Back Seat Driver: voice assisted automobile navigation

by

James Raymond Davis

B.S.A.D., Massachusetts Institute of Technology (1977)

Submitted to the Media Arts and Sciences Section
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 1989

Signature of Author ....
Media Arts and Sciences Section
August 4, 1989

Certified by ....
Nicholas P. Negroponte
Professor of Media Technology
Thesis Supervisor

Accepted by ....
Stephen A. Benton
Chairman, Departmental Committee on Graduate Students

Back Seat Driver: voice assisted automobile navigation

by

James Raymond Davis

Submitted to the Media Arts and Sciences Section
on August 4, 1989, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

The Back Seat Driver is a computer navigation assistant for drivers in a city. It differs from earlier navigation programs by using speech, rather than graphics, to give instructions. The advantages of speech are that the driver's eyes are left free for driving and that the spoken directions contain information not easily portrayed in pictures. The program talks about the features of the road in the same way the driver sees them, giving the impression that the program is actually is in the car.

Driving instructions are modeled after those given by people. The two issues for spoken directions are *what to say* (content) and *when to say it* (timing). The content of the instructions tells the driver what to do and where to do it. The program has a large taxonomy of intersection types, and chooses verbs to indicate the kind of intersection and the way of moving through it. The instructions refer to landmarks and timing to tell the driver when to act.

Timing is critical because speech is transient. Drivers hear instructions just in time to take the required action, and thus need not remember the instruction or exert effort looking for the place to act. The program also gives instructions in advance, if time allows, and the driver may request additional instructions at any time. If the driver makes a mistake the program describes the mistake, without casting blame, then finds a new route from the current location.

Street map data bases for navigation programs must distinguish between *physical* connectivity (how pieces of pavement connect) and *legal* connectivity (whether one can legally drive onto a physically connected piece of pavement). Legal connectivity is essential for route finding, and physical connectivity for describing the route. The database must also contain all landmark information, since the program has no "eyes".

The Back Seat Driver is an actual working prototype. It has successfully guided drivers unfamiliar with Cambridge to their destinations. Although much work remains, it is easy to foresee a practical implementation in the future.

Thesis Supervisor: Nicholas P. Negroponte
Title: Professor of Media Technology

Acknowledgments

I owe gratitude to many people for help with this work. In particular I would like to thank: Phil Rittmueller and NEC Home Electronics (USA) Inc., who sponsored it; Symbolics Inc., Digital Equipment Corporation, the Nippon Telephone and Telegraph company, and DARPA, which sponsored my earlier graduate studies; Brewster Kahle, who provided the original inspiration for my work on computerized direction giving in the summer of 1985; Thinking Machine Corporation, where the work was conducted; and Tom Trobaugh, who began that study with me. I thank those who provided me with helpful information: James A. O'Connell, Jr. and Peter Naghavi of the Somerville Department of Traffic and Parking, Jane Kent of the Cambridge Department of Traffic, and George Hawat of the Boston Transportation Department, for providing traffic light data bases; Doris Walter of the Massachusetts District Commission for maps of rotaries; Doug Milliken for explanations of car dynamics; Don Cooke of Geographic Data Technology and Joel Sobel of the Census Bureau for information on the DIME and TIGER map formats; David Pietraszewski of the United States Coast Guard and Michael Fisher of Trimble Navigation for information about GPS. The photo on page 60 was taken by Kyle G. Peltonen, and is reprinted by permission of The Tech, the Mr. Boffo cartoon is reprinted courtesy of the Tribune Media Services. I thank also Weng-yew Ko for building hardware interfaces; Elaine McNair at NEC for removing many obstacles; Michael Sullivan of Spectrum Communication for solving communications problems; Greg Parro of Cellular Phone Services for excellent service with the telephones; Hidehiro Matsumoto of NEC for translations to and from the Japanese; Janette Noss and Elaine McCarthy, who keep the Architecture Machine Group running; Anna Korteweg, Walter Bender, Gary Drescher, and Carol Strohecker for careful editing, and Shawn T. Williams and Greg Grove, who helped by participating in UROP work. I wish to thank all those who taught me of computational linguistics, especially Barbara Grosz, Julia Hirschberg, and Janet Pierrehumbert. They of course bear no responsibility for my failure to fully grasp the material they attempted to teach me. My committee members were Steven Benton, Mike Lesk, Nicholas Negroponte, and Chris Schmandt. None of these people, of course, are responsible in any way for the ways I have misunderstood their ideas or abused their work, or for my failure to heed their advice. Finally I must specially mention Nicholas Negroponte, who created the Media Lab where this work was possible, and Chris Schmandt, the director of the Speech Research Group, who has been directly involved in everything I have done here at the Media Lab. A full list of his contributions to my studies would require many pages.

Those I have forgotten or otherwise neglected, I can only ask to deepen my debt by forgiving my shoddy memory and inexpressive words.

This thesis is dedicated to my son, Adam, may he find his way back soon.

Contents

List of Figures

List of Tables

frontispiece

MISTER BOFFO by Joe Martin

Chapter 1

Overview

This thesis is about the design and construction of a machine that does something difficult and useful in a new way. The machine's purpose is help people find their way by car from one place to another within a city, a task which is clearly useful and also worth improvement. A study done for the United States Federal Highway Administration estimated that 45 billion dollars are wasted each year in the U.S. because of ineffective routing, from causes including being lost, stuck in traffic, or choosing bad routes[1]. The machine discussed here is called the **Back Seat Driver**[1]. The Back Seat Driver is a computer program which uses synthetic speech to give instructions to the driver of a car as needed while driving. The Back Seat Driver differs from previous navigation assistance programs by using speech to give directions[2], instead of drawing a map or displaying symbols.

My two concerns are to determine the best form and content for spoken instructions, and to determine what information a program requires in its map to find routes and provide excellent instructions. My approach towards both questions is

[1]The usual sense of this term is an unwanted critic of one's driving skills. This is not what I intend.

[2]Although there are reports of earlier navigation systems using speech, none are described in the literature.

empirical. The initial design of the program was based on a short study of natural direction giving. This initial design was the "initial conditions" for an iterative process of design, testing with drivers, and revision. When the design was wrong, drivers complained, or just got lost, and I changed the design until they stopped complaining. This thesis describes the design that emerged from this process.

The thesis is divided into three parts. The first part, **Natural Directions**, describes the strategies and styles than people use when giving directions. The second part, **Implementation**, describes the program. Chapter 3 describes the map database. The next chapter tells how the Back Seat Driver decides what to say and when to say it. This is the key to the entire thesis. Chapter 5 tells about the other things the program does (such as read mail), and how it decides which task to work on at a given time. Chapter 6 tells how the Back Seat Driver evaluates routes when searching for the best one. The final part, **Conclusion**, describes related work, then tells how the Back Seat Driver might be improved and suggests directions for future research.

Three appendices provide background material and certain details of the implementation. The first two set out the technologies and concepts used in this program and in vehicle navigation in general. Depending on what you know already, you may have to read one or both of these section before reading earlier sections. The third appendix describes data communications using cellular telephones.

Why speech?

The first issue in design of a vehicle navigation system is the choice of modality for giving instructions. Two channels seem possible: vision and hearing[3]. Given the technical ability to display a map on the dashboard, or to display directional

[3]We could also consider the haptic channel. This is how people tell horses which way to go. It is unlikely to be popular with users, however.

arrows on the windshield, or the ability to synthesize speech, which should we choose?

Previous in-car navigation systems have used the visual channel to give navigation information. The Back Seat Driver uses speech because there are reasons not to use vision, and some advantages to speech.

Much work has been devoted to developing systems that display the vehicle's position on a map in the car (or at a dispatching facility). When combined with the ability to display a route (or at least the relative or absolute position of the destination), such displays can be used for navigation. One argument against such displays is that they require that the driver look at them while driving, and this makes driving less safe. Although drivers can spare some visual attention while driving along straight roads[88], it is not as clear that they can afford to look at the map while turning, yet it is just while turning that drivers are most in need of navigation information. Visual displays are most easily used when they are least needed.

A second argument against the use of maps is that many people have difficulty finding and following routes on paper maps[81]. That is, they are not "map literate". Information presented on maps will simply be unintelligible to such people. Paper maps do not really qualify as navigation assistance, since they do not show the driver's current position, though they may show the route.

An experiment comparing navigation aids was conducted by Streeter [82], who compared the performance of drivers under three different conditions. One set of drivers received customized paper maps with the route to be driven highlighted in red. A second set of drivers heard spoken instructions from a tape player that permitted them to play the next or previous instruction. Drivers using the maps took longer to arrive than those who had verbal instructions. They also made more errors and drove further. This can be explained by the extra effort required to consult the paper map. The third set of drivers got both navigational aids.

Surprisingly, drivers with *both* sources of information did *worse* than those using only voice, though better than those with a map only.

Neither navigation aid in the Streeter experiment included information about the current position. Both required the driver to determine when to carry out the instruction and to decide whether the instruction was correctly executed. That is, the driver could play an instruction as often as she liked, but had to decide for herself when to advance to the next instruction. The experiment did not compare voice to either an electronic map (indicating current position as well as route) or directional symbols (indicating which way to go). For this reason, the experiment does not directly answer the question about choice of modality. It is, at best, suggestive.

One advantage of speech is that the driver's eyes are left free for driving. In addition, speech uses words, and can therefore refer to past and future actions and objects not yet seen. This is hard to do with symbolic displays or maps.

On the other hand, speech has some problems as well. Speech is transient. The driver must remember what the program said, and the program must be prepared to repeat itself on demand. A second consequence of the ephemeral nature of speech is that the driver has no evidence of the program's operation except when it speaks. In a period of long silence, the driver may fear that the program has failed[4]. The program must do extra work to keep the driver confident in its continued operation. This turns out to be beneficial. Some of the remarks the program makes have little to do with route following per se, but rather are descriptions of the immediate vicinity of the road. When these are uttered at the right time, the driver gets a very strong sense that the program is seeing the world in the same way she does, and this is very reassuring.

The arguments here against maps and for speech convinced me that the Back

[4]Programs – especially prototypes – are not as reliable as, say, telephones. No one worries that a silent phone may be broken. A program silent for too long, though, is cause for concern.

Seat Driver should speak its directions. The next section tells how people speak their directions.

Part I

Natural Directions

Chapter 2

Human Direction Giving

My investigations began by studying how people give directions when they are passengers in the car. The intent was to discover common patterns in the directions that could be duplicated by the Back Seat Driver. The number of subjects (six) was far too small to justify any general conclusions about direction giving, but that was not the intention. I studied human direction giving to get a starting point for the design of the Back Seat Driver, not an ending point. I built the Back Seat Driver through an iterative process of design, test, and modification which converged to the system described in this thesis. All I needed was a good first approximation, and a sense of the kinds of variation in direction giving style, and for this I think six subjects was sufficient.

Procedure

My six subjects told me how to drive to destinations of their choosing while riding in the car. All were experienced drivers. The subjects actually wanted to go to their destination, so they had an incentive to give good directions. (They did not all provide the best routes, but route finding was not a subject of this investigation.) Most, but not all, of their utterances were spontaneous. In some cases I asked questions - "What now?" or "How long will I be on this road?". Our conversations were recorded on a cassette recorder in the car. All talk relevant to the route was transcribed, but most personal conversation was not transcribed. The transcriptions included approximate position of the car, as determined from memory. A sample transcription appears at the end of this chapter.

There are limits to the usefulness of this method. By its very nature, there is no way to control for the route or the destination. No two trips led to the same destination, though several had common beginnings. I assume that there is nothing special about the routes, so I can generalize. The usefulness of this data depends upon the assumption that the subjects were competent to give instructions and that the instructions were in the same form that subjects would have wanted to hear, had they been the driver. There is certainly some doubt about this point. We could do better than to emulate subjects who hesitated, gave misleading directions, or simply pointed[1]. Another limitation is that the transcriptions show position only coarsely, and velocity not at all, so they can not be used to answer questions of timing.

Overview

My understanding of driving instructions comes from treating driving as decision making. I think of the driver as constantly aiming the car at a moving target, a patch of pavement some few yards ahead. The driver is making a new decision several times a second. Some of these decisions involve choosing the next street to go along. When following instructions, at each point of decision one of two con-

[1]In fact, some navigation systems do no more than point in the right direction.

ditions must apply: either the instructions must say explicitly what to do, or the driver and instruction-giver must mutually believe that the thing to do is obvious and does not need to be said. This is by far the more common condition. It is only at an intersection that there is any meaningful choice, and even there there is a presumption that drivers will continue forward. Giving instructions should be the exception, not the rule. The traditional "back seat driver" - the annoying critic, not the program I built - bothers the driver by giving frequent warnings about conditions the driver is perfectly aware of, acting as if the driver had no common sense.

Navigators must give the driver the information she needs at the time she needs it. At the most basic level, this information consists of **instructions**. Instructions concern what to do and when (or where) to do it. People also provide **advice** which concerns unseen conditions ahead that the instruction giver can anticipate based on experience. The driver can ignore advice, but it may then be harder to follow the instructions. There is also **orientation** information, describing the surroundings, and placing the route in a larger image of the city. Instructions are essential; advice helpful; and orientation optional. Although I think orientation is important for future investigation, I do not address it in this study of natural direction giving.

2.1 Instructions

In the instructions I recorded, people used many different verbs to describe motion through intersections. I believe that people choose particular verbs to help describe the shape of the intersection and the kind of movement through it. One object of this study is to identify the reasons people prefer one verb to another, and the syntactic constructions in which verbs are employed. In addition to the kind of motion, instructions must also specify the direction of motion. This they do either with the words "left" and "right" or by naming landmarks in the desired direction, or both.

Now let us examine some of the verbs people use.

2.1.1 Verbs

People use a variety of verbs to describe motion through the streets. Table 2.1 lists the ten most common verbs used in directions, in descending order of frequency in my data.

verb	count	verb	count
take	35	turn	4
bear	22	follow	4
go	22	make	3
keep	13	get	3
stay	8	continue	2

Table 2.1: Verbs in descending order of frequency

On what grounds do people choose one verb over another? It would be naive to expect to find a single meaning for each verb. The verb "go" is used in several different contexts – "go straight" and "keep going" and "go right". There is almost no situation where the verb "go" is not used. Fortunately, some other verbs have more restrictive contexts. In this study, I concentrate on the more specific verbs, to the exclusion of the more "generic" ones, on the grounds that automated directions should use the most specific verb that is still correct. The goal is not to reproduce natural speech, with all its complexity. I am not trying to simulate human behavior, I am trying to impart information concisely.

The verb "take" designates a turn. To turn is to change heading by more than (about) 45 degrees, at an intersection where there are always at least two possible ways to go (though not necessarily both legal). There are two sizes of turns, "hard" turns (more than 90 degrees) and ordinary turns. After a turn, the car is on a "different" street than it was before.

Although the data does not show when two streets are the "same" and when they are "different", it does let us rule out some possible definitions. For instance, it is clear that change of name alone is not sufficient to make a street "different". There are plenty of streets that "change names" at an intersection. For example, if one drives up Ames Street from the Media Laboratory, and crosses Main Street in Kendall Square, the street name changes to Sixth Street. Certainly no one would call this a "turn", and nobody would say they were on a different street, either. Other examples are Hampshire Street in Cambridge, which is called Beacon Street in Somerville; and the O'Brien Highway in Cambridge, which is the McGrath Highway in Somerville (and which, by the way, I have never heard called anything other than McGrath). Aside from these counter-examples, name change can not possibly be the criteria for "different", because many drivers do not even *know* the names of the streets[2].

Change of heading is also not sufficient to make a turn. Near MIT there is an intersection where Fulkerson Street "turns" into Binney Street. An illustration of this place appears in figure 2-1. A driver proceeding up Fulkerson from Main has no choice about which way to go because the street is divided to prevent her from either continuing up Fulkerson or turning left onto Binney. Nobody calls this a turn, even though both angle and name change. Instead, the road seems to be just curving around to the right. This intersection is, in a way, a pun on the word "turn", which, in the phrase "Fulkerson turns into Binney", means "becomes". On the other hand, near Harvard Square there is a place where Appleton Street makes a right hand turn, and the subject whose route passed through this intersection described it as a turn, though the name was the same.

[2]But it will turn out that that the program does have to use names as part of its concept of "different", for reasons explained later.

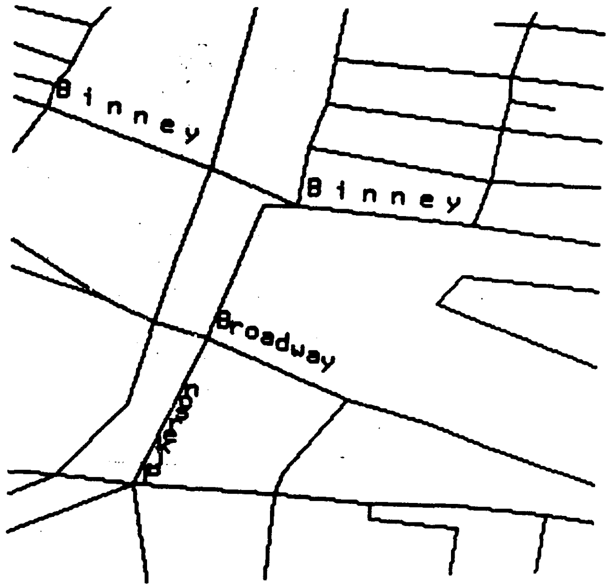


Figure 2-1: Fulkerson turns into Binney

A necessary feature for a "turn" is that there be choice about which way to go, even if the other choice is illegal.

The usual syntactic form for "take" is "take a DIRECTION", where DIRECTION is either "left" or "right". Other verbs with the same meaning as "take", but used less often, are "make", "hang", and "turn". The first two have exactly the same syntax as "take".

To "bear" is to make a lesser change of heading than a turn. After bearing, one may or may not be on a different road. The most common construction is "bear DIRECTION", with variant forms "bear to the DIRECTION" and "bear off to the DIRECTION". In order to "bear" the change in heading must be small and there must also be at least one other road which also requires only a small change in heading. Nobody used the word "bear" to describe the move from Albany Street to Main Street (a change of about 30 degrees). So using the word "bear" not only tells the driver how much turning will be required, it tells something about how the intersection will look. There is an exception. In one case, a subject used the word "bear" to describe motion when there was no choice at all, saying "bear left as the road goes around".

The main use of "keep" is to continue forward motion on the same street[3]. When subjects used "keep" at forks, it was always to select the straighter of the two alternatives. Verbs "stay" and "continue" were similar.

2.1.2 Direction

Verbs tell about the kind of motion. Other words tell the direction of the motion. In almost all cases, it is sufficient to say either "left" or "right". In some cases, there will be more than one street on the same side, and some other tactic is required.

One is to name a landmark that lies in that direction, e.g. "you bear to the left here and go under the bridge" or "straight to where those lights are.". This tactic can also be used when there is no ambiguity, and has the advantage that it also works for people who confuse left and right. In addition, combining a landmark with a direction name ("Go left towards the blinking lights.") adds redundancy to the instructions, thus making them easier to follow. Redundancy is a feature worth incorporating in the Back Seat Driver.

Other approaches to ambiguity are less successful. One subject tried to specify the street using "clock face" terms, where 12 o'clock is straight ahead, 3 o'clock is right, and 9 o'clock is left: "bear to the left about ten o clock" but when this did not work, he simply pointed. Another ordered the streets by amount of turning: "Now here you want to bear left. *Not all the way left* but straight to where those lights are". Note that this specification of directions also uses a landmark.

[3]This surprised me. In my own speech it is synonymous with "bear".

2.2 Distance and Time

In addition to knowing what to do, the driver must know where (or when) to do it. Notably absent from the directions are units of distance. Subjects *could* have given instructions by saying "Drive 310 yards, then ...", but did not. When subjects did give distances, they used qualitative terms like "in a while", "up ahead", "soon", or "not too long after ...". This is not simply because subjects are ignorant of the distances. When I asked them how long we would remain on the current road, most answered with units of either miles or minutes, and the answers were reasonably accurate. Subjects did have some idea of distance, but instead chose other means to tell the driver when to act.

The most common strategy is to give the instruction **immediately** before the action, just early enough that the driver has time to slow down for the turn[4]. Instructions given immediately often include a reference to "here" or "right here" as the place to act.

I think that subjects use this form in part because they are not able to formulate the instruction until they are actually at the place of choice. In Benjamin Kuipers' model of navigation ([46, 47], and discussed in 8.5) the earliest form of route knowledge is a "felt path" where a route is a sequence of pairs of scene description and action. The navigator can not give the next instruction until actually at the place described. Suggestive evidence for this account is that some drivers broke up instructions into two parts, first telling how to get to the next choice point, and then giving the next instruction only when at the point, as shown in this excerpt from a transcription[5].

[4]This is an assumption, not a fact. I did not measure the relationship between vehicle speed, distance to the intersection, and the time when the passengers spoke. Nor do I have evidence that passengers could reliably estimate these quantities. The subjects had all driven for at least five years. It would be instructive to hear the instructions given by those who have never driven.

[5]The transcription conventions are explained at length below. Briefly, punctuation represents features of spoken delivery, not grammatical form, so sentences do not generally end with periods.

Just keep going until () I think it's until the next major intersection
at least it's the next light anyway
Take a left at the light

Further evidence is that subjects sometimes changed their instructions when approaching the intersection:

Now we're gonna pass Harvard Ave and it's gonna be the next (no)
maybe we're gonna turn onto Harvard Ave come to think of it gonna
turn right () on Harvard

To make sense of this, assume that people are giving instructions according to a deficient mental map. When they see the actual intersection, they are able to recall the correct instructions.

Subjects who do have an accurate mental map can give the instruction at the earliest moment when it is unambiguous, that is, when the turn is the next turn, no matter how far away it is. They can also count ahead, and express the distance in units of blocks, turns, or lights ("take the second right"). This gives the driver plenty of notice without requiring much extra work. But such a count can be ambiguous if some of the objects counted are uncertain instances of the category. A small alley or a blinking light might or might not be included in the count. Either for this reason, or lack of knowledge, subjects did not make much use of counting.

One place where it is useful to give distance by a measure is in going around rotaries. In the one case where a route included a rotary, the subject said "You're gonna go around three quarters of the way and head across the bridge". In a rotary, the only appropriate units are those of angular distance, since there may not be signs, and the exits come up quickly.

Second, the empty parenthesis "()" represents a brief pause.

2.2.1 Landmarks

Subjects can tell the driver when to act by naming a landmark at the place of action. Perhaps the best example of use of landmarks is in an instruction for the turn from Massachusetts Avenue onto Back Street, just after crossing the Harvard Bridge into Back Bay, Boston:

> At the very end of the bridge here there's um a v- hard right which is h- very hard to see uh you want to take it umm it's like right beyond one of those jersey barriers you want to go in there behind this building where this taxi is coming out

This instruction includes four separate descriptions of the *position* of the street ("end of the bridge", "beyond the barriers", "behind this building", and "where this taxi is"); a description of the relative angle of the turn ("hard right") and a description of the street itself ("hard to see"), which may also warn the driver to devote extra effort to finding it[6]. It is not clear why the subject mentioned the angle of the turn, since there is only one right turn at that place, unless it was either to warn the driver to go extra slowly (for the sharp turn) or because it is hard to see.

