design and a Computer Scientist who regularly designs roads for driving simulators for human factors research.

The interviews indicate that in the early design phase of ac-

IUI 2009 Workshop on Sketch Recognition
February 8, 2009, Sanibel Island, Florida
Chair: Tracy Hammond

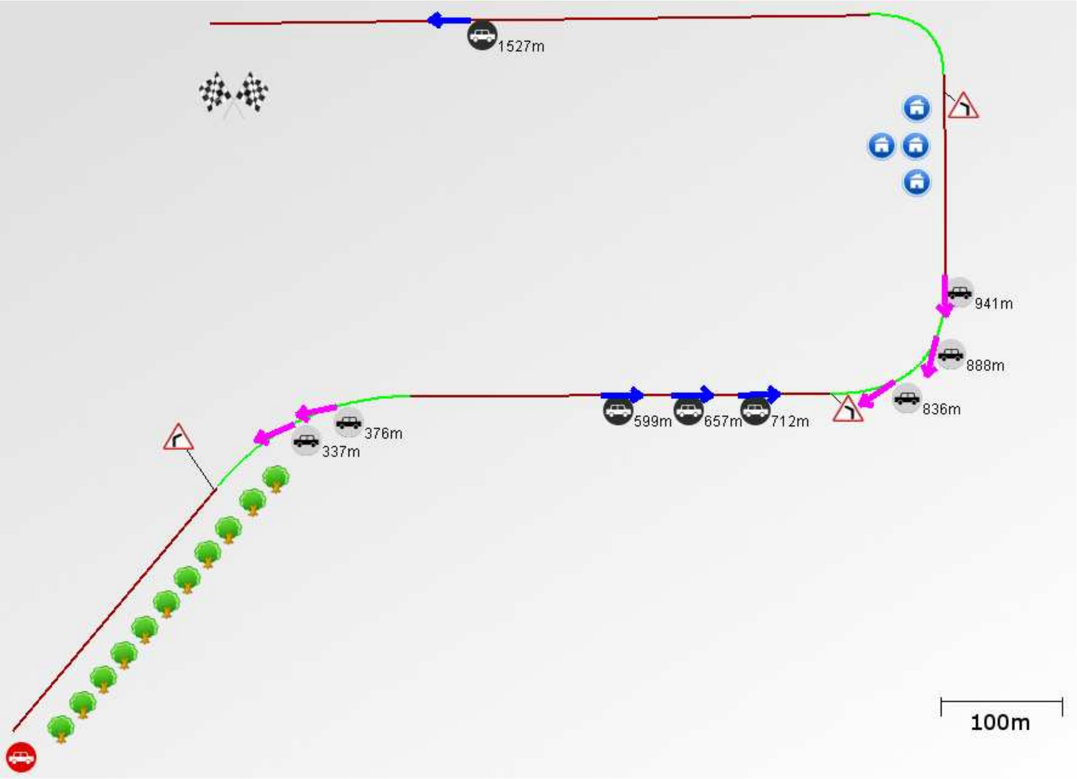


Figure 1. Screenshot of MIRA: Multimodal Intelligent Road Design Assistant. The road is displayed as a series of straight and curved segments. The vehicle icon in the lower left corner indicates the starting point of the drive in the simulator. Vehicles making up the traffic are annotated with their direction of travel, and distance from the start line of the simulation. Other elements of the road design include the trees, buildings, and signs.

tual road design, Transportation Engineers work with paper maps. They explore alternative layouts for the road by drawing straight line segments called "tangents" on the map. Tangent lines along the road serve as visual representations of the curvature constraints along the road. The tangent lines are combined by special curves called "clothoid curves," which provide a smooth change in curvature across successive tangent lines. The drawings are usually done on large scale maps (e.g., 1:5000), and when the designer is satisfied, the details of the road design are finalized using traditional WIMP interfaces.