One commonly used landmark is the name of the street. Street names make up about one quarter of all landmarks. A street name is not a good clue for when to turn, because signs are hard to see, even in the day light, even when they are present and pointing in the right direction. Nevertheless, drivers do use them. A possible alternative reason for providing a name is to help the driver learn about the city.

Other, equally commonly used landmarks are traffic lights and stop signs.

[6]Note also the use of "very" and "right" to mean "immediate".

These are especially useful because the driver is actively looking for them anyway, simply from a desire to avoid accidents.

Notably absent are "famous" landmarks. It is possible to name directions with reference to features such as those named by Kevin Lynch[50] in his classic book *The Image of the City.* These features include widely visible landmarks, nodes (major concentrations of flow or activity), and districts (visually distinct areas within the city). Some subjects did name prominent cultural and geographical features along the route, but only as background, not to tell me when to turn. The landmarks that people did use are what Lynch calls "local landmarks", having meaning only in an immediate context.

2.2.2 Advance Notice

Instructions for an act can be given more than once. One subject gave instructions twice, first in a general form well in advance of the action ("So we're gonna be following Commonwealth for a while and in maybe a mile it bears off to the left and we'll follow it to the left.") and then again at the time of execution: ("it bears left here").

Advance instructions may refer to the same landmark more than once. The way that people talk about a landmark depends upon its proximity. Subjects indicate the distance to a landmark implicitly by how they refer to it. While approaching an intersection, the landmarks may not be visible (in this case, the traffic lights are around a curve).

> There's a set of lights right up here (gonna go) straight through them and bear to the right

The subject uses an indefinite article since the objects are not visible. (Note also that two instructions are combined into one utterance.) After going straight through the lights, the next landmarks are now visible in the distance:

> **Bear right** at that gas station and blinking lights

The subject uses a distal deictic ("that"). Later, when close to the lights, he used a proximal deictic ("these"):

> at these flashing lights you'll bear right

2.3 Advice

Cooperative subjects give extra information to make following the route faster and safer. Lane advice tells the driver which lane to drive in, when driving on a road with more than one lane. Lane advice can be expressed as positive ("get in the left lane" or "stay in the left lane") to prepare for a turn which can only be made from the selected lane, or as a negative ("make sure you don't get caught too firmly in the left lane") to avoid a lane blocked by traffic waiting to turn or by parked cars, or reserved for turns.

Other advice includes warnings about speed traps (clearly a service that the Back Seat Driver should also provide) and about foolish pedestrians (desirable, but difficult to automate). Other advice is possible. For a discussion of advice about traffic conditions, see chapter 7.

2.4 Style and Packaging

The information that navigators supply includes instructions, advice, and orientation. Above these three kinds of information is a level of **style**.

People gave instructions in several different ways. Most used simple imperative sentences ("Take a right", "Keep going") while a few used future tense, either second person ("You're gonna take a right") or first plural ("We're gonna turn right"). Still others used an indirect style, talking about the driver's "wants" ("You're gonna wanna take a right"). I think this style is more polite, by speaking as if the goal of getting some place belonged to the driver, not the instruction giver, that is, it's not that the instruction giver is giving orders so as to get to her destination, it's that the instruction giver is providing the driver with information that the driver needs. This style was often combined with a description of the road ("The road forks up ahead."), which justified the choice.

Subjects were fairly consistent in style, but I have no idea why they preferred one form to another. I believe that stylistic choices have more to do with interpersonal relations than with the essentials of direction giving, and interpersonal factors are not accessible through the method of investigation employed in this stuidy. It is essential to this method that people complain about what they dislike. People will complain if instructions are wrong, or even unclear, but they might not complain if they dislike the interpersonal message they think they are getting. (Communication on that level is rarely explicit.) Moreover, I do not believe that my subjects took the computer as a "person", so there were no interpersonal factors. In the section on "mistakes" I describe one way that the Back Seat Driver tries to adopt a gentle style with drivers.

2.5 Silence

We should also consider what is *not* said. At most intersections the driver has a choice of directions, yet we do not find subjects giving instructions at every intersection. This can only mean that subjects assumed that there was a unique choice that was obvious, and that this choice was also obvious to the driver. This follows from H. P. Grice's maxim of QUANTITY[28]: "Do not make your contribution more informative than is required." Cooperative speakers speak only when required. If a subject is silent, the thing to do must be obvious. Unfortunately, the inverse is not true. That a subject spoke is not evidence that the action described was not obvious. Some subjects give explicit instructions at places where there is no choice whatsoever for example, on one-way streets. Speaking might also serve to reassure the driver of the subject's attention and competence. When we find some subjects speaking, where others are silent, we can guess that the act was not wholly obvious, but we cannot be sure.

The obvious thing is usually to stay on the same road. This may be less obvious when staying on the road requires crossing a bigger road, since subjects sometimes spoke in this situation ("Go straight through the lights"). This follows from the usual pattern for cross-town routes: to go from local streets to progressively larger streets (collectors, then arterials), and then back again to small streets when near the destination. Subjects might want to override a presumption of turning onto the first collector encountered.

At some forks, one branch is the obvious next branch. An example is the connection between Memorial Drive, Brookline Street, and the Boston University Bridge. The right branch (going either up or down the Charles River) leads to a rotary from which one may turn onto Brookline Street or the bridge. The left branch leads up and over the rotary. Most (not all) of those whose routes stayed on Memorial Drive passed through this fork without comment. The left branch is the obvious place to go because it has two lanes and is more straight. An example of a fork without an obvious branch is the connection between Memorial Drive and Massachusetts Avenue, near MIT. Here the left branch leads underneath Massachusetts Avenue, and the right branch forces a turn. Even though the left branch is wider than the right, it is not obvious, perhaps because it departs at a steeper angle than the right branch does. Subjects always were explicit about which branch to take.

2.6 Example Instructions

Here is one of the transcriptions, and a map showing the route. I have adapted the notation system of Gail Jefferson [72]. This notation system is specialized for study of interaction between speakers. Words are spelled the way they sound, not the way they would be written, so, e.g. "going to" is written as "gonna", when so pronounced. The notation is explained in table 2.2. I have chosen to capitalize proper nouns to make the account easier to read.

symbol	meaning
curly braces	indicate location
–	hesitation or cut-off speech
L:	the driver is speaking
R:	the passenger is speaking
()	an untimed pause
single parenthesis	enclose uncertain words
double parenthesis	vocal style or non-vocal sound
left bracket	marks simultaneous speech
colon	marks extended length syllable

Table 2.2: Notation system for transcriptions

This route goes from the parking garage at 12 Albany Street, Cambridge, to a garage at Glenville Terrace in Allston.

```
   {at 12 Albany}
1 R: and at the stop sign take a right on Main
     {on Main Street}
2 R: and you'll keep going straight
3 R: this is Tech Square
4 L: mhhmm
5 R: MIT AI Lab is to your left and behind you
```

6 L: hmm

7 R: so keep going straight past Ames Street

8 R: dont hit the pedestrians

9 L: ((in silly voice)) We want to die::

10 R: (keep) going straight

11 R: merge here I believe with Broadway

12 Go straight at the stop sign

13 and then you're going to take a right on Memorial

14 (at the) sign that says Memorial Drive West

{on Memorial}

15 you can see uh that you're driving right along the river and

16 Boston is on the other side on the left

17 L: mhmm

18 L: what part of Cambridge are we in

19 R: we're in Kendall Square () area ()

20 and basically we're passing by the uh passing by MIT the long way

21 much of it is right on Memorial Drive

22 L: mhmm

23 L: how long will I be on this street

24 R: oh about another () mile mile and a half

25 L: ((unclear))

{approaching Massachusetts Avenue fork}

26 R: you bear to the left here and go under the bridge

27 under the overpass I guess

28 L: uhhuh

((noise of tires on pavement))

29 R: down the ways a bit we're gonna cross over the river to our left

30 umm cross a bridge called the B U Bridge

31 but you have to bear off to the right and circle around

32 R: you're gonna want to get into your right lane
33 R: and you're gonna bear off () to the right here ()
34 where it says route 2 ()
35 and this is a traffic circle
36 you're gonna go around three quarters of the way
37 and head across the bridge
38 R: stay in the right lane here cause you're gonna take a right
39 L: uh huh
40 R: so we're leaving Cambridge and going into Boston now
41 and uh Boston University is right around here
42 right along the river
43 L: mhmm
44 R: and we'll be driving past () some of BUs buildings
45 R: you're gonna take the first right here on Commonwealth () Ave
46 R: so we're gonna be following Commonwealth () for a while
47 an:d in maybe a mile it bears off to the left
48 and we'll we'll follow it to the left
{Commonwealth and Babcock}
49 R: might need to get into the left lane for this uh bearing left
50 ()
51 in fact it bears left here
52 ()
53 middle or left lane
54 R: So I guess a good thing to remember for the directions here
55 is you follow the Green line around (when) you bear left
56 R: now we're gonna pass Harvard Ave
57 and its gonna be the next
58 (no) maybe we're gonna turn onto Harvard Ave
59 come to think of it

60 gonna turn right () on Harvard
61 an:d that it be coming up right here?
62 yes its the first right then after you bear left
{Commonwealth and Harvard 6:01}
63 R: okay now we're gonna turn right on the first street
64 which I believe is (glenough)
65 and this is our destination ()
66 towards the end of the street
67 L: here's Glenville on the left
[
68 R: Glen- Glenville
69 L: This street here?
70 R: yes
71 L: Glenville Terrace
72 R: Glenville Terrace

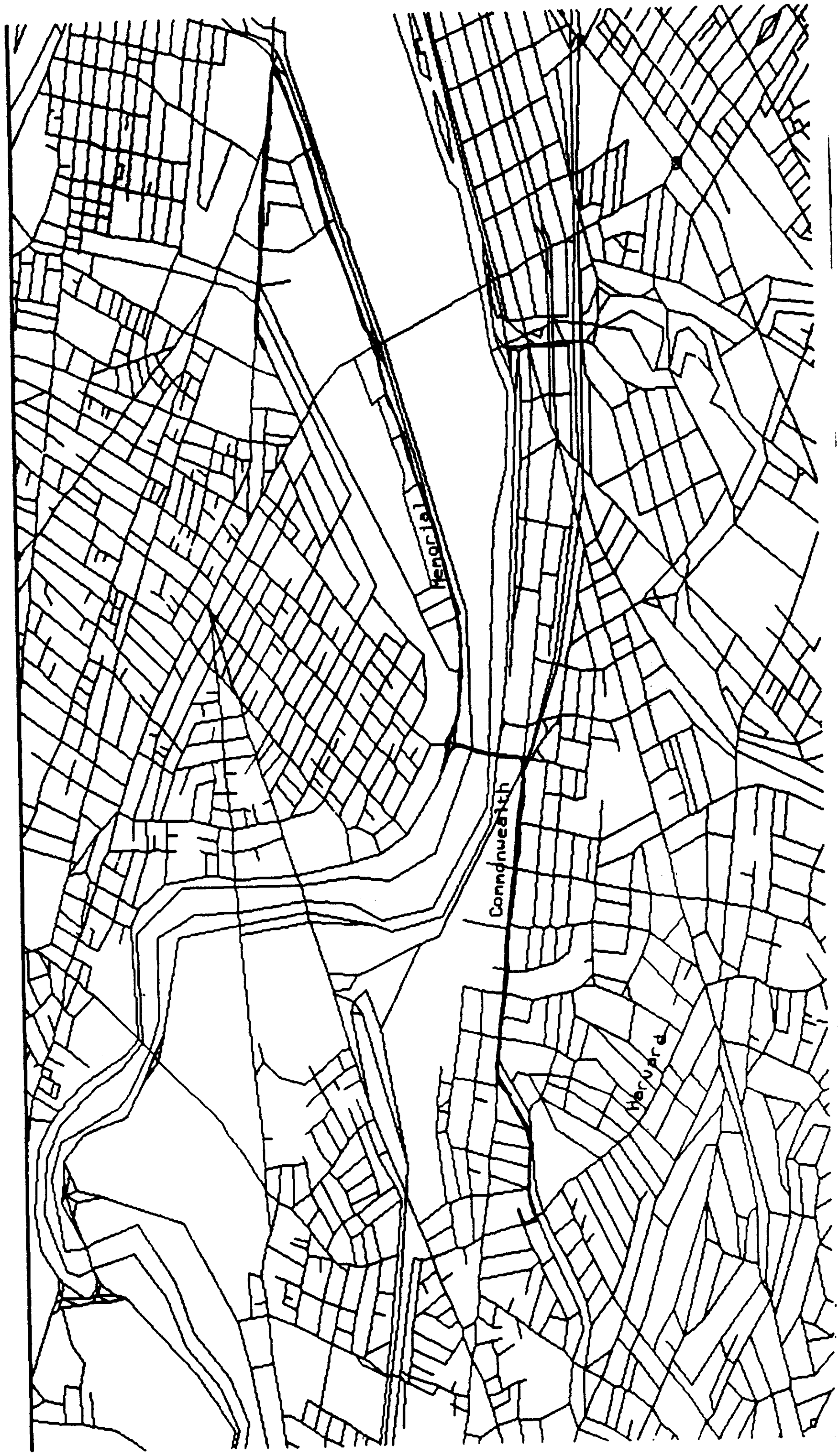


Figure 2-2: Map for sample route

Part II

Implementation of the Back Seat Driver

Chapter 3

Cartography

Every navigation system requires a street map. The Back Seat Driver's map originated as a DIME (Dual Independent Map Encoding) file, a map format invented by the US Census Bureau for the 1980 census. Although DIME maps are a useful beginning for navigation, there are problems inherent in the structure which make them unsuited for route finding and route description. One aim of the research here is to determine what information must be in a map database used for these purposes.

The first section of this chapter describes the DIME map format. The second section shows why DIME is not sufficient for route finding or route description. The third section describes the extensions to the map required for the Back Seat Driver, and the final section compares these extensions with those in other navigation systems.

3.1 DIME format

An important point, sometimes overlooked because it seems obvious, is that the map should be a vector representation, not a raster picture. One reason is economy of storage: the vector representation takes far less room. But the main reason is that only a vector representation is suitable for programmed manipulation. The mathematical basis for route finding is graph theory, not tracing lines on a picture. Even when the intended application is simply to show a picture, the vector representation allows a program to display the map at a variety of scales, detail, and orientation, to highlight significant information and remove unimportant features.

The basic unit of the DIME file is the *segment*. A segment is a portion of a street (or other linear feature such as a railroad, property line, or shoreline) chosen to be small enough that it is a straight line and has no intersection with any other segment except at its endpoints. The two endpoints are designated `FROM` and `TO`. If the segment is a street segment (as opposed to, say, a railroad) and has addresses on it, then the `FROM` endpoint is the one with the lowest address. Otherwise, the endpoint labels are chosen arbitrarily. A segment has two sides, left and right. The sides are chosen with respect to travel from the `FROM` endpoint to the `TO` endpoint.

Attributes of a segment include:

- its name (40 characters)
- its "type" (a one to four character abbreviation such as "ST")
- longitude and latitude of the endpoints
- ZIP code for each side
- addresses for each endpoint and each side
- list of segments connected at each endpoint

A program using DIME can find the location of an address along the segment by interpolating the addresses between the low and high addresses for the two endpoints.

3.2 The limits of DIME

The form and content of a map database depend upon the purpose for which you intend to use it. The DIME file was invented to allow the Census Bureau to determine the proper census tract for any given address in the USA, and for no other purpose. The DIME file is well suited to determining the absolute position of a building from its street address. But the DIME database is not sufficient for route finding, and it only marginal for generating route descriptions. The problems with DIME fall into two categories:

3.2.1 The model of connection is deficient

The DIME file indicates that two segments are *physically* connected (that is, they touch), but not whether they are *legally* connected (i.e. it is legal to travel from one to the other). Legal connectivity is crucial for route finding, unless we only want to go to jail or the hospital. It must be explicit. Legal connectivity does not replace physical connectivity. Only legal connectivity is needed for route finding, but route description requires information about physical connections as well. Physical connectivity also affects route finding directly when seeking the simplest route, since ease of description is determined in part by physical connectivity.

The DIME file is a planar graph. This means that no two segments can cross except at an intersection, so there is no way to correctly represent, say, an overpass. The DIME format represents an overpass by "breaking" both streets at the point where they cross, and creating a fictitious intersection even though the segments do not touch in reality. These false intersections are particularly troublesome since DIME does not represent legal connectivity, so it appears possible and legal for a car to jump straight up and turn onto the overpass.

3.2.2 Position resolution is inadequate

Coordinates in DIME are stored in ten thousandths of a degree. This means that the position of an endpoint in the map differs from the true position by as much as 6.5 meters in latitude (north/south) and 5 meters in longitude at the latitude of Boston. (The size of a degree of longitude depends upon latitude, since longitude lines converge at the poles.) This inherent position error causes problems because it introduces error in length and in heading. See figure 3-1 for an illustration. Here, the points A and B could have been located anywhere within

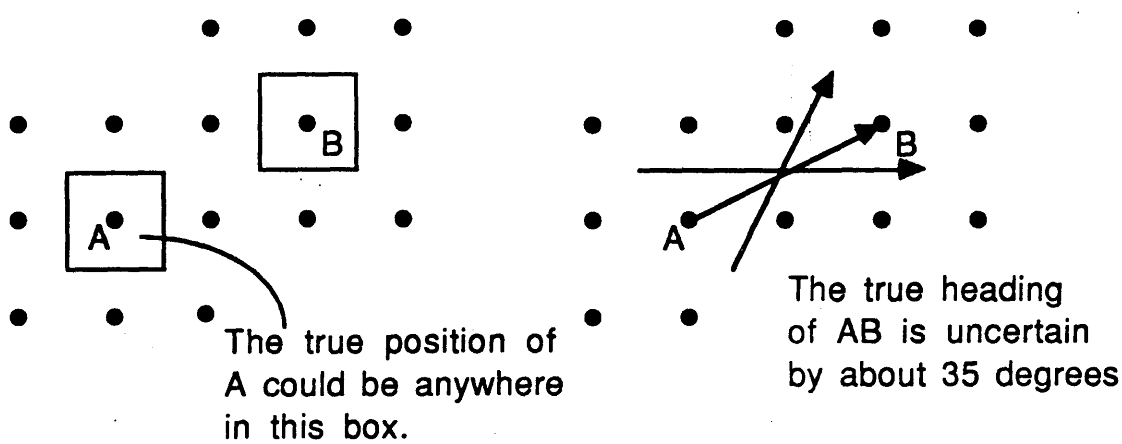


Figure 3-1: Position roundoff makes headings uncertain

the boxes surrounding them. Depending where A and B really are, the heading of line AB changes by almost 90 degrees. Figure 3-2 shows how the accuracy in heading decreases as segments become shorter.

Uncertainty in heading causes uncertainty in the angle between two segments. A straight street can appear to wobble if it is made of many short segments. See for instance Pleasant Street in figure 3-3. The dots in this figure are the possible coordinate locations. Pleasant Street lies in between two columns, so it wobbles back and forth.

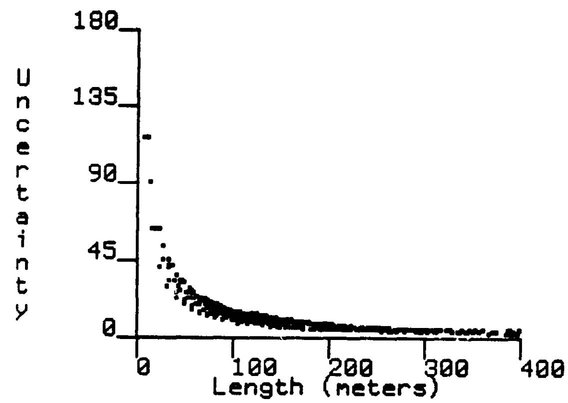


Figure 3-2: Short segments have less angular resolution

Segment wobble causes problems for the route finder, makes it hard to generate correct descriptions, and interferes with position determination. A straight road that "wobbles" will appear to be slower than one that does not because cars must slow down for turns. The Back Seat Driver corrects for this by assuming that the angle between two streets is the smallest possible value. This means it sometimes overestimates the speed it can travel through an intersection. Uncertainty in the angle of segments at an intersection also makes it difficult to describe the intersec- on correctly. After all, the difference between going straight and turning is "just a matter of degree". As a result, the Back Seat Driver sometimes uses the verb "bear" where "turn" would be more appropriate, because it underestimates the angle of the intersection. Segment wobble also interferes with navigation because it makes it difficult to compare compass headings with the heading of a street.

3.2.3 The forthcoming TIGER format is better, but not enough

The Census Bureau, in cooperation with the United States Geological Survey,

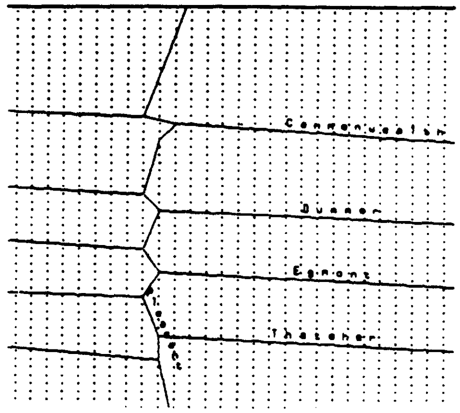


Figure 3-3: Coordinate rounding makes streets seem to wobble

has designed a new map format known as TIGER (Topologically Integrated Geographic Encoding and Referencing). This format has several improvements from the DIME format[79, 52, 42]. The map format is structured to permit automated consistency checking, so it should be more reliable[13, 54]. TIGER coordinates have two more decimal places of precision (bringing accuracy to something like four inches), and the segment representation includes "curve vectors" so that a segment need no longer be a straight line. Two dimensional areas are explicitly represented as closed polygons, which will make it much easier to tell whether a route crosses a river. The database includes point landmarks (fire towers, churches, schools) and "Key Geographic Locations" (mostly commercial buildings)[53] and this will make it easier to find locations and give directions. But the TIGER file is still a planar graph and has only physical connections, so it will not be sufficient for route finding. TIGER maps will be an important component of future navigation programs because they are more up to date and more accurate, but will still be insufficient. Those who wish to use TIGER files will have to obtain and represent connectivity information in some other way.

3.3 A better map

Implementing the Back Seat Driver required me to extend the map format. Each extension described here is used in one or more ways by the Back Seat Driver. Since I have not yet described all the features of the Back Seat Driver in detail, it may not be clear why certain extensions were required. I ask the reader's patience if certain decisions seem unmotivated.

3.3.1 Legal connectivity

The most significant addition is an explicit representation of legal connectivity. This is crucial for finding routes. At each endpoint of the segment is a list of all segments which are legally accessible from that endpoint. This list allows the route finder to consider only legal paths. Although this change is easy to implement, it does not seem to have been included in any other navigation system.

3.4 Additional segment attributes

The DIME file records a small amount of information about the segment. I found it useful to add additional attributes to the segment to make better descriptions. These new attributes are:

- street quality
- divided roads
- signs
- other landmarks
- traffic lights

- stop signs
- lane information
- speed limit

The first four were added for Direction Assistance, the remainder for the Back Seat Driver.

3.4.1 Quality combines size and speed of a segment

The street quality is a number from 1 ("super") to 4 ("bad") which combines the ease of locating and following the street and the expected rate of travel along it. Super streets are the access highways (e.g. the Massachusetts Turnpike, and most of Storrow Drive, but not Memorial Drive). Figure 3-4 shows the network of super streets. In general, the "super" streets are well known, though perhaps not by their correct name. The "bad" streets may also be well known, to local residents, but not by name. (For instance, see "Back Street" on page 31.) The street quality attribute affects the route finder and the route describer. The route finder ordinarily prefers the street of highest quality. The route describer uses the quality of a street in several ways, described below.

3.4.2 Expanded street classifications

It is useful to have a richer taxonomy of street types than that provided by DIME. There are five categories of streets: **ordinary**, **rotary**, **access ramp**, **underpass**, **tunnel**, and **bridge**. There are also four categories of non-streets: **railroad**, **water**, **alley** and **walkway**.

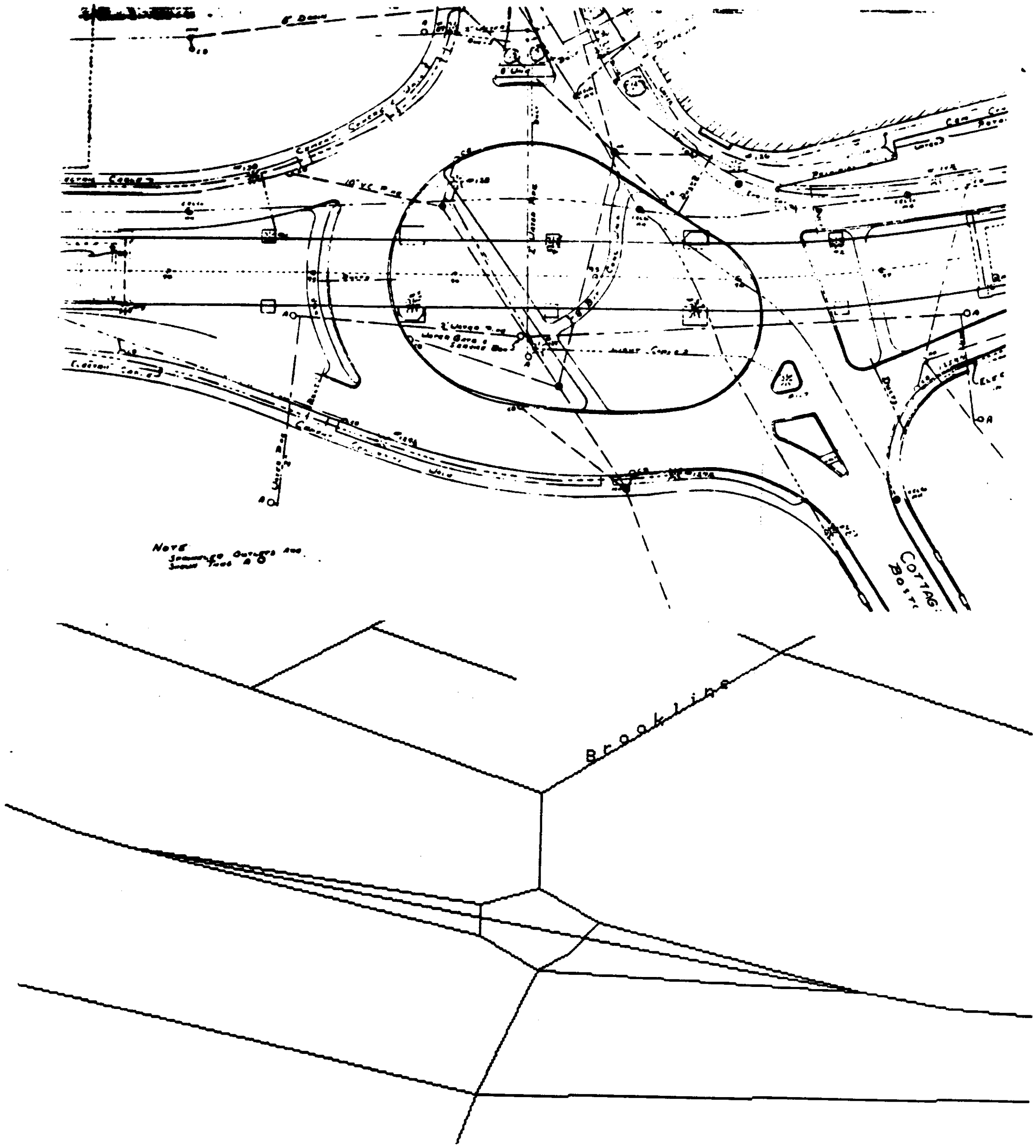
A rotary (or roundabout, or traffic circle) is a kind of intersection in the form of a one way circular road (figure 3-5). Traffic enters and exits the rotary on lines

Figure 3-4: The "super" streets of Boston

more or less tangent to the circle. Rotaries are efficient, since traffic can move continuously instead of alternating; and safer than ordinary intersections, since all traffic is moving the same direction, with a low relative velocity. On the other hand, rotaries require more space, and can be somewhat confusing. The Back Seat Driver needs an explicit indication of whether a segment is part of a rotary, since it would be too difficult to guess from the map geometry.

An access ramp is a nameless piece of road that leads to or away from another street. The prototypical access ramp is the highway entrance or exit, but access ramps can also link interchanges between ordinary streets. Access ramps are almost always one-way streets, and they have no address numbers. It is important to distinguish access ramps because they must be described differently than ordinary streets.

The classes underpass, tunnel, and bridge have obvious meanings. They are helpful in route descriptions because the driver is very likely to notice these features; and in forming names, since bridges and tunnels have singular names (*the* Harvard Bridge, *the* Sumner Tunnel).



Rotary as drawn by Metropolitan District Commission and as DIME file.

Figure 3-5: A rotary

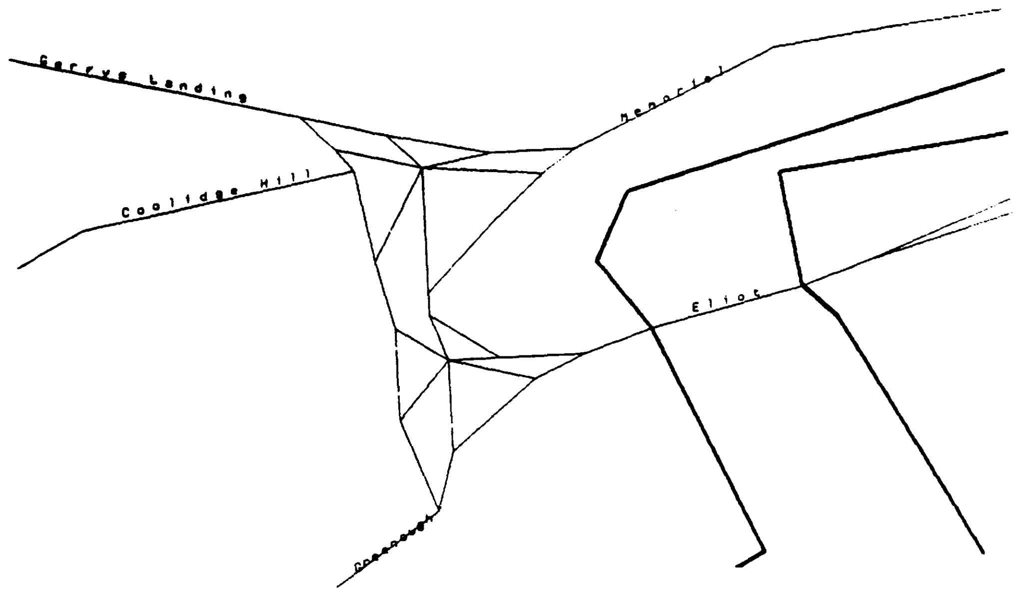


Figure 3-6: Access ramps at an interchange

3.4.3 Divided Roads

A divided road is one with a physical barrier preventing traffic from crossing the center line. Direction Assistance uses this attribute to avoid U Turns (Back Seat Driver only makes U Turns if there is no other alternative). The significance to the Back Seat Driver is that a divided road is safer than an undivided road. The route planner does not at present take this into account, but the module that gives warnings (page 90) does use it.

3.4.4 Landmarks help directions

It is very desirable to use landmarks in giving directions. Landmarks may be fixed in one place or they may move (e.g. "follow that car"). The Back Seat Driver can only use landmarks fixed permanently in place, since it has no eyes. This also means that it can only use landmarks when it can be sure that the driver can see them. This precludes the use of distant landmarks (e.g. a tall building at a distance) because it can not tell whether such a landmark is visible at any given place.

The most useful landmarks are traffic lights. There are two kinds of lights, three way ("stop") lights and blinking lights. Traffic lights are stored independently for each endpoint of each segment, since the presence of a light at one segment of an intersection does not imply that all others do.

Signs are another class of landmarks. These are especially useful on highways, where exits have both a number and a name. Highway signs are designed to be visible under all conditions and at great distances, so they are reliable landmarks. A sign or exit number is stored as a **connection cue**, which is a text string that gives a cue for moving from one segment to another. Every cue has a **type** which tells the kind of cue, e.g. **sign** or **exit-number**. There may be more than one connection cue for a given pair of segments, but there is never more than one of a type. A problem with this representation is that the information in the sign is stored as unstructured text. It is important that the Back Seat Driver understand what the sign says, not simply utter the words. There are two reasons for this. First, the Back Seat Driver's internal representation for text is based on syntactic structure, not text strings. Second, the objects mentioned in the signs (cities and roads) should be entered into the discourse model. They should become salient for future reference. This means that the text of a sign must be parsed, so that e.g. the sign text "Cambridge, Somerville, and Storrow Drive" should become a conjunction of the two cities "Cambridge" and "Somerville" and the street named "Storrow Drive". The Back Seat Driver parses sign text by separating it into tokens delimited by commas and the word "and", then attempts to recognize objects on the map (street names, cities, neighborhoods) from these tokens. When recognition fails, the token is not entered into the discourse model. When parsing fails, the spoken output has incorrect grammar.

Buildings are a third class of landmarks. The two types of buildings used at present are toll booths and gas stations. Toll booths and gas stations are stored in two different ways. Toll booths were added to the map database for Direction Assistance and are stored as connection cues. This is a bad representation for buildings for the same reasons that it is bad for signs, but is used for historical reasons. Gas stations are stored in the Yellow Pages file (page 88). This is better, but costly, as it requires searching the file for gas stations. A better approach would be to index gas stations (and other buildings) by street, as is used in TIGER (page 46).

3.4.5 Lane information

Roads often have more than one lane. Selecting the proper lane can make travel faster, and it may even be mandatory, since certain turns may only be possible from some lanes. For both directions on the segment is recorded the **number of lanes** and whether one or more lanes is reserved for **turn restrictions** – either left turn only or right turn only.

3.4.6 DIME files need correction

As distributed by the Census Bureau, the DIME file for Boston has many errors. Errors arise from imprecise surveying, inconsistent coding, clerical errors in data entry, and changes since the survey date. The map is especially bad for access ramps, which often have gross errors in angle, or are missing altogether. This is probably because the map makers had little incentive to be accurate with roads where no one lived, because there are no census data to be taken there.

Names are often inconsistent, either because of spelling errors or because a portion of the name is omitted (e.g. "North" in "North Beacon Street".) To use the Boston DIME file required reexamining the entire area in person. For the most part, this required no more than an hour for each square mile. More time was required for complicated intersections which had the kind of anonymous roads so poorly surveyed in the DIME maps. To enter these, I would make a sketch map on the spot, then enter the changes using a graphical editor. This allowed for reasonably quick data entry, but was not very accurate, since I used only my eyes to measure distance and angle.

I had edited much of the map earlier for Direction Assistance, but I also found that the Back Seat Driver requires more accuracy than Direction Assistance because the Back Seat Driver needs accurate positions to deliver messages at the right time. Also, people following written directions (as in Direction Assistance) rely more on their own intelligence to figure out when to act. If there is a discrepancy between the instructions and the world, they look around to try to understand it and correct it. But users of the Back Seat Driver just do what the machine tells them to do.

3.5 Other maps

Those who have built other automotive navigation systems have also found it necessary to add features to their maps. Here I describe features beyond those present in DIME.

Neukircher [58] describes the features of the map used in the EVA system. This map has better position information than DIME. Points are stored in three dimensions and are accurate to 2.5 meters. Road segments are straight lines, chosen so that a new segment begins at either an intersection or when the change in direction exceeds 30 degrees, or when the distance from the center line exceeds 5 meters. Additional attributes of the roads include height and weight restrictions, location of magnetic anomalies, warnings, landmarks, special objects useful in descriptions (e.g. underpasses), layout of complex intersections, and signs. The map has two levels of detail[65]. The coarse level is used for route finding, and the fine level has more detailed information for position finding. Route finding information includes *two* values for expected speed (one for normal conditions and a second for times of high density), the expected wait time at segment endpoints, and areas where children are likely to be playing.

The University of Calgary AVL-2000 system uses a map that originated as a Canadian government Area Master File. This format, similar to DIME, also required extensive augmentation[33]. Link (segment) attributes include distance, expected travel time, safety, scenic value, tolls, "impedience value" [sic], one way limitations, banned turns, road type (over- and under-pass, traffic circles, clover leaf), presence of meridians (divided?), and restricted areas. Harris describes as a "special problem" those "source or destination points which correspond to a street addresses [sic] which do not have a unique node identifier". Either their map representation cannot interpolate addresses along segments, or the route finder is restricted to finding routes to nodes only. Harris also mentions auxiliary road information including landmarks, points of interest, emergency services, commercial establishments, weather conditions, traffic flow, and road characteristics, and stresses the importance of being able to update the map database over a communications link while driving.

Most systems have expanded the classification of streets. The ETAK map classification is:

- Interstate highway
- Semi-limited Access Roads and State Highways
- Arterial
- Collector

- **Light Duty Roads**
- **Alleys or Unpaved Roads**
- High Speed Ramps
- Low Speed Ramps

This is a richer taxonomy than that of the Back Seat Driver, and is essential to ETAK because of its significance in choosing which roads to display (lesser roads are suppressed at larger scales to control detail) and in which colors to display them. The Back Seat Driver could also benefit from such a taxonomy. The EVA system has a two level taxonomy:

- rural
  - motorways and federal highways with separate directional lanes and without intersections
  - federal highways
  - roads wider than six meters
  - roads four to six meters wide
  - others
- urban
  - divided
  - through
  - main
  - side
  - restricted

These maps have some questionable design decisions on the representation of legal restrictions. The ETAK map has no legal topology at all. It is not intended for route finding. The EVA map apparently encodes restrictions on turning by signs, rather than directly in the network. The Calgary map represents legal topology (one ways, banned turns) as a link attribute instead of in the network topology. It may be that the street network represents only physical topology, with the assumption that legal topology will be equivalent to physical topology unless specially indicated. The Back Seat Driver appears to be unique in maintaining separate but equal representations for physical and legal topology. These two topologies should be integrated because legal topology is needed for route finding, and physical topology for route description.

Some other navigation systems attempt to give warnings about hazardous conditions. In those systems (EVA, Calgary) the hazards (about slope, width, or curves) are encoded explicitly into the map. It is unclear how "safety" (in the Calgary map) is represented, or whether it *should* be explicitly represented. A system which can compute safety from other attributes (traffic flow, slope, width) will be more general than one which relies on stored data. On the other hand, if safety is based on external information (accident statistics) then it belongs in the database.

3.6 Improvements are still needed

The Back Seat Driver's map, advanced as it is, could still be extended. Both route finder and descriptions would benefit from a more powerful map database. Desirable enhancements include:

- **Time dependent legal connectivity**. Sometimes a given turn will be prohibited only at certain hours of the day, typically rush hour. At present, Back Seat Driver must record such turns as always prohibited. This results in less than optimal routes. In Cambridge, Memorial Drive is closed on Sundays during the summer to create a large play space. The Back Seat Driver map representation has no way to record such arbitrary, yet predictable changes. In general, the Back Seat Driver needs the ability to change legal connectivity dynamically, since a street can be closed suddenly because of an accident or weather. This is more than just a database issue, so I discuss it in Chapter 8.

- **Expected rate of travel**. Rate of travel is at present taken to be a function of street "quality". This is probably a mistake. Although there is a correlation, travel rate should be a separate, explicit segment attribute. One reason is that travel rate, unlike quality, changes during the day. For example, the Central Artery in Boston is a "super" street, but is also very slow during rush hour. Either the Back Seat Driver should have a model of traffic flow like that of an experienced driver (i.e. it should know what "rush hour" means), or it should have some means of getting real time traffic conditions, perhaps broadcast by a central reporting agency. I discuss this further in Chapter 8, since I see it as more of a planning issue than a database issue.

- **Turn resistance**. Some turns, though legal, are also difficult to make (e.g. turning left across a large traffic flow at an intersection without a traffic light). The route finder should avoid these turns if possible. To an extent, the difficulty of a turn is implicit in the quality of the participating street segments, but an explicit model might prove useful. The TAXI! driving simulator[12] has the concept of turn resistance, which is a number from one to ten specifying the difficulty of making a given turn. This concept should be adopted.

- **Lane restrictions**. Some lanes or streets are restricted to certain kinds of traffic (car pools, no commercial vehicles). Including this information would make the Back Seat Driver useful for more kinds of traffic. An important subclass is **height restrictions**, since some underpasses are so low that tall vehicles will not fit under them. (The Memorial Drive underpass under Massachusetts Avenue is an infamous example. See figure 3-7.) The route finder should not attempt to send tall vehicles to their doom[1].

[1] I assume that the route finder knows about the height, weight, etc. of the vehicle in which it is installed.

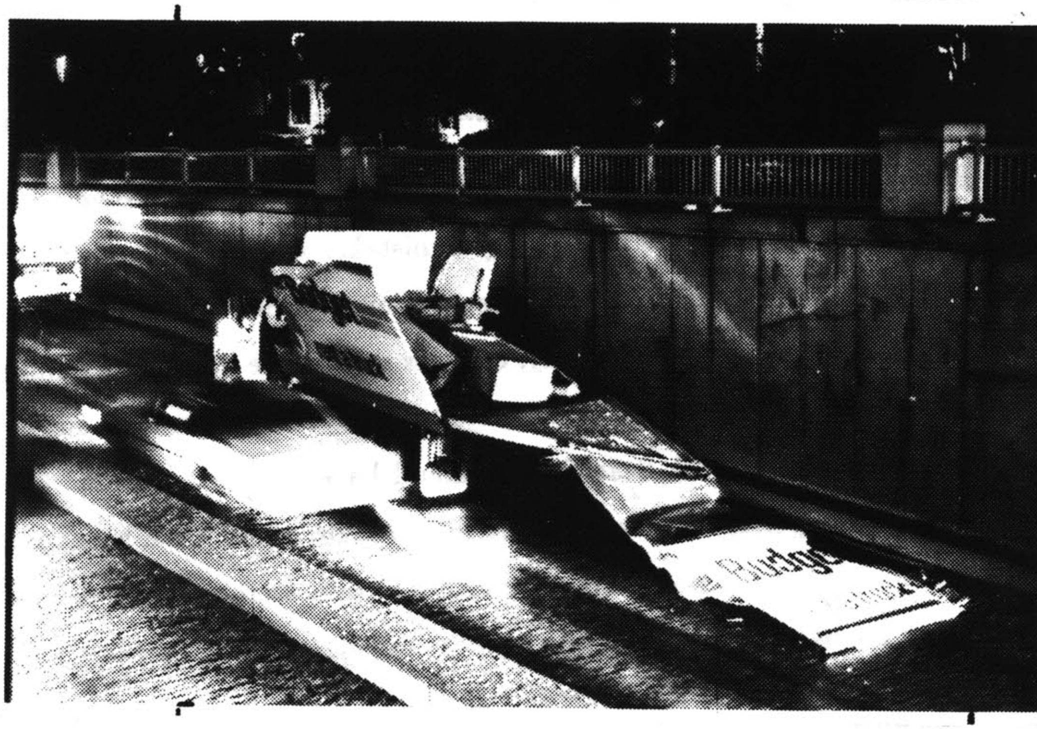


Figure 3-7: When trucks ignore height restrictions

- **Traffic light information**. At some lights it is permitted to make a right turn at a red light after a full stop. Right turns here will be no slower than right turns at a stop sign, so the route finder should prefer such intersections to those that do not permit it. Also, traffic lights have differing cycle lengths.

- **Altitude**. Points on street maps should be three dimensional. Route descriptions would be better given knowledge of the underlying topography ("Go up the hill, and take a left"). Stopping distance is affected by slope, so instructions must be given sooner when traveling down a hill. Slope affects safety. The route finder should avoid steep slopes in snowy weather. Knowledge of altitude provides a constraint for GPS position determination. Given exact knowledge of altitude, three satellites suffice for determining horizontal position. Less exact knowledge might still be useful. Finally, distance between points will be more accurate when change in altitude is considered.

- **At grade or not**. Roads designed for high speed may be more level than the underlying topography. They may be elevated or they may be depressed. A road which is not at grade will not have the slope of the land beneath it.

- **Local knowledge** is a catch-all term for facts about the road which can not be predicted from the map. These are facts about how people and institutions acts on or near the road; e.g. that a speed trap is here, or that this road is one of the first ones plowed after a snow storm.

3.6.1 Summary

A street map used for route finding must include a representation for legal connectivity. This is essential. To find the fastest routes, the map should also include features that affect speed of travel, including street quality, speed limit, traffic lights and stop signs. To generate directions, the map should include landmarks such as traffic lights and buildings, and additional descriptive information about the street segments, including street type, number of lanes, turn restrictions, street quality, and speed limit. The Back Seat Driver map includes all of these.

It would also be desirable to include other features, such as time dependent legal connectivity, and expected rate of travel along streets and across intersections. Positions should be stored in three dimensions, not two, and with sufficient accuracy that the headings of segments can be accurately determined from the map. These features should be added in future work.

Chapter 4

Descriptions

The two key issues in describing a route are deciding **what** to say and deciding **when** to say it. There is a tradeoff between these two factors. At one extreme are directions given completely in advance, with no control over when the driver reads them. A directions of this kind might be "Go half a mile, then take a left onto Mulberry Street". A driver following such an instruction must use the odometer to estimate distance or look for a street sign. The instruction itself does not say when to act. On the other extreme are instructions which rely totally on timing for success. Such an instruction might be: "Turn left *now.*".

The more the Back Seat Driver uses timing, the less the workload on the driver. Where possible, instructions should use timing as much as possible, and contain only as much description as necessary. The limiting factor is the accuracy of the information the Back Seat Driver has about position. For instance, consider a turn to the right where there are two roads to the right, one a sharper turn. If the Back Seat Driver has sufficient directional resolution to tell which way the car is facing (and if the map is correct) it can designate the correct road as the driver is turning. If the driver is headed the wrong way it can say "no, the other one".

The current Back Seat Driver does not have this kind of information, so it must *describe* the turn, saying either "a sharp right" or "an easy right".

The Back Seat Driver tries to minimize the driver's workload. It does not rely on the driver to measure distances. This is just as well, since many drivers do not even know what an odometer is. More important, the driver has a lot more to do than keep track of distance and heading – namely keeping the car on the road and avoiding collisions, not to mention listening to the radio, talking to someone, or just thinking. Minimizing the driver's workload means telling the driver when to act, not just what to do.

The next section tells how the Back Seat Driver forms its descriptions. The basic process is to look ahead at the route and decide what action the driver must take to follow it, then form sentences that tell the driver about these actions, uttering them at the appropriate times. Following sections take up some additional issues, namely giving appropriate advice while following the route, and handling mistakes when the driver does not follow the instructions.