Although there are a number of commercial driving simulators, our expert used the STISIM Driving Simulator [3], which appears to be a popular choice in Human Factors and Road Safety research communities [6, 1]. Road design for the STISIM simulator is done through programmatic means using a scripting language called Scenario Definition Language (SDL). The road layout, signs, signals, and the details of the traffic are specified using ASCII text commands [1]. Our domain expert indicated that, because the overall appearance of the road is hard to visualize, there is a need for graphical means of viewing and editing road designs.

Based on the interviews with the domain experts, we have identified the following affordances desirable for road design:

- **Road-layout specification by drawing:** The system should be able to recognize the freehand sketch of a road consisting of straight line segments connected by curves. Because road designers use tangent curves when they work on paper, a natural choice for road-layout specification would be sketching the tangents of the road.
- **Road scenery:** We would like the system to provide support for adding trees and buildings to the scenery. Both buildings and trees should be easily draggable.
- **Traffic:** When designing roads for simulation, the system should allow easy specification of the traffic conditions in the road (e.g., the number of cars on the road, spacing between vehicles etc.).
- **Road signs:** The system should support adding road signs.
- **Editing:** Editing and correcting errors should be easy.

SYSTEM DESCRIPTION

Based on our interviews with domain experts we have developed MIRA, the **M**ultimodal **I**ntelligent **R**oad Design **A**ssistant, which supports sketch and speech based interaction.

The requirements list compiled after interviews with the domain experts served as a set of basic features. We imple-

[1] Although there has been some research on developing alternative interfaces for the task, the state of the art at the moment is limited to an interface based on MS Excel [7].

mented these features using speech, sketching and WIMP-based modalities. We came up with an initial mapping between the features and the input modalities while trying to match the affordances required by each feature to those supported by the modalities.

While certain operations could be completed through individual modalities (speech, sketch and WIMP-based), for others we used combinations of modalities. For example, using conventional tools, it is inherently difficult to describe the behaviour of an object or the interaction between a set of objects (e.g., "the cars keep a constant distance," or "the traffic light turns green when the driver approaches the 50 meter mark"). Such commands require several steps of interaction in a traditional GUI (e.g., object selection, launching a properties panel, setting new properties, interaction with dialog boxes). On the other hand, these commands can be specified concisely using natural language, therefore we used speech for these tasks.

Specifying the road layout by drawing tangent lines appears to be appropriate. Selection operations, which are inherently spatial, can be completed more easily by a stylus, hence these are supported by the pen interface.

For adding road signs, a traditional WIMP interface is likely to be inefficient, mainly because locating the target road sign among a set of signs would require substantial amount of navigation, while simply drawing the required road sign using a pen-based interface is likely to be more appropriate. Although the option of adding signs through spoken commands is also plausible, we opted for pen-based input for the sake of not overloading the speech modality.

Where possible we used sketching and speech in a complementary fashion so the user can select a group of objects by pen and specify/edit their properties by speech. For example, the user can select a group of cars and say "the cars are separated by 200 meters," or "the cars are traveling in the same direction as the driver[2]." MIRA has a simple reference resolution scheme where if the recognized speech command requires an accompanying sketch component, it is associated with the nearest compatible drawing/selection event. Next we describe the sketch and speech processing in MIRA.

Sketch recognition

MIRA supports three different kinds of processing for pen input: fragmentation, single object recognition and gesture recognition.

Fragmentation transforms the user's sketch of the road tangents into a segment of straight lines and Bézier curves. We first connect consecutive tangent lines by cubic curves to obtain a continuous stroke. The stroke information is then processed by our fragmentation algorithm.

Fig. 2 shows the three steps of the fragmentation process. This algorithm runs in $O(n^2N)$, where n and N denote the number of segments and number of points, respectively. Since the number of segments is small compared to the number of points in our application, the algorithm runs fast enough to provide real-time feedback.

MIRA supports a small range of **gestures**. The intention to add a road sign is indicated using a gesture. There is a gesture for adding a single car or removing an object (car, tree, building) from the scenery. Gestures are recognized using a recognizer based on the Rubine features [19].

The features extracted from the stroke are: the cosine and the sine of the initial angle of the gesture, the length and the angle of the bounding box diagonal, the distance between the first and the last point, the cosine and the sine of the angle between the first and last point, the total gesture length , the total angle traversed, the sum of the absolute value of the angle at each mouse point and the sum of the squared value of those angles.

The features are passed on to a Support Vector Machine (SVM) for classification. This is done using LIBSVM [11]. The gestures used in our system are listed in Fig. 3. A separate gesture, the selection gesture, uses only two features, the distance between start and end points and the total gesture length. A stroke is identified as a selection if the distance between start and end point is small compared to the total stroke length. The selection gesture is used to select objects it encloses. This is useful for operations such as dragging a group of trees or modifying properties of a group of objects using speech.

Road signs can be sketched in a separate panel (shown in Fig 4) that gets launched when the user draws gesture 2 in Fig. 3. When the drawing panel is active, the system attempts to recognize the sign after each stroke is added, and displays the three most probable interpretations in the bottom. The list is updated after each stroke. The user can then select the target sign as soon as it appears in the list. This relaxes the restriction that each object has to be fully drawn for recognition, and speeds up the interaction for cases where a partial drawing of a sign has sufficient information for recognition.

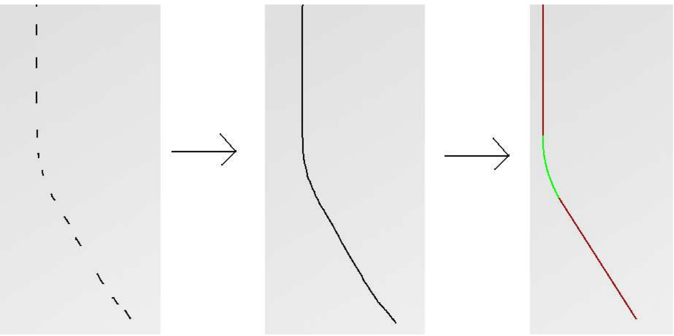


Figure 2. The three phases of fragmentation. A continuous curve is obtained by connecting line segments with Bézier curves. Then, the entire stroke is segmented into linear curved segments using its global properties.

[2] "The driver" refers to the participant who would be using the driving simulator in an experiment

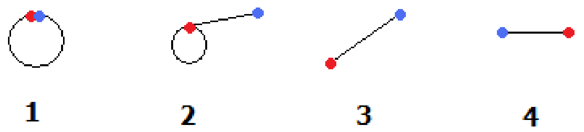


Figure 3. Gestures used in MIRA. The red and blue dots indicate the start and end point, respectively. Gesture 1 adds a single car to the road, whereby gesture 2 opens a new panel to draw a road sign. Gesture 3 deletes the underlying object and the last gesture undoes the last stroke in the road sign panel.

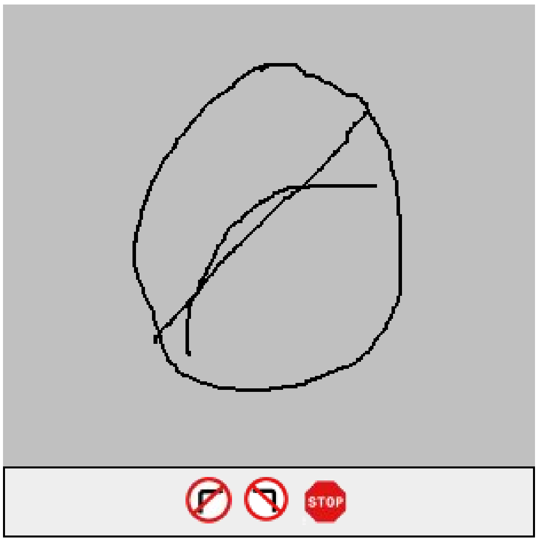


Figure 4. When the drawing panel is active, the system attempts to recognize the sign after each stroke is added, and displays the three most probable interpretations in the bottom.