4.1 Classifying movement

To the Back Seat Driver, a route is a sequence of street segments leading from the origin to the destination. Each connection from one segment to another is considered an "intersection", even if there is only one next segment at the intersection. At any moment, the car will be on one of the segments of the route, approaching an intersection. The task of the Back Seat Driver is to say whatever is necessary to get the driver to go from the current segment, across the intersection, to the next segment of the route. Most often, nothing need be said. But at other times, the Back Seat Driverwill need to give an instruction.

Instructions must use terms familiar to the driver. An example is what to say at a fork in the road. Considering only topology, there is no difference between a fork and a turn, but it would be confusing to call a fork a turn.

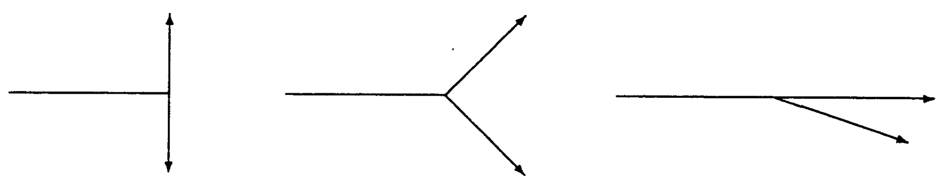


Figure 4-1: T, fork, and exit all have same topology

The difference is geometry - the angle between the two segments. Likewise the difference between a fork and a highway exit is not a matter of geometry, but requires knowing that the exit is an access ramp.

The Back Seat Driver has a taxonomy of ten intersection types. An intersection type is called an **act** because the important thing about an intersection is what action the driver takes to get across it. The Back Seat Driver is implemented with an object oriented programming methodology, so for each act there is an "expert" capable of recognizing and describing the act.

The Back Seat Driver generates speech by consulting these experts. At any moment, there will be exactly one expert in charge of telling the driver what to do. To select this expert, the Back Seat Driver asks each expert in turn to decide whether it applies to the intersection. The experts are consulted in a fixed order, the most specific ones first. The first expert to claim responsibility is selected. This expert then has the responsibility of deciding what (if anything) to say.

Each act has a **recognition** predicate. A predicate can consider topology, geometry, the types of street involved, or any other factor. The predicate also decides whether the move is *obvious*, that is, the driver can be trusted to do it without being explicitly told to do so. Actions that are obvious are not described. If the next action is obvious, the Back Seat Driver looks ahead along the route until it finds one which is *not* obvious. There will always be at least one, because stopping at the end is never obvious.

The actions are:

- CONTINUE
- FORCED-TURN
- TURN-AROUND
- ENTER
- EXIT
- ONTO-ROTARY
- EXIT-ROTARY
- FORK
- TURN
- STOP

4.2 Definitions of actions

A CONTINUE is a place where the driver stays on the "same" road. Almost always, a continue is obvious. This act is the first on the list to catch to cases where nothing should be said. The continuation of a street depends on the type of street:

- from a rotary, it is the next rotary segment.

- from an access ramp, if there is exactly one next segment, that is the continuation, otherwise there is no obvious next segment.

- otherwise, it is the one segment that requires no more than 30 degrees of angle change (if there is exactly one, and if it is not a rotary) or the one segment with the same name (if there is exactly one). The reason for comparing names is *not* because the driver is aware of the name, but because the designer who named the street was. The assumption is that if two segments have the same name, they are the same street, and that is why they have the same name. This "sameness" is presumably reflected in details not captured by the map, for example continuity of painted centerline. There are many places in the area where the obvious "straight" continuation of a segment is at an angle as great as 45 degrees, but it would not be right to call this a turn.

In almost all cases, to continue is the "obvious" thing to do. However, some drivers asked for explicit instructions before continuing across an intersection with another, larger street. This makes sense, since a common pattern for routes is to start on local roads, move onto larger roads (collectors and arterials), and then back again to local roads near the end. Drivers who ask for confirmation at major intersections are noticing that they have come to a major choice point.

In general, it would be wrong to expect to find a single universal definition for obviousness. The Back Seat Driver has a limited ability to be adjusted to suit the idiosyncrasies of the driver. This ability is in the form of a simple **user model**, which is a set of parameters about a particular user. For instance, one parameter specifies whether the user needs confirmation when crossing a major intersection. The user model can only be adjusted by a programmer, not by the user, and not by the system itself, either. A major topic for further work is to learn how a system like the Back Seat Driver can learn for itself the user's preferences and desires. Despite its simplicity, the user model makes the Back Seat Driver more comfortable for its drivers.

A **FORCED TURN** is an intersection where there is only one next street segment where the road bends more than 10 degrees. Even though there is no decision to make at a forced turn, it is useful to mention because it strengthens the driver's sense that the Back Seat Driver really knows about the road conditions. (It may be helpful to think of "obvious" as meaning "worth mentioning". It is worth mentioning a bend in the road, even though it is also obvious that one stays on the road.) The natural directions collected at the start of this study also included examples of "forced turns". A forced turn is not worth mentioning if both segments are part of a bridge, a tunnel, or an access ramp, or if the angle is less than 20 degrees. The intent here is to estimate whether the driver can see the continuation.

The **TURN AROUND** action is recognized when the heading of the car is the opposite of what it should be. Recall that a route is a sequence of segments and endpoints. At all times the car will be on one of the segments in the sequence. If the car's orientation is not the same as the endpoint in the path, then the driver must turn around. Note that the route finder only calls for a U Turn if there is no other way (e.g. when facing into a dead-end street).

The next four actions depend heavily on the street type and street quality in order to be recognized correctly. This must be emphasized because these features are not always present in digital street maps, yet without them, these acts can not be identified. (One could imagine attempting to infer the presence of a rotary from the geometry of streets in a map. It seems very difficult.)

To **ENTER** is to move onto a super street (or an access ramp that leads eventually to a super street) from an ordinary street, but not from a super street or an earlier access ramp. Similarly, to **EXIT** is to move from a "super" quality street onto a street with lesser quality that is either an access ramp or has a different name. The extra condition is needed because some "super" roads are not uniformly "super": for instance, the McGrath Highway in Somerville is a limited access road in places, but has stoplights at other places. It would not be right to call the change in quality an "exit". The name criterion does not apply to access ramps because their names have no significance.

To go ONTO a ROTARY is just to move from a segment that is not a rotary onto a segment that is, and to EXIT a ROTARY is to do the reverse. Again, this is an act which can be correctly described only if the street map database includes an explicit marking of streets as rotaries.

At a FORK, there must be at least two alternatives, all within a narrow angle, and none of the branches must be the obvious next segment – that is, the branches must all be more or less equal. Either all the alternatives must be access ramps, or none of them must be. One branch is "obvious" if it is the only branch with the same level of quality, or if it is markedly straighter than the others, or if it is the only one with the same number of lanes, provided that all of these clues agree. If one branch is stronger than the others, the intersection is not a FORK. It is either a CONTINUE or a TURN.

The STOP action is recognized when the car is on the destination segment.

Finally, a TURN is anything not handled by one of the above cases.

The greatest weakness of this approach is that the classification predicates are sensitive to small changes in the angles between segments. It is not likely that people use absolute numbers (e.g. 10 degrees) as cut-off values for their determination of how to describe an intersection. More likely, different classifications compete. Still more important, people making classifications will use visual cues, not just facts from the map.

By classifying an act, the Back Seat Driver selects an "expert" to describe it. The driver needs to know what to do, and when to do it. The next section tells how the experts decide what to say to convey this information. A general concern, not limited to any particular area, is to give the driver the impression that the program is actually present in the car, and sees the road in the same way the driver does. As will be seen, this concern has consequences in several different places.

4.3 What to do

Each action has a description function to generate a description of the action. The description function takes inputs specifying the size of the description (brief or long), the tense (past, present, or future), and the reference position. A short description is the minimum necessary for the act. It is typically an imperative (e.g. "Bear left."). A long description includes other facts about the action, an expression indicating the distance or time until the act is to be performed, and possibly information about the next act, if it is close. The reference position is a position (along the route) from which the action is to be described.

To motivate this discussion, here's a sample of the description of the left turn from Fulkerson Street to Main Street in Kendall Square, Cambridge (as seen in figure 4-2):

> Get in the left lane because you're going to take a left at the next set of lights. It's a complicated intersection because there are two streets on the left. You want the sharper of the two. It's also the better of them. After the turn, get into the right lane.

This instruction begins with a piece of lane advice, an action to be taken immediately, then describes an action in the near future. The action is a TURN, though that word is not used explicitly. It tells the direction of the turn (left) and specifies a landmark (the lights) that says where the turn is. In many cases, this would be enough, but here there are two streets on the left, so the instruction goes on to specify the desired road in two ways (by comparative position and relative quality). Finally, it concludes with some lane advice to be executed during (or just after) the act.

This is the most complicated text that the Back Seat Driver has produced. Remember length and detail are *not* virtues in giving directions. The Back Seat

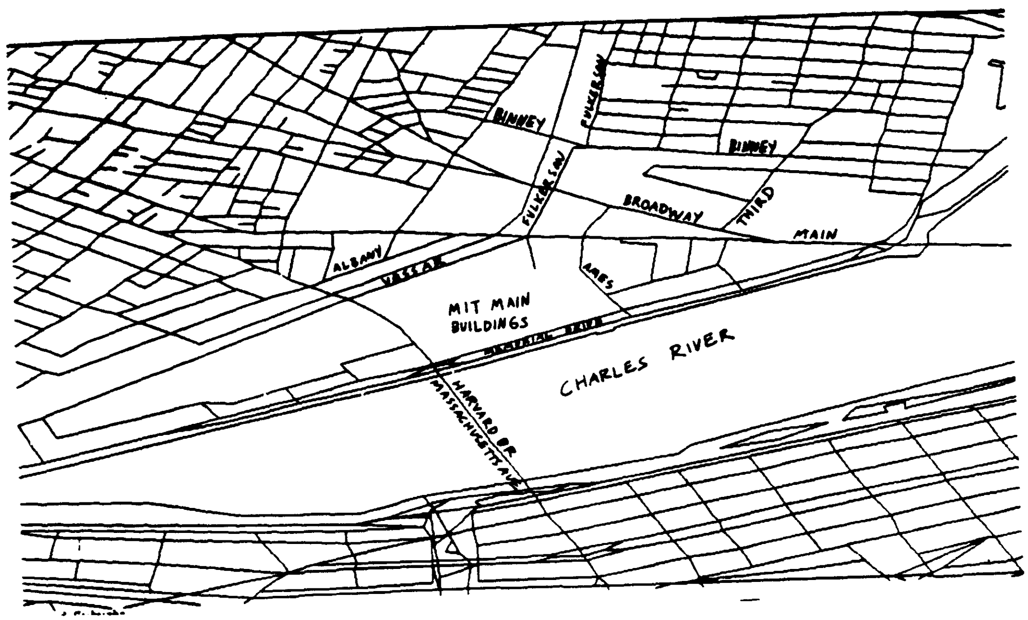


Figure 4-2: Map near MIT

Driver produces this long a text only because it has no better means to make the driver follow the route. If a shorter text would accomplish the same aim, it would be better.

A brief description consists only of a verb phrase. The verb depends on the type of act and perhaps on the specifics of the act. For instance, if the turn described above were at a less sharp angle, the verb phrase would have been "bear right" rather than "take a right". Besides the verb itself, the verb phrase must say which way to go. In most cases, the word "left" or "right" is sufficient. Where it is not, the possibilities are to use a landmark or to describe the turn. A landmark can be either in the appropriate direction ("towards the underpass") or the other direction ("away from the river"). Specifying direction with a landmark has the advantage that some drivers confuse left and right sides, or mishear the words, so it is a redundant cue. Also, it increases the driver's confidence that the system really knows what the land looks like. A description of the turn can mention either quality or the relative angle of the desired road. The angle must be described qualitatively (more or less "sharp"). It would be more precise to use the angular distance (e.g. "turn right 83 degrees"), but drivers would not understand it.

Besides telling drivers what to do, the Back Seat Driver must also tell them when to do it. The Back Seat Driver has two ways to do this. It uses *timing* ("Take a left here") when the driver has reached the place to act. When the act is more than a few seconds in the future, it uses a long description, which includes one or more **cues** which either describe the place for the act, the features of the road between the current location and the place, or the distance or time until the act. The next section describes these.

4.4 When to do it

Good directions tell the driver where or when the next action will be. One way to do this is by speaking at the moment to act, and this is the subject I discuss first. But it is also useful to give instructions before the act, if time permits. This allows time for preparation, if required, permits the driver the hear the instruction twice, and also spares the driver the need to be constantly alert for a a command which must be obeyed at once.

4.4.1 Timing

The system gives instructions at the time the act is to be done. This is the most important time to give instructions. The Back Seat Driver does not assume that the driver will recognize the place to act (e.g. by seeing a street sign) so the driver must be told when (or where) to act. The system calculates the place to begin speaking by finding a distance back from the intersection which is $v * (t_{speak} + t_{reaction})$, where t_{speak} is the time to speak the utterance and $t_{reaction}$ is the driver's reaction time. The time to speak depends on the number of words in the utterance. (The synthesizer speaks 180 words per minute.) Reaction time is taken to be two seconds.

The system speaks autonomously, but can also speak on demand. At any time the driver can push one of two buttons ("what next?" and "what now?") to ask for instructions immediately. The first button I added was "what next?" which gives the next instruction immediately. In testing the Back Seat Driver, I found that drivers hit this button when they were unsure of what to do at the time they hit it, either because they had come a place where they did not know what to do, even though the system thought it was obvious, or they were unsure whether they had come to the place where the system expected them to act. What they really needed was a "what now?" button. The "what now?" button gives the description for the very next segment transition, regardless of whether it is obvious. This function is placed on the "1" button on the cellular phone keypad, so it can be found by touch while driving.

4.4.2 Cues

A cue gives either a description of the place for the next action, or the distance or time to it. This description should be so clear that the driver cannot only recognize the place when it comes, but can also be *confident* in advance that she will be able to recognize the place. They must not only *be* clear, they must also *seem* clear[68]. Appelt points out that a description of the place to act must uniquely specify the place, but it need not be the case that the one who hears the description know what the place is at the time the instruction is heard. It is sufficient if the hearer can be sure that there is a plan that will identify the referent when it becomes visible[5]. Appelt explains the joke where one passenger riding a bus asks another at which stop she should get off, and gets the reply "one stop before I do" as a case of description which does uniquely specify a location, but for which there is no effective plan to locate.

The Back Seat Driver uses a landmark when it can, and otherwise gives a distance. Landmarks are described in the next section.

A numeric distance is the cue of last resort. Very few natural direction givers used distances, and almost all of my test drivers complained about difficulty understanding distance. The voice directions tested by Streeter[82] included distance, even though they are not much use, because it could be easily and accurately calculated, and did not hurt. I found that people differed in whether they wanted to hear distances. Some never wanted to hear them, others wanted them all the time[1] and others wanted them only when the distance exceeded a certain threshold. To allow for these differences, there is a parameter in the user model for the minimum distance expressed as a number. If the distance is below this, a qualitative phrase is used, if above, a number is used. The cutoff can be zero, in which case numbers are always used, or set to an infinite value, in which case they never are.

A cue is expressed either as a full sentence ("Drive to the end of the street, then ...") or a preposed preposition phrase ("At the next set of lights, ..."). Experience has shown that a cue should not be expressed by a preposition after the verb ("Take a left at the lights.") because some drivers start to take the left as soon as they hear the word "left". This may be because synthetic speech does not provide enough intonational cues for the driver to reliably predict the length of the sentence, leading the driver to act on syntactic information alone, and thus taking the sentence to be complete as soon as the word "left" is heard.

The description of a road feature depends upon whether or not it is visible. If it is, it can be referred to with a definite article ("*the* rotary", "*the* overpass"). If not, an indefinite article is used. The program cannot tell whether an entity is actually visible, so it uses distance as an approximation. If the feature is closer than one tenth of a mile, it is considered to be visible.

[1]A subject who played a lot of football had a good sense for distance in yards.

A special case of cues is when the driver is *at* the place to act. When stopped a few meters from the intersection, it is wrong to say "Turn at the next lights" even if it is literally true. In some cases, drivers thought that this meant not the current set of lights, but some set further on. The Back Seat Driver thinks of itself as being *at* at intersection if it is less than thirty yards away, except that if there is a stop light at the intersection and the car is not moving, then the intersection distance is fifty yards, since cars might be backed up at such an intersection. When at an intersection, the Back Seat Driver says "Take a left here". In an earlier version, it said "now" instead of "here", but this evoked violent protest from drivers waiting for a traffic light[2]. People rightly resent being told to do something they have good reason not to do.

4.4.3 Landmarks

Traffic lights are very good landmarks because they are designed to be easily seen and drivers have an independent reason to watch for them, namely a desire to avoid accidents. In an earlier version of the Back Seat Driver I used "major intersection" as a landmark. A "major intersection" is an intersection with a street whose quality is "good" where there is also a traffic light. I found that drivers preferred directions that spoke only of the lights, since that is a more precise term. When referring to a traffic light, if the car is at the intersection for the lights, the Back Seat Driver uses a proximal deictic ("this" or "these", as opposed the the distal "that" or "those") to show it means the lights that are here.

Buildings can also be landmarks. The Back Seat Driver has a very small database of buildings as part of its directory of services. If it finds a building on

[2]Given that the system knows where it is relative to the intersection, and can make good guesses about where the light is, perhaps it is possible to build a computer vision system to look for the traffic light.

the corner, it includes it as a potential landmark; e.g. 'Look for Merit Gas on the left side."

Other landmarks are features of the road, such as underpasses, bridges, tunnels, bends in the road, and railroad crossings. It is very hard to miss an underpass, since the overhead road fills the whole visual field for a time. Large bridges are likewise unmistakable, but small bridges might not be noticed. A weakness in the current map is that there is no distinction between the size of bridges. It is not at all clear that a short bridge over a railroad should be called a bridge, as opposed to an overpass. Perhaps a new category is needed.

Still another potential landmark is the road coming to an end. This is a landmark that is impossible to miss. Some people call such endings "T" turns. The Back Seat Driver says "Drive all the way to the end, then ...". A problem with this landmark is that people can recognize it too soon. Apparently some drivers take "the end" to mean not "the farthest you can go along this road" but just "the next intersection".

4.4.4 Street names as landmarks

A street name can be a landmark, but not a good one, unless the driver already knows the street. (Street names are appropriate for a driver who knows the street network, but not the route. In such a case, street names are the best kind of landmark, because they are meaningful miles in advance.) There are several reasons why street names should not be used. First, the driver may not hear the name correctly. Second, the driver may hear the name, but not know how to spell the name after hearing it, so she may not recognize the name in its printed form. This is especially a problem when the driver is from out of town. If you were told to turn on "Lemon-stir" street, would you recognize it spelled as "Leominster"?

Finally, even if the driver knows the spelling, street signs are often missing, turned around, or invisible due to weather or darkness.

The problem of hearing bears further discussion. Synthetic speech is less intelligible than human speech. This is a problem for everything the Back Seat Driver says, but the problem is worst in name recognition, because with other parts of the utterance the driver can use knowledge about the English language to help in interpreting the utterance. A driver who hears "bear left" as "bir reft" can probably reconstruct the intended meaning. A similarly distorted name often can not be repaired. Worse, drivers may not only fail to understand, they may misunderstand. One driver misheard the names of several streets, yet was so confident of the correct hearing that he drove past the turn, despite being told to "take a right".

Despite all the problems that come with using street name, many drivers ask for them. To accommodate them, a parameter in the user model controls the use of names. If set, names are supplied as part of the instruction. When names are included, they are attached at the end of the instruction ("Take the second left. It's Elm Street.") rather than directly ("Take the second left onto Elm Street."), which hopefully weakens their salience some what, and makes them more of a confirmatory cue than an essential one.

It can sometimes be difficult to form the proper name for a segment. The naming of streets is a difficult matter if the street is not an ordinary street. A bridge, tunnel, or highway should be named as a singular. We say "*the* Harvard Bridge" and "*the* Sumner Tunnel". Access ramps are harder to name, because they must be described, not named. If the ramp leads to a super highway, it can be called by the name of the highway. "Drive onto the Massachusetts Turnpike" is just as good as "Drive onto the ramp for the Massachusetts Turnpike", even if it is not literally true. If the ramp leads towards an obvious feature (an underpass, a bridge) it can be elided, and we can just say "drive towards the bridge". Otherwise, the only possible name is "an access ramp", which is probably not much help, but at least says that the road has no name.

4.5 Advice

The Back Seat Driver gives advice about how to prepare for actions. There are two forms of advice, lane advice and speed advice.

Lane advice tells the driver which lane to get into (or stay out of) when applicable. The system gives lane advice as part of the instruction when approaching an intersection where it matters. The instruction may also include advice about what lane to be in *after* the intersection, in preparation for the next act.

Speed advice warns drivers to slow down if they are traveling too fast to safely negotiate a turn. The limiting factor for angular acceleration is the driver, not the cornering ability of the car. The average driver will accept no more than .1 G radial acceleration[57]. Radial acceleration is v^2/r where r is the turning radius of the turn. The Back Seat Driver knows the geometry of the road, so it can predict the maximum tolerable velocity for the turn. It does not tell the driver about this speed (it assumes the driver will choose a comfortable speed without being told), but it does estimate the distance required to decelerate, and it tells the driver to begin slowing down early enough to do this gently.

4.6 Discourse

In order to generate more fluent text, the Back Seat Driver keeps track of what has been mentioned. Some instructions are obvious after having been given. If the system tells the driver to go straight through a set of lights, there is no reason to repeat the instruction when actually at the lights. This is in contrast with a turn, where the driver hears advance instructions to know what to do, and immediate instructions to know when to do it. This can be important, for if the driver hears "go straight through the lights" twice, she may try to go through *two* sets of lights. Indeed, not only is it redundant to say it twice, it is misleading, because some drivers take this to mean that there is a *second* set of lights to be driven past.

To implement this, each instruction is able to determine whether it is obvious after having been given once. When it is time to speak the instruction, if the instruction has already been given, and it is obvious once spoken, then it is not spoken again.

The Back Seat Driver also retains a history of the route. This allows it to generate cue phrases for the instructions. If the route calls for doing "the same thing" twice in a row (e.g. two successive rights), the system uses the word "another" to indicate this doubling. This is important for polite behavior. If I (as a passenger) give you (as a driver) instructions by just saying, e.g. "Take a right. Take a right. Take a left. Take a right.", pronouncing each the same, you will judge me to be rude. My speech is not acknowledging your actions or your history[3]. There are two ways I could acknowledge your work: I can use cue words, and say instead: "Take a right. Take another right. Now take a left.", or I can use intonation to indicate that my instructions are all part of a series. In this case, I would use the so-called "list" contour[49]. The Dectalk speech synthesizer does not support flexible control of intonation, so cue words are the only possibility.

4.7 Mistakes

If the driver leaves the route the system immediately informs the driver and

[3] I thank Chris Schmandt for this explanation.

begins to plan a new route. Route planning after a mistake is no different from any other time, except that the vehicle is more likely to be moving. Telling the driver what she did wrong prepares her for hearing new instructions, and perhaps helps her learn to better interpret the style of language that the Back Seat Driver uses.