Signs are recognized by an isolated object recognizer. We use an algorithm that uses a combination of Zernike moments [16] and Hidden Markov Models (HMMs) [20] to provide scale and rotation invariant symbol recognition. For the Zernike moment-based recognition method, we use the first 12 moments. Fig. 5 lists the road signs that our system currently recognizes, while Fig. 6 shows two sketched representations of those symbols. We believe the benefit of supporting hand-drawn sign recognition will become more evident as the number of road signs increases.

Speech recognition

MIRA's speech recognition system is based upon the Microsoft Speech API 5.3 (SAPI) [4]. We decided in favour of Microsoft's speech recognition engine due to its good recognition rate and its free availability. The speech library was developed in C#, and the Java Native Interface (JNI) was used to communicate the recognition results to MIRA's speech handler (written in Java).

A subset of the grammar used by our system is listed in Fig. 7. The grammar serves two major purposes: It allows the user to specify actions without having to switch modes. For example, instead of clicking on a button, the user can simply create the road using an appropriate voice command.

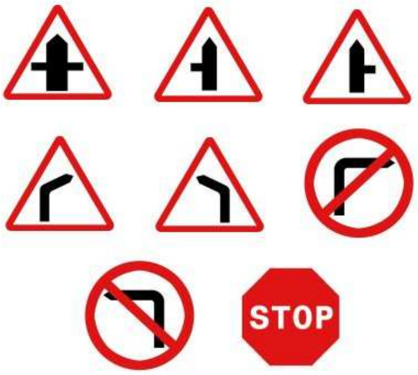


Figure 5. Road signs currently recognized by our system.

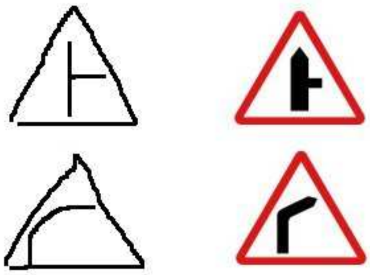


Figure 6. Example of sketched symbols and the corresponding recognized objects.

Other examples include adding trees, buildings or cars to the road or removing a group of objects. More importantly, speech enables the user to convey information that is otherwise more difficult to express. For example, users can use the selection gesture on a group of cars and specify their relative displacement using speech. A video demonstrating MIRA's speech and sketch processing capabilities can be seen at on our web page [2].

PILOT TESTING AND EVALUATION

We conducted pilot testing with four users before evaluating our system.

Pilot testing

The purpose of the pilots were to improve our implementation and revise our design choices based on the iterative design methodology. We asked four users to complete representative design tasks at various points during the design process, and updated our design based on the feedback that we received.

One aspect of the design that we modified after pilot testing had to do with editing. Initially we included speech support for short spoken commands such as "undo," "redo," "delete," and "clear." What seemed like a reasonable design choice proved to hamper the usability of the system, because the limitations of the speech recognition technology occasionally caused noise, and spurious speech to be misrecognized

```
#ABNF 1.0;

$Construct = Construct [the] road ;

// Some basic constructions often used
$Pre =   (There are) |
         (Place) |
         (Put) |
         (Add) |
         (Create) ;

$Number = 1 | 2 | 3 ... | 9 | 10 ;
$Distance = 20 | 50 | ... | 100 | ... | 200 ;
$Direction = in (driver’s | opposite | other) direction;

// Construct the road
$Construct = (Construct | Create) the road;

// Add intersection
$Intersection = Add (Intersection | Junction);

// Add cars
$Car =   $Pre $Number cars [here] ;

// Add buildings
$Building = $Pre [some] (buildings | houses) [here] ;

// Make cars equally spaced
$EqualSpace = The cars (have the same distance | are equally spaced);

// Space the cars
$SpaceCars = ( (These | The) cars
                are (separated | spaced)
                by $Distance meters);

// Change cars direction
$ChangeDirection = (These cars | This car)
                   (goes | go | are going | is going)
                   $Direction;

// Add cars to driver’s direction
$CarDriverDirection = $Pre $Number (car | cars) [in driver’s direction];

// Add cars in opposite direction
$CarOppositeDirection = $Pre $Number (car | cars) in (other | opposite) direction;
```

Figure 7. A subset of MIRA’s grammar.

as one of these short utterances[3]. Although this did not happen very frequently, it was a major problem, because unexpected executions of these commands resulted in confusing software behaviour (especially inadvertent recognition of destructive commands such as “delete,” and “clear” frustrated the users). Therefore, we removed these commands from the speech grammar, and added easy-tap buttons to the graphical interface to allow these operations to be completed by tapping with the pen.

Another issue that came up during our pilot tests was that the users had preconceptions about how a speech-based system would work based on their past experience and interaction with existing speech systems, which are mostly based on keyword spotting (e.g., phone-based customer systems with speech recognition capabilities). Because we initially did not instruct our pilot test participants on how the speech interface would work, some of them initially tried to control the system by uttering keywords that they thought would elicit the appropriate response. Rather than modifying our speech system to be based on keyword spotting, we chose to prepare a 2 minute long video that showed someone interacting with the system using speech [2]. We found the video to be highly effective in helping the users to break their preconceptions. Not having to convert to a keyword-spotting-based approach proved to be particularly advantageous, because certain operations that we could perform within a grammar-based approach could otherwise not be supported by speech.

Evaluation

We finalized our design through several iterations of pilot testing and reimplementation. In order to measure the utility and usability of our system, we conducted a user study where we asked four subjects to complete a set of road design tasks using MIRA and a WIMP-based baseline. Users who participated in the pilot testing did not take part in the evaluation.

WIMP-based baseline

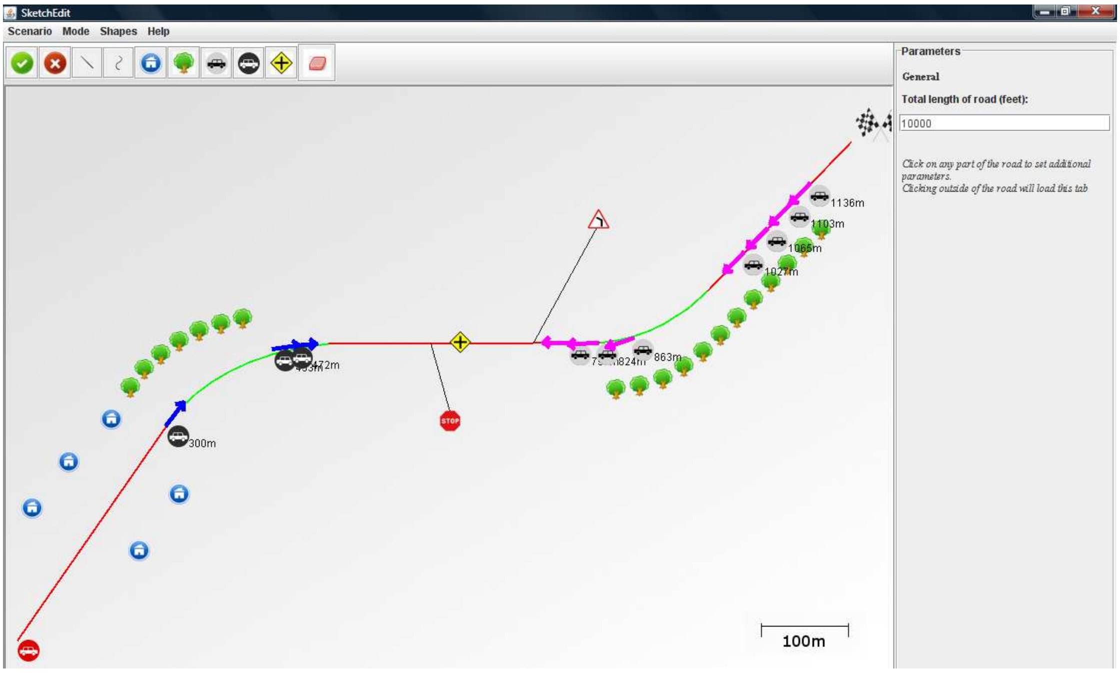
We implemented a traditional WIMP-based road design tool to serve as a baseline in our evaluation of MIRA. Fig. 8 shows a screenshot of this interface. It makes no use of sketch or speech functionality, and supports the affordances required for road design within the WIMP framework.