To describe an error, the system looks forward from the last saved position for the first action, even if obvious. (While the driver is on a route, the system records her position, so that if a mistake occurs the system can find the place where the mistake occurred.) This is action that the driver failed to perform. It utters a description of this action, saying e.g. "Oops, I meant for you to take a right." This error message is a change from the original message, which was of the form "You made a mistake. You should have taken a right." The new message is neutral, because test drivers complained about being blamed. A driver might leave the route deliberately, or the error could be system's, not the drivers.

An unsolved problem is how to reliably detect when the driver has left the route. At present, if the driver turns into, say, a gas station, the system will believe, falsely, that the driver has turned onto some street, because the street map includes only streets, and not other paved areas such as parking lots and filling stations. From this false belief, the system will conclude that the driver has made a mistake. If the street map included more information, the system could avoid this error.

Sometimes the driver will choose to not follow for good reasons that the Back Seat Driveris unaware of, perhaps because the road is blocked or because of a traffic jam[4]. In the first case, the driver can push a button - the "I Can't Do It" button - which informs the system that the road is (temporarily) blocked. The system automatically finds a new route. In the second case, the driver's only

[4]In the future, we can expect that the Back Seat Driver will be monitor machine-readable broadcasts to obtain such information, and plan a new route automatically

recourse is to cancel the current trip (by pushing another button), then drive in the desired direction, and then request a route to the original destination (which she may easily do by touching a single button). It is essential, though, that the driver either notify the Back Seat Driver of the impossibility of the requested action or cancel the trip, because otherwise the system will treat the deviation from the route as a mistake, and continue to attempt to find a new route, which may very well lead back through the street the driver is trying to avoid.

4.8 Reassuring

While the driver is following a route, the system adopts a persistent goal of keeping the user reassured about her progress and the system's reliability. If the Back Seat Driver were a human, this might be unnecessary, since the driver could see for herself whether the navigator was awake and attending to the road and driver. But the driver can not see the Back Seat Driver and so needs some periodic evidence that the system is still there.

One piece of evidence is the safety warnings the system gives (page 90). But if all is going well, there will not be any. The system gives two other kinds of evidence that things are going well. First, when the user completes an action, the system acknowledges the driver's correct action, saying something like "nice work" or "good". This feature is very popular with most test drivers. When I turn it off, most people complain. A few find the choice of words patronizing, but still want the function. Those who want no confirmation messages can turn them off by changing the user model.

The second form of evidence is to make insignificant remarks about the roads nearby, the weather, and so on. If the driver assumes that the navigator is being cooperative, as set out in Grice's maxims of cooperative conversation [28], then the driver can infer that everything is going well, for otherwise the navigator would not speak of trivial matters.

Chapter 5

User and System Goals

The Back Seat Driver is a truly *interactive* system, by which I mean that both parties are active all the time. Most "interactive" computer systems are better called *reactive* - the machine and the human take turns. A typical reactive system is the command interpreter for a time shared operating system. While the human is typing, the computer does nothing except echo typing. It begins to work only when the human hits the `Return` key. Then it's the human's turn to wait. While the machine is working, there there is little the human can do, except to interrupt a calculation gone awry. Giving driving instructions is an inherently interactive task because the concept of "turn taking" does not apply. The driver is always in some kind of action. The driver may be waiting (e.g. for a traffic light to change) but is never (or should never be) waiting for the system to say what to do. The task of driving does not divide into discrete steps, with an arbitrary pause between each step. The driver may choose to *stop* driving for a time, but if the driver is driving at all, she's driving all the time. So the machine must be ready to speak when needed.

5.1 Resource allocation

The Back Seat Driver pursues many goals at once. Its main goal is to get the driver to some location. This goal requires many actions over an extended time. Other goals include delivering messages, weather reports, and notifications[1]. These goals can not all be met at once, so it must choose which to pursue at any given moment. To make this choice properly requires a model of action which accounts for the time required to perform an action. The only kind of action the Back Seat Driver takes is talking. Talking takes time, and only has value if completely performed. Talking is different from, say, drinking a cup of coffee. A partially drunk cup still has some effect. An incomplete sentence has an effect, too, but not a useful one. The program should only take an action when it can expect to complete it.

Since the Back Seat Driver can only say one word at a time, it must choose its words carefully. At any moment when the Back Seat Driver is not already speaking, it can either begin saying something, or it can wait. This decision must be made again and again, at each passing moment. A wrong decision can not be undone. Words once spoken can not be made unsaid, though the program can stop talking at any time; more seriously, there is nothing to be done if the program decides at some moment that it should have begun speaking two seconds previously[2]. Since the driver's actions are unpredictable, the program must improvise. Plans can not be made in advance.

The problem, then, is to allocate resources to goals so that the most important goals get all the resources they need, while lesser goals get any useful amount left over. Although the only resource used is speaking time, the method used is designed to be applicable to goals that use many different kinds of resources.

[1]and someday, reading mail, advertisements, tourist information, and traffic advisories
[2]Save perhaps to speak faster, a possibility I do not consider further.

Each goal provides three pieces of information about itself:

- Is it ready to run – (e.g. is there something that could be said right now which would help achieve the goal?)
- If so, the resources it will require, and the maximum time it will require each resource.
- If not, the minimum time until the goal will be ready to run.

Each goal decides for itself whether it is ready to run. The criteria depend upon the goal. For instance, the goal to deliver electronic mail will only be able to run if there is some mail to speak. A goal can be certain of whether it is ready to run, but the latter two measures may be estimates if they depend on the driver's future actions. For instance, while following a route, the time when the next instruction will be given depends on the speed of the car. If the driver speeds up, the time will come sooner, and if the driver makes a mistake, corrective action is required at once. Given these estimates, and a preassigned priority for each goal, the program will at each moment authorize at most one goal to run. It does this by examining each goal in order of decreasing priority. The goal of highest priority that is ready to run will be allowed to do so, unless there is some other goal of greater importance which uses a resource used by the lesser goal and, though not currently ready, expects to begin in less time than the the lesser goal will use in speaking. (See figure 5-1.) Thus the Back Seat Driver does not start a narrative it can't finish. Nevertheless, interruptions do sometimes happen when driver does something unexpected or adds a new goal.

5.2 Architecture

This section discusses the architecture of the goal mechanism.

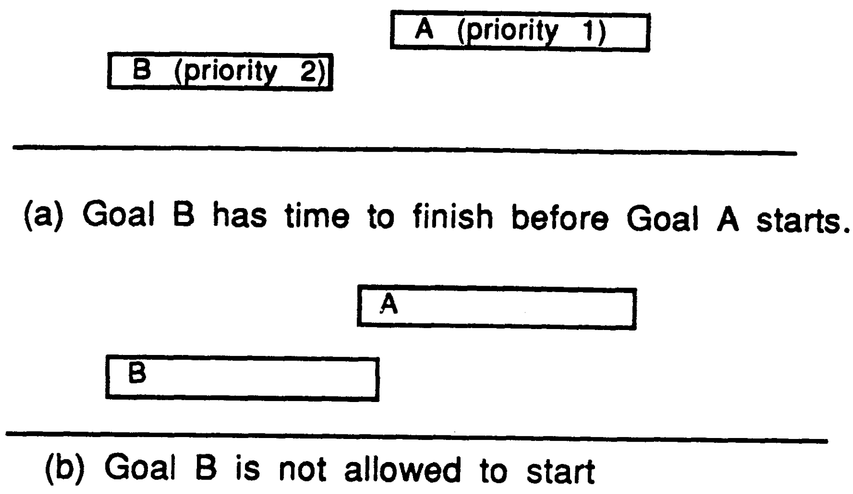


Figure 5-1: Resource allocation

The unit of planning in Back Seat Driver is the **goal**. Every goal has a **name** and an **importance**, which is a number from 1 to 9. Goals may also have any number of **slots**, each of which has a name and a value. Slots hold information about the goal. The unit of execution is the **goal function**. A goal function is a module capable of taking action on behalf of a goal. The definition of a goal function includes the name of the goal it can work for. In theory, there could be several goal functions relevant to a goal, but at present there is always exactly one. A goal is mostly a holder for slot information. The actual work is done by the goal function. In the discussion that follows, the term "goal" is used for both goals themselves, and goal functions, except where the distinction is important. The unit of resource allocation is the **resource**. There is only one resource, the speech resource.

The system maintains a list of goals. When a goal is added to the list, the system finds the appropriate goal function and attaches it to the goal. The goal function can then access the slots of the goal to determine what it needs to do.

The system communicates with goals by message passing. The goal function protocol includes the resource allocation messages described on page 84 and a message which allows a goal to run. When a goal runs, it is expected to take one action, and then return. The system has no way of interrupting a goal function while it is running, so actions should be quick.

Most goals achieve their aims by speaking. A goal speaks by requesting the speech resource to send characters to the speech synthesizer. This can be done fairly quickly, but the synthesizer requires substantial time to finish speaking. This is why I say that actions take time. During this time, the goal can make no further progress. The goal must be able to tell when speech is complete, for only then can it claim to have achieved the desired effect. (I assume that if the speech was spoken, it was also heard and understood by the driver. If not, the driver will take some action such as requesting a repeat.) The speech resource retains the identity of the goal which last spoke. Periodically it polls the synthesizer to determine whether the speech is complete. When the speech has finished the speech resource becomes available again. Usually the goal that produced that speech will run a second time, notice that the speech is complete, and finish its operation.

Goals may interrupt lower priority goals by requesting the speech resource to interrupt the lower priority goal. Interruption stops the speech synthesizer immediately. The interrupted goal is informed of the interruption, and can react as it chooses. There is no way for the goal to know whether any of its words were actually spoken, so it has to start all over. Most goals attempt to run again as soon as possible. The "help" goal simply quits when interrupted, on the assumption that the user has learned whatever she wanted to know, so no further help is required. The assumption is that the interruption occurred because the user started some higher priority goal after learning how to do so through the help command.

Not all actions of a goal are of equal importance. A goal can lower its own priority (but not raise it) if it wishes to say something of lower importance.

Goals may have subgoals. Subgoals can be executed for side-effect or to return a value. (The route finding goal returns the route found as a value.) A goal with a subgoal is not eligible to run until the subgoal completes. A goal that is a subgoal has a slot pointing to its parent goal. When a goal finishes, if it is successful, it adds a slot to itself to hold its returned value. This slot is empty if no value is returned, or if the goal failed. It then informs the parent goal of the outcome (success or failure). If a subgoal succeeds, the parent goal is again eligible to run. It can get the value returned by the subgoal by looking at that goal's result slot. If the subgoal fails, the parent goal may try again or may give up.

5.3 Kinds of Goals

Goals can be temporary or persistent. Temporary goals can be satisfied, but persistent goals never can be. Temporary goals are about attaining a condition which does not currently hold, and persistent goals are about maintaining a condition which is true now, should stay true, but might become false without at least occasional action by the system.

The user creates a goal by striking a button on the cellular phone keypad. (In a more advanced system this might be done through speech recognition.) Examples of user initiated goals are:

- to get to some location
- to find the closest provider of a service
- to hear a weather report
- to hear the last statement repeated

In order to get to a location, the system must first know what the location is and how to get there. For each of these it creates a subgoal. The subgoal to get the address of the desired location asks the user to enter it with the cellular phone keypad, in the same manner used in Direction Assistance (see page 102). The subgoal for finding a route invokes the route finder, as discussed in Chapters B and 6. Once the system has a route, it creates two subgoals, one to have the user follow the route, and one to keep the user confident of her progress along the route. These goals are described in chapter 4. There is just one goal for following the route, rather than a separate subgoal for each intersection along the route. There is no large theoretical reason for this, but it does make it easier to give information about the route as a whole (e.g. "How much longer is it?")

5.3.1 Finding Services

Drivers sometimes know what they want, but not where to go to get it. What they need is a geographically coded directory of services, like the Yellow Pages. Drivers may want to browse the directory and select a service themselves. If they are interested in a commodity where the vendor is unimportant, it may be more convenient to let the system select the destination. When running low on gas late at night in an unfamiliar area of town, you want to find the closest gas station that is open. You do not care which station you go to, you just want the closest. The Back Seat Driver can find the closest provider of a service. The directory of services, an electronic "Yellow Pages" has entries for services indexed under topics. The directory holds only three kinds of services: gas stations, automated teller machines, and ice cream stores. The entry for each service includes the name, the address, the hours of business, and a phone number. Upon the request[3], the

[3]The present interface will not extend gracefully to a much larger number of services, because it asks the user to select the kind of service by choosing from a list, in the same way that that the user might select a city when choosing among alternative destinations. The Boston Yellow Pages has more than 3600 categories, so choosing in this way would take about an hour on the average. On the other hand, it is unlikely that a "spelling" interface would be successful, since there are many alternative wordings for each category. Selecting from a list this large is better done with a textual interface which allows rapid scanning.

system finds the geographically closest[4] provider known to be open, and offers to find a route to it.

5.3.2 Other user goals

The system obtains weather information from an online weather station. The station returns a structured list of the current weather conditions, and the Back Seat Driver assembles a set of sentences to speak the information.

The system treats "repeat the last statement" as a goal, rather than as a special purpose function, except that the importance of this goal is set to the value of the the last goal to speak (the goal whose utterance is being repeated). This guarantees that if some more important goal desires to speak, it will be able to. A repetition of an utterance is no more important than it was originally.

Repetitions should not necessarily be literal. If the situation has changed, a repetition of what was once true may now be a lie. For example, the car may have moved, and the turn that was once the "second right" might now be the next right. The goal function for repetition determines the last goal to speak by asking the speech resource. This goal is then asked to generate an utterance equivalent to its previous utterance, providing for the opportunity for the goal to choose different words (or a wholly different message) if need be.

5.3.3 System goals

All system initiated goals are persistent. The system goals are:

- to inform the user of new electronic mail

[4]It would be better to find the one that is most easily reached, but it is too expensive at present to compute.

- to deliver messages from the base station
- to warn the driver of dangers ahead.

These goals can never be satisfied: mail or messages can arrive at any time, the user's safety should always be preserved.

The first two goals are easily implemented. The Back Seat Driver can check for new mail by consulting the computer that stores electronic mail. This computer is accessible via a local network. At present, the Back Seat Driver can only say that new mail exists. It would be possible, and desirable, to read the mail aloud, and to integrate voice mail and text mail, as in the Phone Slave[75, 77]. The Back Seat Driver can also deliver messages entered at the base station. Communication from the laboratory to the car has mostly been useful for debugging (and the occasional joke).

The Back Seat Driver can warn the driver about those dangers which can be inferred from knowledge of the road network. These dangers include:

- driving above the speed limit
- driving the wrong way on a one-way street
- driving too fast for an upcoming curve
- driving on a one-way street that becomes two-way ahead
- merging traffic
- "blind" driveways ahead
- speed traps
- poorly repaired roads
- dangerous intersections

These dangers are not of equal severity. Some are potentially fatal, others mere annoyances. The first five have been implemented. The others can be added if the required features were added to the database. A difference between the Back Seat Driver and other systems which have provided safety warnings is the the Back Seat Driver attempts to determine hazards by reasoning about road conditions rather than requiring them to be built in. Warnings are remembered, and not repeated.

Speed limits and one-way streets are included in the map data base. If the driver is driving more than five miles per hour above the speed limit, the system issues a warning. This feature was actually requested by some users. Warnings about driving too fast were not nearly so popular. The maximum safe speed is computed by finding the safe turning speed (see page 77) for each possible segment at the next intersection. To avoid false alarms, the maximum value is taken. This speed is too great if the driver intends to turn, but it is better to give false negatives (i.e. be silent) than false positives.

The Back Seat Driver warns drivers about potential dangers in traffic flow. There are intersections in the Boston area where traffic flows from a one-way street straight onto a two-way street. The danger here is that a driver could take the oncoming left lane to be a second lane for forward travel. A second "flow" problem is where the number of lanes is reduced ahead. The driver may need to change lanes, or watch for other cars merging in at the last minute. Both these dangers are marked with warning signs on the street, so it appears reasonable for the Back Seat Driver to also give warning.

Giving warnings is itself something of a danger. One person who tested Back Seat Driver remarked that the system did such a good job of telling her what to do that she assumed it would watch over *all* details of the driving, and she did not feel as responsible herself. It is not at all clear what to do about this possibility.

Chapter 6

Comparing Routes

Drivers want the Back Seat Driver to find the best route to their goal, or at least a good one. This means that the Back Seat Driver needs a way of comparing two routes, and selecting the better of them. This chapter describes how the Back Seat Driver makes such comparisons. In addition, since different drivers have a different sense of what makes a route "good", the Back Seat Driver has three different comparison functions. The driver can define a good route as one that is either short, or fast, or simple. (Planned, but not implemented, are the abilities to find "scenic" routes, novel routes, and routes that pass near providers of services, for example gas stations.) Readers who are not already familiar with the A* search algorithm should consult chapter B before reading this one.

6.1 Comparing routes requires a metric

To find the best route requires a function which can compare two routes and select the better of them, according to some fixed criteria provided by the user. For technical reasons described in chapter B, the comparison function must not only select the better of two routes, it must also provide a numeric rating of "goodness" for each route. Such a function is called a **metric**.

The Back Seat Driver has three different metrics. The **distance** metric finds the shortest route, the **speed** finds the fastest route, and the **ease** finds the easiest route. Figure 6-1 shows the route each metric produces for a trip from 12 Albany Street to 4 Glenville Terrace. Compare these routes with the route provided by a human in figure 2-2.

Figure 6-1: Four routes from 12 Albany Street to 4 Glenville Terrace

The shortest path in figure 6-1 is 2.73 miles long. Neither the "fastest" nor "easiest" paths found are optimal by their respective metrics, because the A* bias parameter (see page 147) is greater than 1. This makes the route finder less willing to go out of the way in searching for a route. The actual "fastest" path is shown as a thick line. This is also the "easiest" path. When finding the shortest path, the A* bias parameter (D, see page 147) is 1. This makes search slower, but if the driver is really interested in the shortest path, the extra time to search is justified.

The distance metric is just the sum of the lengths of the component segments. The other two metrics are more complicated than the distance metric, because they must consider intersections as well as segments. In general there is a cost to travel along a segment and a cost to get from one segment to another. All costs are expressed as an "equivalent distance" which is the extra distance one would travel to avoid the cost.

The metric for speed estimates the cost for traveling along a segment by multiplying its length by a constant which depends upon the quality of the street. In principle, one could calculate expected time by dividing length by the average speed on the segment were this quantity available in the data base. The constants used are:

Quality	factor
super	1
good	1.2
average	1.5
bad	2.0

All multiplicative constants must be greater than or equal to one, to ensure that the cost of a route is never less than the straight line distance between two points. This condition is essential to the correct operation of the A* search algorithm, since the estimation function ($g*$) must always return an under-estimate.

The time to cross an intersection is modeled by a mileage penalty which depends upon the nature of the intersection.

Factor	cost	reason
turn	1/8 mile	Must slow down to turn
left turn	1/8 mile	May have to wait for turn across traffic flow
traffic light	1/8 mile	Might be red

If the segment is one-way, the penalties are cut in half, since there will be no opposing traffic flow. The turning penalties are computed based only on the angle between two segments, not on segment type or quality. The angle used is the smallest possible angle for the two segments, given the uncertainty of positions inherent in the DIME file. This nullifies the DIME "wobble" described on page 46.

The metric for simplicity seeks to minimize the driver's effort in following the route. Again, driver's effort is the sum of the effort to travel along a segment and the effort to get from one segment to another. Travel along a segment depends upon its quality. A limited access highway is the easiest to travel along, since all traffic is going the same way and there are no intersections. Low quality segments are the worst to travel along – the driver must be wary of potholes, and their narrowness requires careful attention. Turns of every sort are penalized equally, since they all require decisions. The intention of this metric is to find routes which require the least amount of speaking by the Back Seat Driver, leaving the driver free to concentrate on other matters. A more advanced Back Seat Driver should perhaps select this mode automatically if the driver is listening to music or having a conversation.

6.2 Estimating the time required to find a route

The Back Seat Driver must be able to find routes while the vehicle is moving, yet the route finding algorithm requires a fixed origin. When the car is moving, the Back Seat Driver first estimates the distance the car will travel during the route finding process by multiplying the current velocity by the estimated time to find the route. Then it finds the position the driver will reach after traveling this distance, assuming that the driver will not make any turns without being told to do so. It then finds a route from this extrapolated position to the goal. Finally, it finds a route from the cars actual position to the estimated starting position. This second route is so short that the car is unlikely to move far during the time it is computed.

The route finder estimates the time to find the route between two points by multiplying the distance between them by a constant. This constant was initially determined by running the route finder for 20 randomly selected pairs of origins and destinations. As the Back Seat Driver runs, it accumulates additional values for the constant.

Part III

Conclusions

Chapter 7

Related Work

This chapter first surveys related work on computer programs which provide navigation assistance to drivers, then develops a taxonomy of such systems. A related survey can be found in [56].

7.1 Early Work

Early application of computers to navigation was intended to reduce traffic congestion by providing route information to drivers. The designers of the Electronic Route Guidance System (ERGS) intended to make traffic flow more efficient by balancing load. They believed that reducing driver uncertainty at decision points would make traffic flow faster and more safely. In the ERGS design, a driver beginning a route finds the intersection closest to the destination, then enters a five letter code word for the intersection. When the vehicle passes over an induction loop sensor in the road it transmits the destination to a central computer. The computer determines the best route, and relays instructions to the car. This interchange of information occurs at every instrumented intersection. Driving directions combine simple text from a nine word vocabulary and directional arrows, and are displayed by a "heads-up" display. The ERGS system was designed but never implemented[70]. A similar system was designed and tested in Germany in the late seventies[14].

7.2 Elliot and Lesk

The pioneering work on computer navigation assistance is by Elliot and Lesk[21, 22]. They showed that general purpose graph search algorithms[1] are not suited to the problem of finding useful routes for people traveling in the real world. There are two reasons why this is so. First, general purpose graph search algorithms are complicated because they must work with any kind of abstract graph, but street maps are among the simpler forms of abstract graphs. The extra complexity in an algorithm for arbitrary general graphs makes it slower than one which is specialized for searching simpler graphs.

A second problem with general search algorithms is that the shortest route may not be the *best* route. It might be a maze of shortcuts. Elliot and Lesk say that people who saw such routes "recoiled in horror", and so they modified their algorithm to prefer a route which was slightly longer but had fewer turns. The usual algorithms for graph search consider only the cost of traveling along an arc

[1]Abstract graphs are the mathematical basis behind all computer route finding. An abstract graph is a set of **nodes** (points) and **arcs** (lines joining two nodes). To represent a street map as a graph, think of intersections as nodes, and streets as arcs. If two intersections are directly connected there will be an arc between the corresponding nodes. A route between two nodes is a chain of arcs, each leading from one node to the next, such that the origin node is the first node, and the destination is the last node. Each arc may have an associated **cost** (greater than zero) which represents the effort required to travel the associated street. Usually this would be the length of the street, but it might also represent the toll collected on a turnpike. If the cost is the distance, then the shortest route is the route the sum of whose arcs is least. Finding a route, or the shortest route, through a street map is a special case – and perhaps the best example – of the problem of searching an abstract graph.

(a street), and take the cost of going from arc to arc to be zero – that is, having arrived at a node, all arcs are equally accessible. But this is not true of street driving. It takes more time to make a left hand turn across oncoming traffic than it does to go straight, and it takes extra effort to locate the place to turn. Elliot and Lesk added a system of **weights**, which are extra costs associated with turns. To their algorithm, the cost of a route was the sum of the distance along the streets and the difficulty of making the turns. They set the weight of a left turn to be 1/4 mile and a right turn to be 1/8 mile. This caused the algorithm to prefer slightly longer routes with fewer turns to short, twisty routes. Every known route finder trades distance for simplicity.