The trace of the road is input by concatenating straight lines and curves. A line segment is specified by its start and end point. Curves are represented with Bézier curves and specified using two end points and two additional points that serve as control points. The road is colour-coded to identify straight and curved portions, which are coloured in red and green respectively (see Fig. 8).

Trees, buildings and intersections can be selected from a toolbar and placed onto the scenery. Dragging allows exact positioning of these objects. More refined options (such as type of building or tree, number of branches at intersection) can be input through a separate parameter panel.

Cars can be selected from a toolbar and placed on a segment of the road. The system makes a distinction between

[3] Recognition of short utterances is inherently more difficult than longer phrases, because longer phrases have more contextual support.



Parameters

General

Total length of road (feet):

10000

Click on any part of the road to set additional parameters.
Clicking outside of the road will load this tab

Figure 8. Screenshot of the WIMP-based interface.

cars travelling in the driver's direction and those travelling in the opposite direction. Arrows indicate the direction. The distance of each car from the start of the road is displayed visually.

Road signs can be placed on any part of the road and are chosen from a list. Currently, we support eight signs (European models). Objects can be deleted using a traditional eraser tool.

Experiment design

For the user study, we recruited four computer science graduate students and asked them to complete a set of road design tasks. One of the participants was a graduate student who had been designing roads for the STISIM driving simulator for the past two years. The participants were asked to construct three road designs for simulation using the WIMP interface and the multimodal interface. The tasks were selected to get an adequate coverage of the features needed for road design.

Visual and textual instructions were supplied to the participants in printed form. The visual component of the instructions were screenshots of a completed road design, which the users were asked to reproduce. The textual instructions asked the participants to make incremental changes to their initial design (e.g., moving cars and buildings around, adding or removing vehicles). The participants were asked to complete all three designs using the multimodal and WIMP-based interfaces in an arbitrary order.

Results

Each participant completed a questionnaire consisting of 16 questions after completing all three design tasks. The questions were designed to measure the overall interaction experience of the participants, and their subjective judgement of both system's effectiveness in addressing specific affordances needed for the task.

Fig. 9 lists the questions included in the questionnaire. The participants were asked to assign a score 1-7 for each question, based on how much they agreed with each statement. The scores were defined as: 1 strongly agree, 2 somewhat agree, 3 slightly agree, 4 neural, 5 slightly disagree, 6 disagree, 7 strongly disagree.

As seen in Fig. 9, the multimodal interface compares favourably with its WIMP counterpart. The responses for questions #1, #4, #9, and #10 were statistically significant using a one-tailed sign test ($p < 0.1$). These suggest that the users found our system more intuitive, and certain operations (such as adding buildings, or modifying spacing constraints between vehicles) were easier with our system.

Interestingly users thought the WIMP interface was missing some functions they expected for completing the tasks, although both interfaces had all the functionality required for completing the tasks, and all participants successfully completed all three tasks.

Obviously it is desirable to extend the evaluation with more subjects in order to obtain a tighter bound on the statistical significance, and for coming to a definitive conclusion on the significance of the other statements. However, we regard the results obtained with this small set of users as a positive indicator of the usability of our system.

RELATED WORK

	Input Method	
	MM Intf.	WIMP
1. The interface was intuitive	++	○
2. I was satisfied with how the interface worked	+	−
3. The interface was simple to use	+	○
4. The functions I expected to complete the tasks were available	+ + +	○
5. I could effectively complete the tasks	++	+
6. I could complete the tasks quickly	++	−
7. I thought the ”look and feel” of interface was pleasant	++	○
8. Adding vehicles was easy	+ + +	○
9. The spacing constraints between vehicles was easy to satisfy	+ + +	−−
10. Adding buildings was easy	+ + +	−
11. Adding trees was easy	+ + +	++
12. Adding road signs was easy	○	+
13. I could easily recover from errors	+	○
14. I enjoyed interacting with the interface	+	−
15. The interaction felt like a human to human interaction	−	−−
16. The interaction felt natural	++	○

Figure 9. Questions included in the exit survey. The first column lists a number of statements. The users were asked to assign scores on a seven-point Likert scale based on how much they agreed with the statements for each interface. We divided the seven-point Likert scale (which has a range of six) into seven equal regions. The region containing the average score for each question is indicated with ’+’ and ’−’ signs, where ’+ + +’ shows the region with most agreement, and ’− − −’ shows most disagreement.