Elliot and Lesk also were the first to implement a program to generate natural language driving instructions for the route. This is not a straightforward translation. To understand why, you must understand that the map represents each street as a set of segments, where a segment is a piece of a street chosen to be short enough that it is a straight line and no intersection with any other segment occurs except at a segment endpoint. The route as represented by the route-finding algorithm is a sequence of street segments, not of streets. A street segment does not match any common sense notion of a road. Route descriptions must be expressed in terms of motion along streets (across many segments) and turns, not as a list of segments.

In their instructions, a route consists of a beginning, a sequence of turns and crossings (of rivers or railroads), and an ending. For each of these, there is a template to generate a sentence. A template is a sequence of words and slots, representing fixed and variable components of a sentence for a given type of act. The words are copied directly to the output, and the slots are filled in according to the particulars of an act. An example template is

```
Go <distance> [<intersections>] turn <direction> on <street>.
```

Here "Go", "turn", and "on" are the fixed words, and everything else is a slot. The slot **<intersections>** is optional. This template might produce:

```
Go 0.3 miles (2 intersections) turn left on TROY HILLS RD.
```

A third contribution of Elliot and Lesk was to integrate the digital map with other location oriented databases, including a Yellow Pages and a personal address book. This allowed the program to find routes to addresses given a person's name, to find the closest store of a specified category, and to mention stores along the route as possible landmarks.

7.3 Direction Assistance

Direction Assistance[19] provides spoken directions between locations in the Boston area. It uses a Dectalk speech synthesizer. This synthesizer includes a telephone interface, so it can answer a phone call and decode touch tone button presses. To use Direction Assistance, you call it from a touch tone phone. It answers the call, and prompts you to enter your origin and destination locations as street addresses. It finds a route, then describes the route to you. Direction Assistance was directly inspired by Elliot and Lesk, and extends their work in three ways.

The most significant difference is that Direction Assistance speaks its directions, where Elliot and Lesk drew maps and provided written text. Using speech makes the program much more accessible, since only a touch tone phone is required, rather than a computer terminal. The disadvantage is that users must remember the instructions or write them down.

7.3.1 Entering addresses

The method of address entry is of some interest. You enter an address by first entering the digits, then a number sign, then spelling the street name using the letters on the telephone keypad. The letters "Q" and "Z" are on the "6" and "9" keys, respectively, and the space character is on "1", which is otherwise unused. These keys are sufficient to spell any street name in Boston. (The spelling rules would require expansion to enter street names with other characters in them, for example "4th Street".)

Spelling a street name requires only one button push for each letter, even though there are three letters on each key. This is because of the redundancy in street names, which are pronounceable words, not arbitrary strings. There are 37 pairs of street names with the same "spelling" in the reduced "alphabet". An example is "Flint" and "Eliot", both encoded as "35468". This is only one percent of the (2628) names of streets in Boston, so collisions are rare. This technique appears workable even for larger sets of names. When the entire word list of the Brown corpus are encoded, there are still only 1095 collisions in the 19837 words (5.5%).

If a name collision occurs, the interface reads the list of possibilities, and asks the driver which one was meant. This is very rare. A more common problem is that street names are duplicated. For example, there are 14 possible meanings for "10 Washington":

```
10 Washington Avenue, East Boston
10 Washington Avenue, Union Square, Somerville
10 Washington Avenue, Chelsea
10 Washington Court, Central Square, Cambridge
10 Washington Place, South Boston
10 Washington Square, Charlestown
10 Washington Street, Charlestown
10 Washington Street, Central Square
```

```
10 Washington Street, North End, Boston
10 Washington Street, Union Square, Somerville
10 Washington Street, Brighton
10 Washington Street, Brookline
10 Washington Terrace, Union Square, Somerville
10 Washington Terrace, Charlestown
```

(This is the worst case example. There are only half as many possibilities for "100 Washington".) When this happens, the system asks the user a series of questions to reduce the list to a single choice. The system tries to ask the fewest questions possible. It asks the user to choose from a list of street types, if that is sufficient to resolve the question, and otherwise from a list of the containing cities (or neighborhoods, if there are two instances within a single city). To select from a list, the system reads the contents, asking the user to push a button when the desired choice is read. The interface is described more fully in [18].

A problem not addressed by Direction Assistance is that some "addresses" do not refer to streets at all, but rather are the names of buildings or developments, e.g. "11 Cambridge Center" or "One Kendall Square". Direction Assistance can only understand addresses expressed in terms of named streets.

7.3.2 Generating text

A second significant difference between Direction Assistance and the work of Elliot and Lesk is that Direction Assistance generates better quality descriptions of the route. The improvement arises because the text generation process first analyzes the route into a sequence of "acts", and then generates descriptions from these acts, instead of working directly from the route. An act represents something that the driver does. There are eleven different acts, each representing a different way of moving. The type of act depends upon topology (how many streets are present at an intersection, and which way traffic can flow), geometry (what angles the streets make) and what kind of streets are involved. Thus we say "bear right at the fork" rather than "turn right", but we don't say that in taking an exit from a highway we are "bearing right". An act may involve more than one segment, as for instance a "U Turn" on Memorial Drive (shown in figure 7-1) takes one from Memorial Drive, to Danforth, and back onto Memorial Drive, yet should not be described as two successive turns. For each act there is a specialized text generator

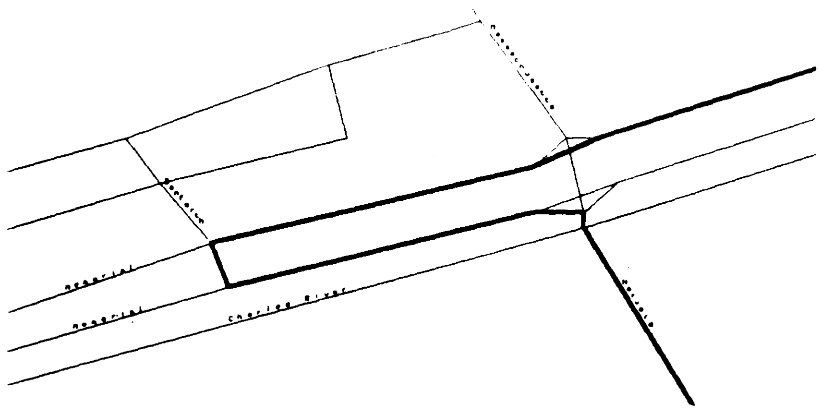


Figure 7-1: A U turn

to describe it. There is also a function to find an appropriate cue or landmark (e.g. a street crossed or an underpass) just before the location of the act.

7.3.3 Route Finding differences

The Direction Assistance route finder uses a different algorithm than Elliot and Lesk, and has a different set of weights. The algorithm is the A* search[35]. The weighting scheme ranks roads by a four-valued "goodness" feature and penalizes routes that use less good roads by multiplying the mileage by a constant factor. It also reduces or waives the penalty for turning under a set of circumstance having to do with predicted ease of following; for example, a turn onto a one way street incurs a lesser penalty, since it is unlikely that the driver would turn the wrong way. It reduces the penalty for "T" turns since the driver can not possibly miss the place to turn. In practice, these weight changes have very little effect.

7.4 Text-Based Directions

7.4.1 Counter Top Directions

The Hertz car rental company offers "Computerized Driving Directions" at some of its rental counters. The directions include approximate mileage and estimated travel time, but are highly schematic, even cryptic. An example appears in figure 7-2. Despite appearances, these instructions are not computer generated.

```
APPROXIMATELY 16.8 MILES 0 :35 TIME
 2.0 MI NORTH  TO   I-78 WEST           enter LEFT
14.0 MI WEST   TO   NEW PRO/BERKELEY EXT bear RIGHT
                    DIAMOND HILL RD continue
 0.4 MI        TO   MOUNTAIN AVE turn RIGHT
 0.4 MI        TO   AT&T/BELL LABS on your right
```

Figure 7-2: example of driving instructions provided by Hertz

The Hertz system is more akin to a database retrieval system than a route finder. A California firm, Navigation Technology, sells a product called "DriverGuide" which is reported to be able to print driving directions between any two points[71, 4] in the San Francisco area.

7.4.2 Ma

Peeder Ma describes a system which gives textual directions in [51]. His work is similar to both Elliot and Lesk's and to Direction Assistance, but was apparently created independently of both. Ma uses A* search with a penalty factor to minimize the number of turns. Unlike Elliot and Lesk, he uses the same penalty for both left and right turns. His street map representation does not include one-way streets or restrictions on turning ("no left turn") so it does not always find usable routes. His route descriptions use a taxonomy about as elaborate as that of Direction Assistance, but the text generated is more stylized.

7.5 Automotive Navigation Systems

Several groups have built position or navigation systems for use in automobiles. For the most part, these systems have not been well described in the literature, probably from a desire to preserve commercial secrecy.

The most well known in the United States is the ETAK Navigator, which displays the car's position on a map display on the dashboard[36, 89]. The map rotates as the car turns so that the forward direction is always straight up on the map. The ETAK Navigator can show the map at four different scales. At greater scales, the display shows only large streets in a straightened form. This is important to keep the map legible at large scales. The system provides a limited amount of navigation assistance. The driver may enter a destination (as a street address or intersection), and the system will display the direction and distance to the point. It remains the driver's task to select an appropriate route to the destination.

The Routerechner provided directions in and between German cities [30]. The route finder could receive real time traffic information by digital radio while on route. This system's map included only the Autobahn, and not the cities (this was before CDROMs were widely available), yet it was also provided a limited navigation service within cities. The user entered the destination as a pair of coordinates, and the system displayed the direction and distance to the destination. As with

ETAK, it was the user's responsibility to select an appropriate road. Haeussermann reports that users were always successful at finding their destinations and were pleased with the system.

The Honda Electro Gyro Cator[83] provided displayed position of the car by plotting a point on a screen. The driver could determine position by placing a transparent map over the screen. This system did not provide route directions.

The Nissan-Hitachi car navigation[38] and information system displays position on a map, finds the shortest route to a destination taking into account real time traffic information, and gives directions by arrows on the face of a display. The system also includes a "secretary mode" which displays the driver's appointments. The system uses a CDROM for map data, and combines satellite positioning with dead reckoning.

The EVA[65, 58](Automatic Navigation System) developed in Germany by Blaupunkt, the University of Karlsruhe, and the federal government, accepts destinations as street addresses, finds minimum time routes, and gives directions by a combination of simple (arrow) graphics and voice. The system can recover from a driver error in following the route and find a new route within 50 meters of travel.

The Phillips corporation, in the Netherlands, is developing a prototype car information and navigation system called CARIN[85, 8]. The driver enters a destination using either a keyboard or a touch sensitive screen. The system displays routes on a map and gives spoken driving instructions. The map is stored on board in CDROM, and a radio link provides for updates on traffic conditions. The system is potentially interesting, but very little has been published about it.

7.6 Classifying navigation systems

Navigation systems can be classified along three dimensions. There are three kinds of navigation service:

- **positional** systems tell you where you are.
- **orienting** systems show the direction of your destination.
- **instructional** systems tell you what to do.

A navigation system can provide one, two, or all of these services. Navigation systems can be further distinguished by how they provide the information:

- **verbal** systems speak.
- **text** systems provide text.
- **graphic** systems provide pictures.

Finally, systems can be classified as either **real time** or **static**. The categories of this classification are not independent. There can be no static positioning system, since one can not predict the future position of the car.

The systems of Elliot and Lesk, Ma, and Hertz provide static, text instructions. Direction Assistance gives static verbal instructions. There are several problems with "static directional" navigation systems. First, they do nothing to help the driver follow the route. The driver must determine for herself when to apply each instruction. Instructions like "drive half a mile, then turn left onto Maple Street" are no use if the driver is unable to measure mileage or can not determine the name of the street. The urban street network contains many short connecting roads (access ramps) which are nameless. Finally, even a named street might be missing its sign. In addition, the driver must keep track of which instruction is next. A second problem is that since the instructions must be specified in advance, there is little to be done if the driver does not follow the instructions, which might happen from error, or because the instructions are wrong, or simply ill-advised (as when confronting a traffic jam). The simulated navigation system constructed by Streeter included instructions of the form "If you see this you've gone too far" but none of the actually implemented system did this.

Most systems which provide positions also provide orientation. (The Honda Gyro-cater provided position only, and the Routerechner provided orientation only.) Positional or orientational systems can be useful for navigation provided one can read a map and find one's own route. It is not clear whether the CARIN system retains a map as a "vestigial" display, or because its makers do not appreciate the superiority of speech, or because they see a need for positional information other than route finding.

Chapter 8

Future Developments

The Back Seat Driver works. It works well enough that one can imagine something very much like it being sold within a few years. There remain some areas for further research. Some of these involve only the Back Seat Driver, while others have to do with how a system like the Back Seat Driver would fit into the larger context of urban planning and public policy. The Back Seat Driver of the future will not simply be something installed in a car, rather it will be part of a network of information and services that includes your home and office, other drivers, and perhaps the local police agency.

This chapter discusses areas for further research. The topics are arranged roughly in order of increasing scale, ranging from the connection between the Back Seat Driver and other systems in the car to the connections with Federal law.

8.1 Integration with the car

The Back Seat Driver should be running on a computer embedded in the car, so that it can get more and better information about the state of the car and driver.

For instance, when the next instruction is a turn, the Back Seat Driver should notice whether and when the driver turns on the turn signals. If the driver applies them too soon, it is possible (but not certain) that the driver has underestimated the distance to the turn; if applied at the "right time" then the system can take that action as confirmation that the instruction has been understood; if never applied, then the driver has either misunderstood, or is driving hazardously[1].

The Back Seat Driver should also be integrated into the car's audio system, rather than having separate systems for voice and music. Furthermore, it should pay attention to what the driver is listening to. If the driver is listening to the radio, or playing a CD (or using a cellular telephone) the program should try to speak less often, on the grounds that the driver has implicitly indicated a preference for what to listen to. The program should suppress reminders and historical notes altogether. When it must speak, it should borrow the audio channel rather than trying to speak over it. The Back Seat Driver should also be aware of the driver's use of other controls in the car. It should defer speech while the driver is adjusting e.g. the heat or the mirrors, and suppress speech altogether if the car makes sudden extreme changes in velocity. A driver trying to cope with an emergency situation does not need another distraction.

Back Seat Driver should also learn about the performance of the car. A suitably instrumented car could also measure the coefficient of friction by comparing the applied braking force and the resulting deceleration. This would allow it to adjust the time factors used in deciding when to speak.

8.2 Knowing the driver

While it may be so that "all men [meaning persons] are created equal", it is not

[1]or the signals may be broken

the case that all persons should be treated alike. Car seats, mirrors, and suspensions are adjustable now, and the Back Seat Driver should also be adjustable, or better yet, it should adjust to you. The Back Seat Driver should know about the knowledge and desires of its driver, and act differently because of this knowledge. There are various ways it could act differently.

8.2.1 Improving the directions

If the Back Seat Driver knows what you know about the city, it can give you better directions by using what you know. For example, to a driver who has often traveled up Mt Auburn Street to Watertown Square, and who now wants to get to Belmont Street, it can say "Remember that fork on Mt Auburn, just after the Star Market, where you bear left to get to Watertown Square? Well, Belmont street is the road that is the right branch of that fork." A user who knows about a city no longer needs instructions, she needs information about structure. The object description system TAILOR describes mechanical devices with two different strategies. Novice users hear a process description which emphasizes causal connections, and experts hear structural descriptions. Experts do not need the causal information, they can derive it for themselves[63]. The analogy to route descriptions is that process information is instructions.

Knowledge of the user's knowledge is crucial if the Back Seat Driver is to give orienting information as described below. It would be horrible to have the car give the same lecture about the history of Beacon Hill every time you drove over it.

8.2.2 Learning about the driver

The Back Seat Driver should acquire a model of the user by itself, without asking or having to be told. For properties which change only very slowly (e.g. the user's login name on the mail computer) it is more acceptable to ask the user for a value. Properties such as color-blindness, or visual or aural acuity do not change quickly. Rapidly changing properties must be learned automatically.

The Back Seat Driver could learn the driver's reaction time by measuring the time between its speech and the driver's operation of the controls. Renault has developed a prototype car which attempts to tell whether the driver is falling asleep by observing the correlation between small motions of the steering wheel and the attentiveness of the driver [84]. This system does not use a generic model of users, but rather learns the correlation for the driver during a registration period.

The the Back Seat Driver should learn the style of instruction giving appropriate for the driver. At present, there is not much to learn other than the settings for the parameters in the user model. In future work, the Back Seat Driver should learn not only these parameters, but also acquire rules (e.g. the rules of classification and obviousness) for the user. To learn the driver's preferences for route description requires either observation of the driver herself giving instructions (applying the Golden Rule – "Give instructions unto others as you would have them instruct you") or getting feedback from the driver about the instructions the system provides. One opportunity for learning the driver's style is during the acquisition of speech recognition templates (for user-dependent speech recognition). The new user should play the role of a "back seat driver" and give instructions, while in a car, for some route. The instructions must be given while driving either a real car or a close simulation (such as an "Aspen" movie map) because the form of static driving instructions is much different from real time instructions. Given some a priori knowledge about the ways that a route can be described, it is not impossible that the system could understand the instructions, and infer style from it. A difficulty here is that if the driver knows the route well, many things will seem obvious to her that would not be obvious to another person.

The driver can provide feedback about the suitability of the Back Seat Driver's instructions either explicitly or implicitly. One explicit indication of comprehension is how often the driver hits the "what now?" button. The system might collect long term statistics on the types of intersections where the user most often requests help, and begin to offer instructions without being asked. Just as the user can ask for more talking with the "what now" button, the Back Seat Driver should provide a "shut up" button (or recognize spoken words to the same effect). The Back Seat Driver could also learn the effectiveness of its directions simply by watching the driver's performance - in particular, her errors. Perhaps it can learn which cues are most effective.

The Back Seat Driver should understand the driver's plans and goals. This is already partially true. When you enter a destination address, you also tell the system that you have the goal of getting to that address. The Back Seat Driver might guess at higher level plans from knowledge about your destination, and take actions to assist you with those plans. To do this, it must know what kind of thing is at your destination address. For instance, if the address you provide is that of a store, the Back Seat Driver can guess that you are going there to purchase something, or at least to do business there. It could can check the hours that the store is open.

8.3 Confidence

People will only use the Back Seat Driver if they trust it. The best way to keep a user's confidence is to never make a mistake. But the Back Seat Driver does make mistakes, and it will make mistakes in the future. What can the system do to regain the user's confidence after an error? It depends on the reason for the error. The most likely causes for error in a future Back Seat Driver are navigation errors and database errors.

When differential GPS is available 24 hours a day, the Back Seat Driver can expect position accuracy to better than two meters. But until that time, there will be navigation errors. The Back Seat Driver will need to model the uncertainty of a position. For instance, when two roads diverge at a narrow angle, it will be unable to distinguish which was taken until some distance passes. It should probably assume that the driver made the correct choice rather than taking the risk of making a false accusation. If it reaches a place where the difference is crucial, yet unknown, it is probably better for it to confess its uncertainty, and say something like "I'm not quite sure where we are, but if you can take a right here, do it, and if not, keep going, and I'll figure things out better in a minute." Or it might ask the driver to pull over and stop (if the driver is at a place where that is safe) to allow for a better position estimate by averaging a few successive estimates. Navigation inaccuracy due to Geometric Dilution of Precision (GDOP) (page 130) is predictable. GDOP from LORAN is function of position alone. GDOP with GPS depends on the positions of the satellites, but position is predictable. The Back Seat Driver could even warn the driver that a planned trip is better deferred because of poor conditions.

Errors in the database will also persist because the database will always be somewhat out of date. The Back Seat Driver can regain at least a little confidence by how it explains the mistake. Suppose that the Back Seat Driver is intending the driver to turn onto "Apple" street, and says "Take a right at the next light". Unbeknownst to it, a new traffic light has been installed at "Pear" Street, so the driver turns there. The Back Seat Driver at present says "I meant for you to go straight." This message is somewhat confusing, because the driver may think that the program has not been consistent. ("First it said turn, so I did, and then it said I should have gone straight.") A better message would be "I did not mean for you to turn onto Pear. I thought that the next set of lights was at Apple Street."

8.4 Speech interface issues

There are two issues concerning speech and the the Back Seat Driver. It would be desirable to make its speech more easily understood, and speech input should also be investigated.

8.4.1 Speech output

Synthetic speech is more difficult to understand that natural speech or digitized speech. Some users complained about speech quality, and others did not understand what they heard. The most common error was to misunderstand the name of a street. This is to be expected, since there are plenty of redundant cues to the meaning of a sentence, but none at all for a name. With practice, people become more able to understand synthetic speech, but even those with a great deal of experience (such as the author) still misunderstand words.

It might be possible to obtain more intelligible speech by using digitized speech rather than synthetic speech. Digitized speech will require a great deal of storage space. There are more than 2000 different street names in Boston. Allowing another 500 words for the actual instructions, and assuming an average duration of .5 seconds for each word, coding this vocabulary at 64 kilobits per second would require 10 megabytes of speech storage. Given that a future Back Seat Driver will use a CDROM for the map, there should be no great difficulty in storing digitized speech on the disk as well. Coded speech would be more intelligible than synthesized speech, and less costly, but not as flexible. It would be impossible to read electronic mail using only stored vocabulary speech.

8.4.2 Speech input

The Back Seat Driver would be much easier to use if you could simply talk to it instead of using the cellular phone keypad. Back Seat Driver could use speech recognition for entering addresses instead of spelling names with a keypad. Address entry is a difficult task for speech recognition for the same reason it is hard for a human to understand machine speech – there are few constraints on a name. No speech recognizer available today can handle a 3000 word vocabulary with acceptable error rates. The car would also have to stopped while the driver was speaking, because noise in moving cars for frequencies below 400 Hz can exceed 80dB[62].

Back Seat Driver could also use speech recognition to replace the "what now?" and "what next?" buttons. This is a more tolerant application for speech recognition because there are fewer words to recognize. It may be possible to dispense with word recognition altogether, and simply recognize paraverbals, as in [76]

8.5 Understanding the route

The Back Seat Driver should help drivers to understand the route it gives. This would make the system more pleasant to use. Some drivers complained about being told what to do without any overall presentation of the route. It would also make it easier to follow routes because a driver who understands the route and the city will use that knowledge to help interpret the commands Back Seat Driver gives. For example, a driver following a route that crosses the Longfellow Bridge and who knows that Main Street leads to the bridge knows about how long she'll be traveling before her next instruction, and she'll be attentive to cues at the right time, instead of having to constantly be on the lookout.

The Back Seat Driver already provides a little information about the route. While following a route, a driver can hit a button and hear the total length of the route and the remaining distance. A second button provides the name of the street the car is on (often hard for the driver to know) and the compass direction (almost always a surprise in Boston). But this is just a beginning.

A route should fit into a larger model of the city. This means that the Back Seat Driver itself must have a model of the city and should speak of the route in terms that relate it to the city. There are several opportunities to do this. At the beginning of the route, the driver might hear an overview of the route, naming the major paths followed and neighborhoods crossed. During the route, locations could be described not just as street address but in larger units of neighborhoods and districts. The system might say not, "You're at 900 Mass Ave." but rather "We are now half way between Central and Harvard Squares." Orienting information can be included in instructions, or it might come between instructions, as a passing comment.

Benjamin Kuipers presents a formalization model of people's ability to learn to navigate[46, 47]. In his model there are three stages of spatial knowledge:

- Sensorimotor Procedures: knowledge expressed as conditional procedures: "When you see this, do that"
- Topological Relations: containment, connection, and order
- Metrical Relations: distance, direction

A route expressed in sensorimotor form is a sequence of cues and things to do. This is sufficient to follow a route, but does not support reasoning about the route. You can not reverse the route, because each cue leads only to the next, in the familiar order, and not the preceding. You may not even be able to give the route to someone else, as each cue may be recalled only in the context of the one just previous. This accounts for the familiar "I can take you there, but I can't tell you how." Topological knowledge tells you, for instance, that Massachusetts Avenue enter Cambridge from Boston by crossing the the Charles River at the Harvard Bridge, then passes MIT, Central Square, Harvard Square, and Porter Square, and continues into Arlington. Topological knowledge tells you that if you travel west from Somerville you are certain to eventually hit Massachusetts Avenue. It is knowledge of topological structure that gives us the confidence to explore, knowing that we will eventually cross a known feature, so we can always find our way home. Topological knowledge is essential to feeling at home in the city. Metrical knowledge is absolute knowledge of distance and direction. The Back Seat Driver has near-perfect metrical knowledge of the city – it can specify the distance and direction of any point from any other. Metrical knowledge is essential for efficient route finding. People's mental maps of the city usually have correct topology but distortions in metric structure.

Sensorimotor knowledge is the first to be acquired. The instructions the Back Seat Driver gives are sensorimotor instructions - they have to be, since only sensorimotor commands are directly executable. Drivers first learn routes as sequences of sensorimotor instructions[2], and gradually acquire topological and metrical knowledge. (Kuipers' program is a model of how this learning occurs.)

A better version of the Back Seat Driver should help people acquire topological and perhaps metrical knowledge of the form of the city. People learn from experience, but they can also use other sources of knowledge. People learn metrical relations partly by seeing maps, so it is possible that they can learn topological relations by being told. This is the justification for including orienting information

[2]I do have a concern that people who depend completely on the system for decisions about which way to turn may never acquire this knowledge, since they would not use their own judgments. In my experience, I learn a route much faster when I'm the driver than when I'm a passenger. Partly this must be because the task of driving forces me to pay more attention to where I am, but also there must be an effect from trying to remember a previous route, or trying to interpret instructions or a map. It would be a pity if the Back Seat Driver actually impaired learning.

in Back Seat Driver. It is also the only reason to show a map. Maps may not be a good way to give directions, but they are an excellent way to show the topological and metrical structure of the city.

8.6 Other directions

The Back Seat Driver helps you get from where you are to some other location. There are some related services that the Back Seat Driver could easily provide.

- It should be able to give the location of a place without giving directions, e.g. the description of Belmont Street in the example above.
- It should be able to give the directions all at once (as does Direction Assistance), and
- It should be able to give directions between any two places.

You probably will not often want these services, but it is easy for the Back Seat Driver to provide them. You might want them because you want to tell someone else how to get to where you are. How frustrating it would be to know that your car could give you the information, but will not.

8.7 Changes to the map

A problem for a practical Back Seat Driver is how to keep the map accurate, since the street network is constantly changing. Over time, new streets are constructed, old streets are renamed or closed. These kinds of changes are predictable, slow, and long-lasting. Other changes are unpredictable, quick, and transient. A road may be closed for repairs for the day, blocked by a fallen tree, or full of snow. Such changes are usually short lived. Only disasters such as landslides or earthquakes have effects that persist more than a day.

Suppose you are getting in your car for your daily commute. This morning, a road you usually use is closed for repairs. Although you do not know this, the the Back Seat Driver does. When you start the car, it can tell you about the closure, and offer to find a different route for you. The most likely means of distributing such changes is to broadcast them on a data subcarrier on FM radio[27]. Such changes might be broadcast once per hour. Your car would be listening even when parked, and could store the changes in local memory.

8.8 Integration with the city

Back Seat Driver should part of a larger system, linking the driver of the car to her home and office, favorite service station, and other cars. When you drive into the city your car should call ahead to parking lots near the destination, find one with an empty space, and make a reservation for you[3]. Your car should keep track of its own health and arrange maintenance appointments with your favorite service station. When you are driving home, your car should call home to see whether anything is needed, and if so, give you a route that passes by a store. Research in Japan on the Advanced Mobile Traffic Information Communication System (AMTICS) might lead to such integration. The committee is sponsored by a consortium of the national police agency and manufacturers of automobiles, telecommunications, and electronics equipment, and is developing standards for communications protocols, database formats, and interfaces between human, telecommunication equipment,

[3]The designers of ERGS anticipated this as early as 1970.

and computers. A European program, Prometheus (PROgraM for an European Traffic with Highest Efficiency and Unprecedented Safety), has similar aims.

When cars have positioning systems, it will become useful to the public if these cars can transmit their positions and routes on request. If enough cars report their positions and speeds, then a central agency can estimate traffic flow. This information will be useful to route planners in the cars. Furthermore, the agency can attempt to improve the traffic flow by balancing load. This could save a lot of money. Widely available positions and routes would also be useful in giving directions. If the Back Seat Driver knew that the car just ahead of you was about to make a turn it wanted you to make (by asking the navigation assistant on that car) then it could give you directions by just saying "Follow that car".

8.9 Policy

Before manufacturing a Back Seat Driver, there are public policy issues that much be addressed. There is a question of liability, and also some issues of policy on privacy. I do not have any answers to these issues, but it is important that they be raised.

8.9.1 Liability

Sometimes the Back Seat Driver gives orders that should not be obeyed. It might make a mistake, and send the driver down a one-way street, or it might just tell the driver to "keep going" when traffic is in the way. Drivers who become accustomed to obeying the voice without thinking for themselves will have accidents.

The most likely source of error is the map database, which can easily be inaccurate. Manufacturers of maps are subject to contract liability law and negligence law. The latter is the greater threat because of the greater potential size of damage awards. According to [26]:

> The manufacturer/seller may be held responsible for any injury to the buyer, or to any other user, or to any foreseeable bystander while the device is being used. The injuries may include direct physical or economic injuries as well as indirect, intangible injuries. In addition, there is the possibility of punitive damages as a penalty for carelessness.

Furthermore "there is little that a seller can do to protect himself from frivolous negligence suits". Once maps (or systems using them) are mass produced, they become subject to product liability law, which has the same potential for damages, and a lesser burden of proof:

> The buyer/user only needs to prove that he was injured, and that the defect caused his injury. The manufacturer of the product becomes liable to the same extent as if direct negligence had been proved....there are very few defenses that are successful in strict liability cases.

Rogoff[69] proposes that the federal government issue electronic maps on the grounds that the government can not be sued for liability.

8.9.2 Privacy

A second issue is the question of privacy. The Back Seat Driver knows if you drive too fast or fail to stop at a stop sign. What should it do? Should it automatically slow the car down, sound an alarm, or just record the violation silently, and later tell the insurance agency – or the police? Will automotive companies seek to record engine statistics to verify that the driver treated the car gently during the "break in" period?

The Back Seat Driver might remember every place it has ever been and the time it was there – there are good reasons for it to do so – but this record might be something the driver would not want anyone else to know about. Sometimes drivers convicted of drunken driving are allowed to keep their licenses, but only for driving to and from work. It is easy to imagine a demand for using a position tracking system to verify compliance with this restriction, since some jurisdictions already use personal locators for similar purposes. Regulations are needed to ensure that the trip history in your car is your private data, not to be inspected without a search warrant, and to require that all vehicles be able to take trips "off the record". These issues become even worse if cars broadcast their positions or routes, as proposed above. I am not at all sure how to resolve the conflict between privacy and utility. I leave it as a problem for the reader.

Chapter 9

Summary

The Back Seat Driver is a working prototype of a intelligent automotive navigation assistant. The important differences between the Back Seat Driver and other such programs are that the Back Seat Driver

- finds routes for the driver, instead of simply displaying position on a map,
- tells the driver how to follow the route, step by step, instead of just showing her the route, and
- speaks its instructions, instead of displaying them.

Each of these design goals has required new features in the Back Seat Driver or its street map database:

Finding legal routes requires that the street map database include legal connectivity in addition to physical connectivity. Finding routes that are fast requires knowledge of expected travel rate, which includes quality of the street and the presence of traffic control devices such as traffic lights and stop signs.

Giving instructions for route following requires breaking the route down into a sequence of driving acts: it requires a taxonomy of these acts, and knowing when an act is obvious to the driver and when it needs to be mentioned. This further requires knowledge about the individual driver, for what is obvious to one may not be so to another.

Speaking instructions requires generating English language descriptions of actions. These instructions tell the driver what to do and when to do it. Instructions must expressed in terms familiar to drivers – in particular, they must use timing and landmarks, rather than distance, to tell drivers where to act. This in turn requires that landmarks be present in the map database. The use of natural language, in contrast to displays, affords the ability to refer to past, present, and future places and events – thus the Back Seat Driver may refer to past mistakes, current actions, and future landmarks with equal ease. It would be more difficult to convey this information by graphical means, moreover it would require that the driver look away from the road to gain navigation information.

Speech, especially synthetic speech, as an output media imposes constraints on the interface. First, the transient nature of speech requires that the program be prepared to repeat its utterances on demand, which means the ability to construct a new utterance with the same intent, not necessarily the same words, as a previous message. Synthetic speech being sometimes hard to understand, the program must choose its words to provide redundancy in its utterances.

Although the Back Seat Driver is working now, much work remains before we can expect to see a system like it used by ordinary people, whether they are tourists, commuters, or taxi drivers. Automakers remember the public rejection of earlier "talking cars", which spoke only of unlocked doors and empty fuel tanks. Nobody wants a nag in the car, and therefore a Back Seat Driver must know about the knowledge, desires, and goals of its driver, so that it will know when to speak, and **when to be silent**. This knowledge will be difficult to acquire, but if it can be done, our culture may come to have more favorable meaning for the phrase "Back Seat Driver".

Appendix A

Location Systems

A navigation assistant such as the Back Seat Driver has to know the position of the vehicle. In this section I present a review of the available technology for determining location. To motivate this, I begin with a discussion of the requirements of the Back Seat Driver.

A.1 Accuracy requirements of the Back Seat Driver

At the minimum, a direction giving system must determine its position to the nearest block. If it is to tell the driver when to turn it must be able to distinguish between the closest two adjacent turns. Hall estimates that a 50 meter accuracy is sufficient because he takes the average length of a block to be 100 meters[32]. Pilsak demands 25 meter accuracy and 10 degrees of heading resolution to select among narrowly diverging intersections[65].

Consideration of the Boston street map shows that it has many streets which are both short and a possible choice point. Figure A-1 shows the percentage of segments which are shorter than a given length. Only segments with at least one choice point are counted here. If the Back Seat Driver has only 25 meter accuracy

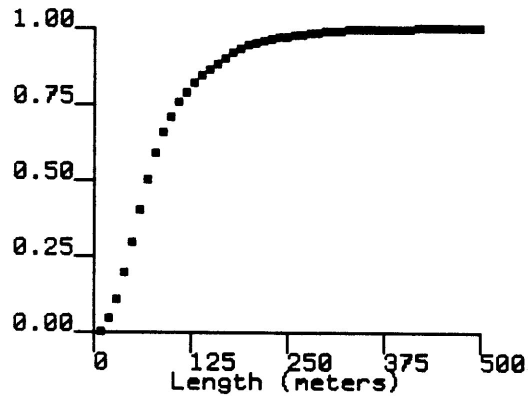


Figure A-1: Fraction of segments shorter than X in meters

then it will not resolve more than one fourth of all streets. An accuracy of 10 meters is more desirable.

The Back Seat Driver's use of speech imposes more strict requirements on position because of limitations on time. Unlike a display, speech is transient. An action described too soon may be forgotten. The Back Seat Driver is intended to speak at the latest time that still permits the driver to act. Allowing two seconds for speech, a car at 30 mph covers 27 meters. This suggests a minimum accuracy of 15 meters.

A.2 Classifying location systems

Following Cooke[15], location systems can be divided into two categories:

- **Position finding** systems determine position directly by detecting an external signal.

- **Position keeping** (or dead reckoning) systems estimate the current position from knowledge of an earlier position and the change in position since that position.

A.3 Position finding

All existing position finding systems use radio signals. The broadcast stations may be located on street corners, seacoasts, or in orbit around the earth. Systems differ in their range, accuracy, and cost. A survey of positioning systems organized by operating frequency appears in [74]. That survey includes many specialized, limited range systems. The list here includes only those systems which might plausibly be used in for automobile navigation.

A.3.1 LORAN-C

LORAN (LOng RAnge Navigation) is a navigation aid originally intended for ships, now extended to civil aviation and land navigation. It relies upon measuring the difference in arrival time between signals transmitted simultaneously from three separate known locations. If the receiver is equally far from two transmitters, the signals will arrive at the same time. If the receiver is closer to one, that signal will arrive first[1]. A measurement of the inter-arrival time for a signal from two positions fixes position to a hyperbolic line, called a line of position (LOP). The line of position takes the shape of a hyperbola because the locus of points with a constant difference in distance to two given points is a hyperbola. The estimated position is the intersection of two lines of position.

[1]This is a simplification. LORAN transmitters do not really transmit simultaneously, the receiver compensates for this.

The accuracy of a LORAN position fix depends upon three categories of factors: geometry, radio propagation, and equipment. The significance of geometry is known as Geometric Dilution of Precision (GDOP) and is illustrated in figure A-2.

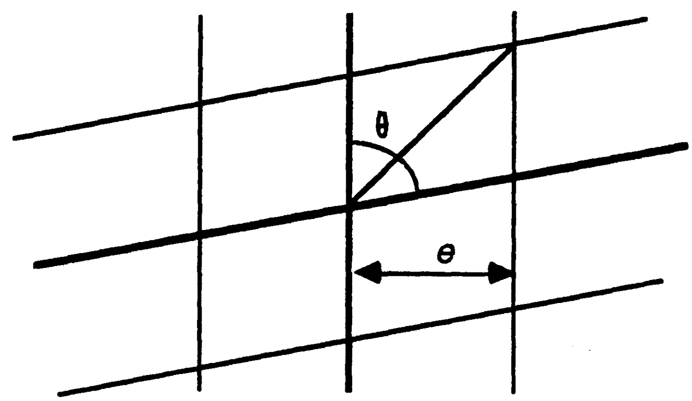


Figure A-2: Geometric Dilution of Precision

The placement of the LOP is subject to some error. If the maximum size of the error is e, then the true LOP will be parallel to the measured LOP, and at most a distance e from it. The two LOPs might meet at any angle, depending where the receiver is. If the LOPs meet at right angles, errors in each are independent, but otherwise the errors add. The maximum error, in the case where the error in each LOP is e is the length of the base of the isosceles triangle whose sides are e. This is $2cos(\theta/2)$, where θ is the angle between the two lines of position. So when the LOPs meet at right angles, this is $\sqrt{2}$.

Radio propagation affects LORAN by causing erroneous time differences, thus spurious position errors. Factors affecting radio propagation include weather, season, time of day, and signal path. Radio propagates more slowly over land than over water, and is reflected or delayed by topography and man-made features[10].

Equipment conditions at transmitter and receiver can also introduce false time differences. Dutton estimates a cumulative error of 15 to 90 meters when within 200 miles of the transmitting station.

Accuracy can be improved by means of differential LORAN[10]. A differential LORAN system requires setting up monitoring stations at fixed, known positions. These stations compare the LORAN estimated position to the known position to obtain instantaneous corrections to the radio signal. Mobile LORAN receivers can use these corrections over radius of approximately 35 miles. Blizard estimates that differential LORAN can provide repeatable accuracy to ± 20 meters. Cooke cites an accuracy of 5 meters.

LORAN is used in at least two commercial vehicle location systems, the Vehicle Tracking System sold by IImorrow, Inc, and the Automatic Vehicle Location system of Motorola[73]. IImorrow claims an accuracy of 22 to 36 meters for their system. Janc (the patent holder for the Motorola system) provides an account of sources and the worst case magnitudes of errors for LORAN, and measures a 500 meter accuracy with 500 meters with a 95% confidence in [39]. El-Sawy reports on a LORAN vehicle tracking system in [23]. He does not give accuracy figures, but says "Some system operators have voiced concern over the inconsistency of the vehicle tracking in some geographic areas". In rural areas without a complicated signal path, McGillem obtained accuracies of 45 meters[55].

A problem that remains is that LORAN does not completely cover the continental United States. The system is being expanded on behalf of civil aviation users and should be complete by 1990[78].

A.3.2 Polled pulse time ranging

In LORAN, vehicles determine position by comparing arrival time difference of signals from three (or more) transmitters. An alternative is for three or more fixed receivers to compare the arrival time of a signal sent from the vehicle. In 1977 Hazeltine developed a system which does just this. A central site broadcasts a prompt, and transponders on vehicles echo the prompt. The obvious drawback is that there must be a way to distinguish each vehicle's response. The Hazeltine system uses time multiplexing. The vehicles do not respond all at once, but rather one at a time. Each vehicle has a time slot for its response which lasts long enough that any previous response will have been received by the time the vehicle transmits. In the Hazeltine system, time slots are two milliseconds long, so 30,000 vehicles can be tracked in each minute. The system was tested in Dallas[66] and Philadelphia[11] and achieved accuracy of 300 feet within a 95% confidence.

A.3.3 GPS

The Global Positioning System (also called **NAVSTAR**) derives position by measuring the travel time of (and thus distance to) radio signals from satellites orbiting the earth. The system is incomplete at present, but will eventually consist of 24 satellites. A GPS receiver can determine both position and velocity in three dimensions, providing that four satellites are visible at the time. If elevation is known (or independently measured) then only three satellites are required.

GPS provides two classes of service. The Precise Positioning Service (PPS) offers accuracies to eight meters, but is only available to military users. The Standard Positioning Service available to the public provides only 200 meter accuracy. One satellite can provide two levels of service by broadcasting position and time information at two different levels of accuracy, using a different code for each. SPS information is encoded with the Coarse Access (CA) code which has been made public. PPS information is available only to those who can decode the P code.

GPS accuracy is influenced by the same factors that influence LORAN: geometry, propagation, and receiver, and by others. The geometric effects are more complicated because GPS satellites move. The GDOP for a LORAN fix depends only on one's position on the Earth, but the GDOP for GPS changes over time. The problem of unpredictable propagation is also worse because the signals travel through several different regions of the atmosphere. There is a well accepted model for ionospheric propagation. The GPS signal includes parameters for this model, enabling the receiver to compensate for the effects. The effects of the troposphere are less well understood. GPS satellites transmit on two different frequencies, L1 (1575.42 MHz) and L2 (1227.6 MHz). The L1 channel carries both CA and P code, the L2 carries P only. PPS receivers can compensate for the effect of the troposphere by comparing the effect on L1 and L2, since this effect depends upon radio frequency. GPS accuracy also requires that the satellite orbits be predictable and stable. Under normal conditions, this is so, but intense magnetic "storms" can raise the outer atmosphere sufficiently that satellites feel a noticeable drag. The geomagnetic storm of 13 March 1989 caused so much drag that GPS tracking was unavailable[34].

As with LORAN, differential GPS is possible[40, 9]. It should be possible for a GPS receiver at a fixed, surveyed location within the city to measure the errors from propagation delay and broadcast correction factors, perhaps over single sideband FM, to receivers in cars. Experiments with differential GPS have shown accuracies of two to three meters using the C/A channel[45, 2, 3]. At least one expert claims that sub-meter accuracy is possible[7]. Surveyors using so called kinematic positioning can obtain even better accuracy but require tens of minutes to do so. It is unclear whether this accuracy can be obtained in real time.

A.3.4 TRANSIT

The Navy Navigation Satellite System system, also called **TRANSIT**, is the older of two satellite navigation systems. NAVSAT differs from GPS in that only a single satellite is required for a position measurement, instead of three or four for GPS. This is possible because NAVSAT satellites travel in a much lower orbit (approximately 600 miles) than GPS, and thus move faster, covering a larger distance during a fixed time. A single NAVSAT position measurement requires about 10 minutes. In this time, the satellite moves more than 2000 miles thus providing a large baseline for calculation. Equipment at the receiving station measures the Doppler shift in the received signal to obtain a measure of the satellite's velocity relative to the ship. From this, and the knowledge of the position of the satellite, the receiver's position may be measured. Under ideal conditions, a NAVSAT position fix is good to within about 45 meters.

NAVSAT alone is not sufficient for car navigation because of the long time to obtain a position measure and the long time between satellite orbits. Even under ideal circumstance, one can obtain but one position measurement every 90 minutes, but at times the period between useful satellite passes can be as high as 16 hours ([80]p 190). The vehicle's velocity must be carefully measured while obtaining a position fix. An error of as little as one mile per hour can cause a position error of up to 1/4 mile in the position obtained. NAVSAT might serve to supplement a position keeping system. It is unlikely to do so, because it is being shut down in 1994.

A.3.5 Other satellite systems

The Soviet Union's satellite positioning system, GLONASS, is quite similar to GPS[16]. Like GPS, it consists of a highly accurate, encrypted signal and a less accurate clear signal. Details on the clear signal have been released and tests in the West show accuracy similar to that of GPS[17].

The STARFIX system[20, 61] operates in a manner similar to GPS, but uses geosynchronous communication satellites rather than dedicated navigation satellites. A ground station sends an "event" message to the satellites, which relay it to STARFIX users and to reference stations. The reference stations broadcast a local differential correction to users in their area. Tests have shown an accuracy as good as 2.5 meters, degrading to 30 meters at maximum range from a reference station (1000 miles). STARFIX appears to have suitable accuracy for direction giving, but present receivers are much too large (ten cubic feet) and heavy (300 pounds) for cars.

GEOSTAR[67] is a proposed, but not yet operating, commercial system to provide two dimensional position and two way digital communication to users in the continental United States. Unlike GPS, GEOSTAR users are not passive. The system determines location by broadcasting an interrogation to a user, which then transmits a response to at least two satellites. The satellites relay the response to centralized processing facilities on the ground, which determine the user's position by measuring the elapsed time for the signals. A network of benchmark receiving stations provide increased accuracy by differential techniques. Position accuracy is said to be within seven meters. Note that the mobile station does not know its position unless the central facility transmits it. Note also that since only two satellites are used, the user must supply the altitude.

The European Space Agency is studying an independent satellite positioning system, also called NAVSAT, which would provide 12 meter accuracy to its users.

A.3.6 Beacons

A less exotic alternative is to place radio (or infra-red) beacons or "signposts" at intersections. These provide position, or rather, an indication of proximity. Such a system was installed in Huntington Beach, California[29, 31]. In this installation, 482 transmitters were installed on utility poles approximately every quarter mile. Huntington Beach is ideal for such a system because it is laid out in a grid. The system provides position to 300 feet with 90% confidence on the major streets, and 600 foot accuracy on the lesser streets. The disadvantage of such systems is that they require a large investment in installing transmitters at each intersection.