We discuss related work in sketch and speech based multimodal interfaces and then summarize the state of the art in road design practices.

State of the art in multimodal interfaces

There is vast amount of previous work on multimodal speech and sketch interfaces. Here, we limit our discussion to the more recent work and also mention a few of the early influential systems.

The survey paper by Oviatt et al. provides an overview of some of the early work on multimodal systems[18], and also provides a discussion of motivations for sketch and speech based multimodal systems. Our work shares the same motivations.

Some of the earlier work in the field has focused on reference resolution issues for matching speech fragments to parts of the sketch [14, 17, 8]. More recent relevant work includes those on reference resolution for speech and sketch-like modalities such as gesturing [15, 13, 10]. All these systems discuss fairly elaborate methods for reference resolution. We show that the simple reference resolution technique used in our system is sufficient for effective multimodal interaction. In this respect, our work illustrates an interesting property of the road design domain.

MIRA is intended to aid a particular design task, hence the most relevant set of work is the work on multimodal interfaces for design. Oltmans et al. have studied the early stages of design for mechanical devices [17]. Their goal was to make conveying the behaviour of mechanical devices more natural for designers of mechanical engineers through sketching and speech. Their system, ASSISTANCE, allows a designer sketch a mechanical device and then verbally communicate the device behavior.

Later work by Adler et al. focused on a different aspect of multimodal interaction in the same domain [8]. Adler’s system allows the designer to sketch a mechanical system, and specify certain properties the system through speech. For example, it allows users to draw a rough sketch with five pendulums, and then say “there are five identical touching pendulums” in order to specify the size and placement of the pendulums through speech. This allows an operation that would otherwise require several steps through traditional modalities to be completed with a single speech command. Achieving such efficient communication has been the motivation behind our system as well. We show how multimodal interaction can be applied to the domain of road design.

State of the art in road design

Actual road designer and designers who design roads for driving simulators use different tools for these tasks.

Road design for simulators

Existing tools for simulators are limited to WIMP-based design systems or programmatic specification of simulator scenarios.

WIMP-based systems usually have their own 3D modelers [12]. These tools not only define the geometrical and spatial aspects of the terrain but also include behavioural models (for example, the interaction between cars and pedestrians). The *input phase* gathers information from cartographic databases (which include topographical information), scanned maps that replace missing details (such as road signs) and a description of the traffic light system. In the *modeling phase* the user can define the road network using a traditional GUI. The exact representation of the streets differ from case to case [12, 21]. Buildings and road signs can be added using a traditional WIMP interface.

Alternatively, the road model can be input programmatically through scenario definition languages (e.g. SDL in STISIM). Both the road geometry and the interaction components are defined in the same file in order to facilitate the easy synchronisation of both sources of information. Unlike MIRA, all these methods are based on WIMP-based or programmatic interfaces.

Road Engineering and traffic planning

Road and Transportation Engineers emphasize the planning of construction projects. This requires models that can capture the constraints of the real environment as closely as possible. Existing tools that can achieve this include commercial tools such as RoadViz [9] or AutoCAD [5]. Again, unlike MIRA, all these products are conventional WIMP-based interfaces.

DISCUSSION AND FUTURE WORK

We have presented MIRA, a **M**ultimodal **I**ntelligent **R**oad Design **A**ssistant, that incorporates sketch and speech recognition for designing roads. Our evaluation showed that users find MIRA to be superior than its WIMP-based counterpart.

The evaluation also helped us to identify issues that need further exploration. In particular, in the postmortem evaluation interview, the users indicated that because the system only had 8 signs to choose from, the menu interface worked just as good as the sketch-based method. If there were more signs to choose from, this would increase the utility sketch-based input method. It remains to be seen how WIMP-based and sketch-based interfaces would scale for larger number of symbols.

ACKNOWLEDGMENTS

We would like to thank Goran Jovanović of APPIA, Slovenia, and Ben Reaves of Toyota ITC Palo Alto, USA for useful discussion.

REFERENCES

1. Annual meeting of the STISIM Drive User Group September 17-18 2008, Universit Laval, Quebec City, QC., 2008.
2. MIRA Demo video: http://portal.ku.edu.tr/˜mtsezgin/MIRA/.
3. STISIM Driving Simulator. Systems Technology, Inc., http://www.systemstech.com/.
4. Microsoft Speech API (SAPI) 5.3, http://www.microsoft.com/speech/speech2007/.
5. Autocad 2000. San Rafael, CA 94903, USA, 1999.
6. *Programs and practices to reduce simulator sickness: lessons learned from the field*. International Conference on Road Safety and Simulation, Roma, Italy., 2007.
7. *An Excel-based programming tool for the STISIM scenario definition language*. Annual meeting of the STISIM Drive User Group, September 17-18 2008, Universit Laval, Quebec City, QC., 2008.
8. A. Adler and R. Davis. Speech and sketching for multimodal design. In *Proceedings of the 9th International Conference on Intelligent User Interfaces*, pages 214–216. ACM Press, 2004.
9. Bashir Research. *RoadViz 2000 Technical Description*, 2000. `http://www.my3d.com/roadviz.html`.
10. J. Y. Chai, P. Hong, and M. X. Zhou. A probabilistic approach to reference resolution in multimodal user interfaces. In *Intelligent User Interfaces*, pages 70–77, 2004.
11. C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines, 2001. software available at http://www.scie.ntu.edu.tw/˜cjlin/libsvm/.
12. S. Donikian. How introduce life in virtual environments: an urban environment modeling system for driving simulation. In O. Balet, R. Caubet, J.-P. Jessel, and G. Subsol, editors, *5th Workshop of the Group GT-RV*, Toulouse, France, 1996. Groupe de Travail sur la Réalité Virtuelle.
13. J. Eisenstein and C. M. Christoudias. Christoudias. a salience-based approach to gesture-speech alignment. In *In Proceedings of the Human Language Technology conference / North American chapter of the Association for Computational Linguistics annual meeting*, pages 25–32. ACL Press, 2004.
14. K. D. Forbus, R. W. Ferguson, and J. M. Usher. Towards a computational model of sketching. In *In IUI 01*, pages 77–83, 2001.
15. D. He, G. Ritchie, and J. Lee. References to graphical objects in interactive multimodal queries. *Know.-Based Syst.*, 21(7):617–628, 2008.
16. H. Hse and A. R. Newton. Sketched symbol recognition using zernike moments. In *ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 1*, pages 367–370, Washington, DC, USA, 2004. IEEE Computer Society.
17. M. Oltmans and R. Davis. Naturally conveyed explanations of device behavior. In *Workshop on Perceptive User Interfaces*, 2001.
18. S. Oviatt, P. Cohen, L. Wu, J. Vergo, L. Duncan, B. Suhm, J. Bers, T. Holzman, T. Winograd, J. Landay, J. Larson, and D. Ferro. Designing the user interface for multimodal speech and gesture applications: State-of-the-art systems and research directions for 2000 and beyond, 2000.
19. D. Rubine. Specifying gestures by example. *SIGGRAPH Comput. Graph.*, 25(4):329–337, 1991.
20. T. M. Sezgin and R. Davis. Hmm-based efficient sketch recognition. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI'05)*, page www, New York, New York, January 9-12 2005. ACM Press.

21. R. Sukthankar, D. Pomerleau, and C. Thorpe. Shiva: Simulated highways for intelligent vehicle algorithms. pages 332 – 337, September 1995.