Signposts can also be passive, and interrogated by the vehicle. Proposed systems include corner cube reflectors tracked by lasers[86], magnetic strips embedded in the road[48], and transponders which respond only when interrogated[41].

A.3.7 The cellular phone system might serve as a location system

I speculate that one might be able to use the network of cellular telephone base stations for a positioning system. This is possible because cellular phone sites already broadcast a status message on a regular basis. This is how cellular phones determine which frequencies are available and which carriers (telephone companies) are in operation. All that is required is that cellular phone sites be synchronized to a common clock. The positions of cellular base stations are well known (and can be known in the car), so it is a simple matter to measure distance to the cellular phone stations. In effect, this would make cellular site transmitters into earthbound GPS satellites. An alternative is to use polled pulse time ranging. Here, the cellular site would compute the position of the car and retransmit it[2].

[2] I do not know how to estimate the theoretical position resolution. The carrier is around 800 MHz. I do not know what the envelope of the carrier looks like when turned on, or how accurately the arrival time of the wavefront could be measured. If measured to the period of the carrier, that is $1.25x10^{-9}$ seconds, or about 40 meters, which is good, but not good enough. But could you measure phase and do better?

A.3.8 Vision

The defense department is sponsoring research on an Autonomous Land Vehicle. This vehicle will use machine vision to determine location. The research is far from ready for application, but we can anticipate a day when navigation systems will determine where they are by just looking out the window, in the same way that humans do.

A.4 Position keeping

Position keeping, also known as *dead reckoning*, obtains position indirectly, by keeping track of the displacement from an originally known position. One can measure displacement directly, or they measure velocity or acceleration and integrate over time to obtain displacement.

The forward motion of a car is measured by the odometer. On late model cars, the odometer cable has been standardized. It revolves once every 1.56 meters. This is a reliable measure of forward progress, as long as the wheels do not slip. Measuring direction, though, is more difficult. Among the possibilities are:

- **magnetic compass** A magnetic compass has the advantages of small size and ease of use[64], but is unreliable because of variation between magnetic and true north and deviation due to the ferrous material of the car and magnetic flux arising from electric currents within the car. For a discussion of the difficulties of magnetism, see [87].
- **steering wheel** The steering wheel could be instrumented to measure the amount of turning.

- **differential odometer** When a car turns, the two rear wheels travel different distances, and thus rotate at different rates. Measuring the difference in rotation should provide an indication of amount of turning. By lucky coincidence, this differential rate of rotation is just what is measured by anti-skid brakes, so no additional instrumentation is required to obtain this measure. There is no obvious reason why it should not be available on all new cars in the near future.

- **gyroscopic** Gyroscopes measure angular acceleration. The familiar rotational gyroscope and esoteric laser ring gyroscope are not suitable for automotive use because they are too expensive. Lower cost alternatives are the rate gyro and the gas jet gyro. The rate gyro measures angular acceleration in a vibrating piezo-electric substance[43, 25]. The gas gyro measures turn (or yaw) rate. In this design a jet of gas travels down the center of a sealed tube. Anemometers are placed on either side of stream. When the car turns, the stream is deflected and the velocity is measured. The velocity of the gas at the anemometer is proportional to the turn rate. Such devices can measure turn rates of up to 100 degrees per second, with a noise of about one half degree/second.

A.4.1 Dead reckoning systems need correction

A position keeping system with no error could be calibrated when installed, and then maintain its own position indefinitely. Unfortunately, errors arise in measuring both distance and heading. Sources for error include difference in tire pressure, composition and wear, slipping, cross steering from winds, change in tire contact patch in turns, magnetic anomalies, and gyro noise. The NEC dead reckoning system, for instance, accumulates about one meter of error for every ten

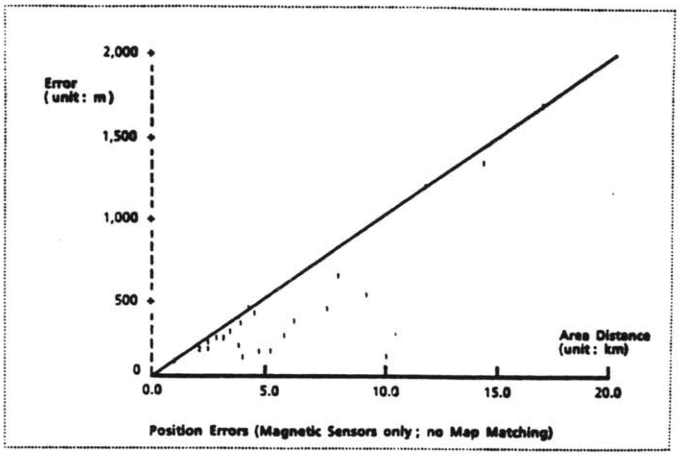


Figure A-3: Dead reckoning errors increase with distance

meters traveled (Figure A-3). The error is even worse when traveling near railroads because the NEC system uses a magnetic compass.

Some dead reckoning systems recalibrate themselves to eliminate systematic errors. Such recalibration is possible when the vehicle is at a known position or when stopped. For instance, if the ETAK navigator finds a consistent overestimation of distance it compensates for tire wear and it adjusts its magnetic compass while traveling on a road with known heading. The Honda Gyrocator adjusts its yaw rate sensor when stopped. Recalibration can also be applied to position finding systems. For instance, a car stopped at a known location could accumulate differential GPS or LORAN corrections on its own behalf.

One way to correct dead reckoning errors is to use knowledge of the road network as a *constraint* on position, in what is known as *map matching.* Map matching requires that the position keeping system have a map of the locale where the vehicle is being driven, and is based on the assumption that the vehicle is always on a street present in the map. If the estimated position falls to one side of the road, the estimate can be corrected. When the vehicle makes a turn, the system assumes the vehicle is at the closest intersection, and thus the absolute position can be estimated. Figure A-4 illustrates the effect of map matching.

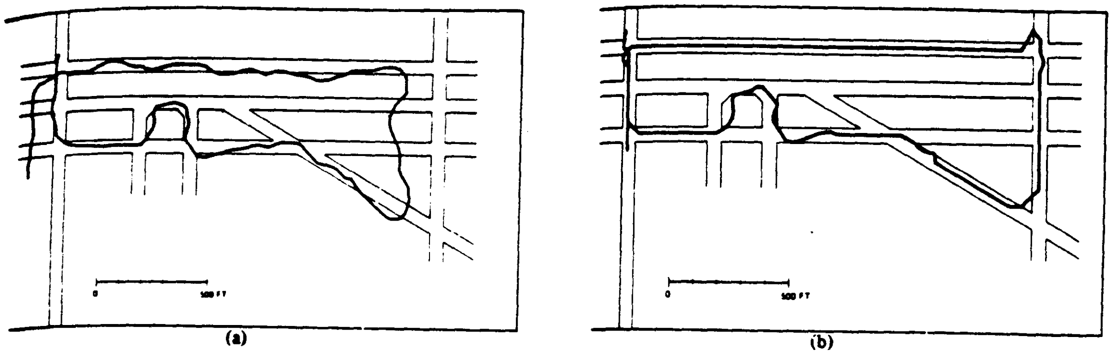


Figure A-4: Map matching corrects dead reckoning errors

Every dead reckoning system uses some form of map matching. Map matching reduces the stringency of position keeping, but accuracy remains a concern, since the system must maintain its position when the driver drives off the map, e.g into a driveway or a parking lot.

A.5 Hybrid systems

Positioning systems must combine dead reckoning and position sensing, since neither alone will work all the time. A position keeping system needs periodic corrections, but an absolute positioning system that depends on radio reception will not work in tunnels or bridges. Such hybrid systems are discussed in [59, 37, 43, 44, 38]. Karimi stresses that hybrid systems should not simply combine all knowledge sources, but should reason about the expected reliability of each one using knowledge of current road conditions, weather, location and so on. For example, when entering a tunnel, the system should no longer trust satellite positioning data, and it should also allow sufficient time after emerging to reacquire position. Itoh and others [38] shows the effect of construction and topography on GPS reception. Under good conditions (in open areas), they obtain positions accurate to within 30 meters, but when driving in urban or mountainous areas they find a GPS reception rate as low as 13% (contrasted with 75% when near the ocean). Although this may improve when the full GPS constellation is operating, it seems likely that hybrid systems will remain essential.

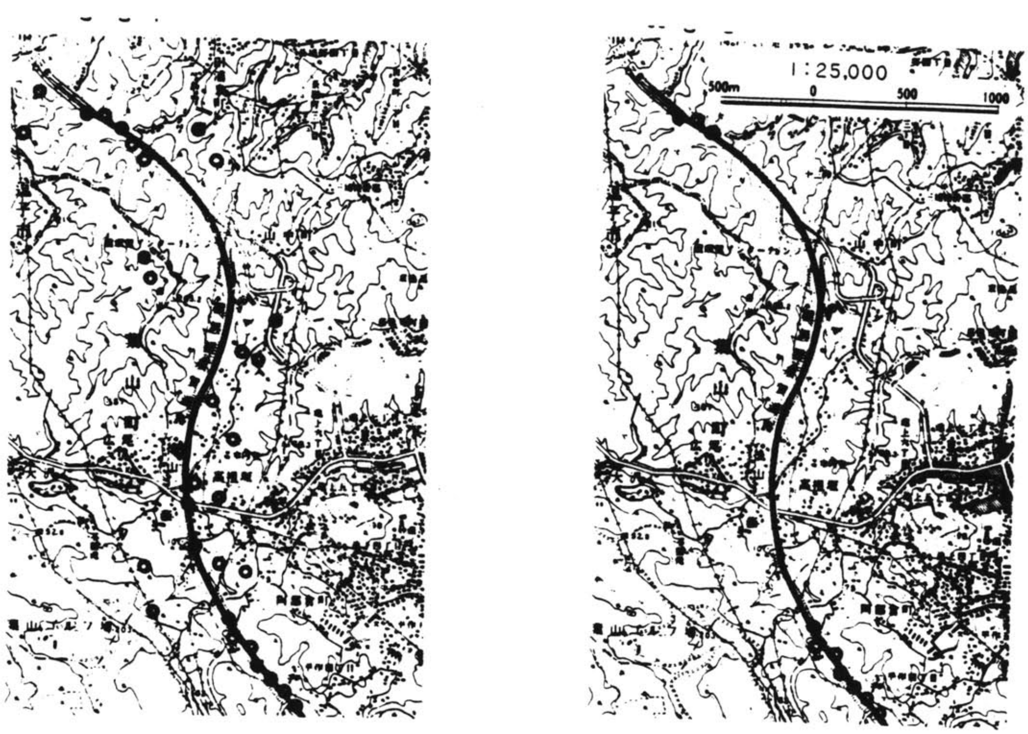


Figure A-5: Dead Reckoning corrects for loss of GPS signal

Appendix B

Route Finding

Finding a route between two points in a street network is an example – perhaps the best example – of searching a general graph[1]. The task is to find a sequence of segments that lead from the origin to the destination. There are usually a great many distinct ways of getting from one place in the city to another, some better than others. Ideally, we want the *best* route, not just any route. Graph search algorithms differ in the quality of the solution they find and the time they require. For the Back Seat Driver we want an algorithm that finds a good route, though not necessarily the best, and does so in a short time.

The Back Seat Driver uses an A* search algorithm[35]. The A* algorithm is a form of best-first search, which itself is a form of breadth-first search. This chapter begins with a discussion of breadth-first search, then shows how A* search is built from it.

[1]See page 99 for more on graph search.

B.1 Breadth-first search considers all possible partial solutions in parallel

Breadth-first search is named so because of the view of search as exploring a tree of all possible decisions. The root of the tree is the current state. Leading from this route are one or more arcs, each corresponding to a possible action. Each action brings about a new situation from which other actions are possible. A solution is a set of actions leading from the initial state to the desired situation. Search algorithms differ in the order in which they consider the possible actions. In a breadth-first search, the tree is divided into levels, where the first level actions are those leading from the root, the second level actions are those that come from situations after first level actions, and so on. All actions at a given level are considered before any at the next higher level. In this way, the search systematically covers the entire tree, skipping nothing.

In the case of route finding, the root state is the initial position, and arcs correspond to street segments leading away from the initial position. It is important to distinguish the search space, which is a tree, from the problem space, which is a general graph. Even when two paths on the map lead to the same place, they are distinct paths in the tree because they have different histories. After driving all the way around a block, you are in the same place as before on the map, but the search tree has become four levels deeper. The position is the same, but the situation is different. Time and distance have both changed.

While the breadth-first search is operating it maintains a list of all possible partial routes. (A partial route is a sequence of segments leading from the origin to some intersection.) The search procedure is a loop. Each time around the loop, the procedure considers each partial path. For each path, it considers each segment leading away from the intersection at the end of the path. If there are no segments leading from an intersection, then that path is a dead end, and can be dropped from the list of candidates. If the segment contains the destination, the search is complete. Otherwise, if the segment is not already present on the path somewhere, a new partial path is formed by appending the segment to the end of the path, and this partial path is collected into a second list. (Actually, the segment may be on the path twice, once for each direction of travel. The route finder does not make U Turns in the middle of the street.) After each possible path has been considered, if none have yet led to the goal, the list collected during the loop becomes the new list of possible routes. In this way the search systematically examines every possible path.

Figure B-1 shows an example of breadth-first search. At the beginning of the search, there is just one possible route, leading from the origin to the end of the current street segment. (This corresponds to driving to the end of the street, to the first intersection, in the direction the car is already facing. The Back Seat Driver never asks the driver to make a U Turn in the middle of a street, unless the street is either a dead-end or the driver is somehow facing the wrong way on a one-way street, because such moves are difficult and often illegal.) Here is the first choice point. The route can continue left or right. Neither of these segments is the destination, so there are now two possible routes to consider. In the next step, one possible path generates three more possible paths, and other just one (because the intersection is with a railroad, the only possible next path is to continue straight). Now there are four candidate paths. As the search continues, the number of possible paths grows quickly. Table B.1 shows how the number of potential routes increases at each step. After 1 minute, there are 11,257 possible paths under investigation, but only 1019 segments have been examined. Clearly most of the paths are overlapping. Breadth-first search is wasting too much effort by considering every possible path.

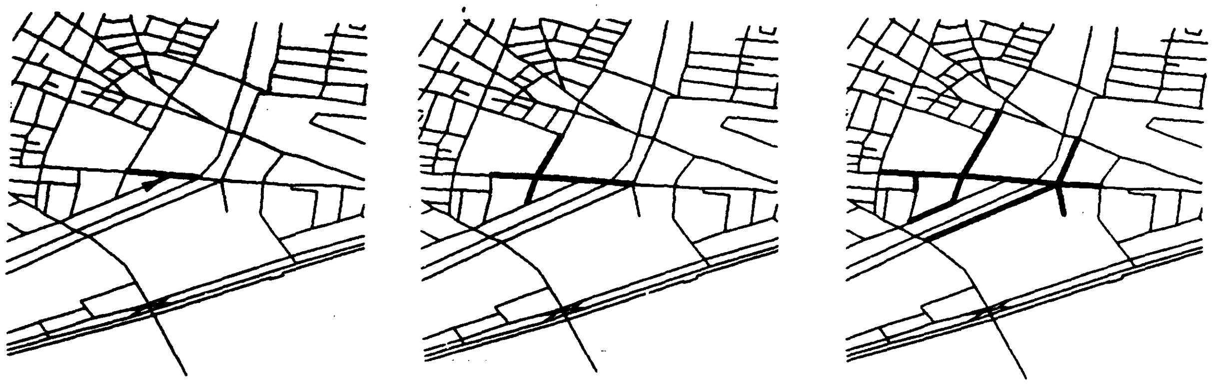


Figure B-1: Breadth-first search example

B.2 Best-first search saves effort

The search procedure given above finds the path with the fewest segments. This is not necessarily the *best* path. Suppose one takes "best" to mean shortest. It is possible (even likely) that there is a path with more segments which is also shorter. To sure of finding the best path, the search can not stop when the first path is found, but must continue, expanding each path, until *all* paths are complete. Only then can the best be found, by comparing each complete path. This is not at all desirable, since there could be (and in fact will be) many paths. The difficulty arises because the algorithm considers paths in order of their size, not their value. The best-first algorithm solves this problem by keeping track of the (partial) cost of each path, and examining the one with the smallest cost so far. This requires a function that can compare two routes and produce a numeric rating. Such a function is called a **metric**. To further reduce the cost of searching, before adding a segment to a path, the best-first search checks to see whether it is a member of *any* other path. If it is, it is not added, for presence on the other path means that

Step	Number of potential routes
1	1
2	2
3	4
4	8
5	19
7	33
7	59
8	117
9	220
10	397
11	697
12	1199
13	2085
14	3700
15	6469
16	11257

Table B.1: Breadth-first search considers many potential routes

that other path was a less expensive way of reaching the same segment. The other path must have been cheaper, since it was discovered sooner.

B.3 A* search avoids falsely promising paths

Best-first search finds the best solution and requires less time than exhaustive breadth-first search, but it can take a long time because it must consider partial solutions with an initial low cost which prove expensive when complete. The A* algorithm avoids wasting time on such falsely promising solutions by including an estimate for the completed cost when selecting the next partial solution to work on. The cost estimate function is $f^*(r) = g^*(r) + h^*(r)$, where r is a route, $g^*(r)$ is the known cost of the partial route, and $h^*(r)$ is the estimate of the cost to go from the endpoint of the route to the goal. The h^* function must have the property of being always be non-negative and never *over*-estimate the remaining cost. An h^* meeting these two conditions is said to be *admissible.* It should be obvious that if h^* is chosen to be always zero, then A* search is just best-first search. In applying A* to finding routes on a map, h^* is just the cartesian distance between the endpoint of the partial route and the destination point. It is certain that no route will be shorter than the straight line, so this estimate is never an over estimate. Table B.2 shows that A* search is more efficient than best-first. The two algorithms were compared on thirty different routes. The routes are identical, but A* took much less time, because it examined fewer potential paths.

	segments touched	time
best-first	355711	2435
A*	59803	455

Table B.2: A* search touches fewer segments, and is therefore faster

B.3.1 A suboptimal, but faster algorithm, is desirable

The A* algorithm finds the optimum route, but the Back Seat Driver might be better served with an algorithm that finds a reasonable route in less time. This is especially true when the vehicle is in motion. The longer the route finder takes, the greater the distance that must be reserved for route finding. As this distance becomes larger, it becomes harder to predict the future position of the car. We can do this by choosing an h^* which multiplies the estimated distance remaining by a constant D. Setting D greater than one makes h^* no longer admissible, since the estimate might exceed the actual cost by a factor of D. The resulting routes are no longer optimal, but are still pretty good. The effect is to make the algorithm reluctant to consider routes which initially lead away from the goal. Table B.3 shows the cumulative path length and search time for thirty different routes with four different values of D. As D increases, the path length increases, but the time decreases more rapidly. In this table *Length ratio* is the ratio of the path length to the length of the optimum path, *Time ratio* is the ratio of search times, and *Payoff* is the ratio of the change in length to the change in time.

Weight	Length	Time	Length ratio	Time ratio	Payoff
1	109.37	455	1.00	1.00	1.00
2	116.81	61	1.07	0.13	7.97
4	122.65	50	1.12	0.11	10.20
8	126.67	45	1.16	0.10	11.71

Table B.3: Comparative search times and route lengths for 30 routes with different values of distance weight factor

The route finder uses a value of 2 for D. This yields the greatest increase in payoff. A possible improvement is to run the route finder twice, first with a high value of D to find an initial route in order to begin the trip, and then with a low D to search for a better route, using spare time while driving.

Appendix C

Communication with the car

The Back Seat Driver is a prototype of an in-car navigation system, but it was actually implemented on a large workstation computer[1]. This computer is too large to fit into the car, so instead I used cellular phones to carry data from the car to the computer, and voice from the computer to the driver. This chapter describes the actual experimental setup. It is of little theoretical interest, but may be of practical value to others attempting to send data through cellular phone links. in this area.

The workstation communicates with the driver and the onboard hardware through cellular phones, as shown in figure C-1.

The position sensor estimates vehicle position, heading, and velocity, and sends a data packet, once per second, through the modem to the workstation. The workstation sends characters to the Dectalk speech synthesizer, which in turn sends voice over a second phone to the driver.

[1]a Symbolics Lisp Machine

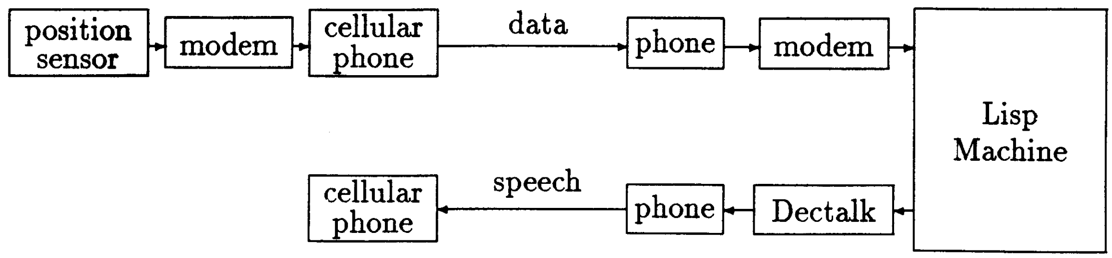


Figure C-1: Communications block diagram

C.0.2 Cellular phones are hostile for data transmission

Nearly everyone who has used a cellular phone knows how noisy they are. Cross talk is common. On several occasions I have heard one and even two other conversations at the same time. Noise bursts and signal loss make it hard to hear. A sufficiently bad noise burst can even cause the cellular system to terminate the call. The problems for data transmission are even worse[6]. By its very nature, cellular radio transmission is subject to multi-path interference, which causes periodic fades as the antenna moves in and out of anti-nodes. In addition to this fading, there is a complete loss of audio signal for as long as .9 seconds when the phone switches from one cell site to another (hand off).

My attempt to use an ordinary (land-line) modem from the car[2] was unsuccessful. Even at 300 baud the connection was too noisy to use. Worse, connections seldom lasted more than five minutes. In all cases, I set the "loss of carrier" register (S10) to its maximum value, 20 seconds. Loss of carrier signal alone can not explain why the connections dropped. The modems were capable of tolerating a complete loss of audio for up to twenty seconds.

[2]I used a Worldlink 1200 from Touchbase systems in the car, with a Morrison and Dempsey AB1 data adapter, and an NEC P9100 phone, boosted to 3 watts. At the base station I used both a Practical Peripherals 2400 and a Hayes Smartmodem 1200.

I had better results using an error correcting modem[3] made by the Spectrum Cellular Corporation. This modem uses a proprietary protocol (SPCL[60]) for error correction. The Spectrum product virtually eliminated noise, at the price of a lower data transmission rate. According to the protocol, the transmitting modem groups characters into packets that include error correction bytes. If only a few errors occur, the receiving modem repairs the data and acknowledges receipt. If there are many errors, the packet is retransmitted. If the sending modem has to retransmit too often it makes the packets smaller, on the assumption that at a smaller packet has a better chance of success. This is less efficient, since packets have a fixed overhead, the percent of the channel used by data decreases. When conditions improve the modem increases packet size again. In theory, the modem can transmit at 120 characters per second, but I estimate an average value closer to 30 characters per second. I made this estimate by recording the time required to receive the three characters of an odometer sequence. This sequence is transmitted once per second. Figure C-2 shows a histogram of durations for the three character sequence. The mean for this histogram is 94 milliseconds, which is 31 milliseconds

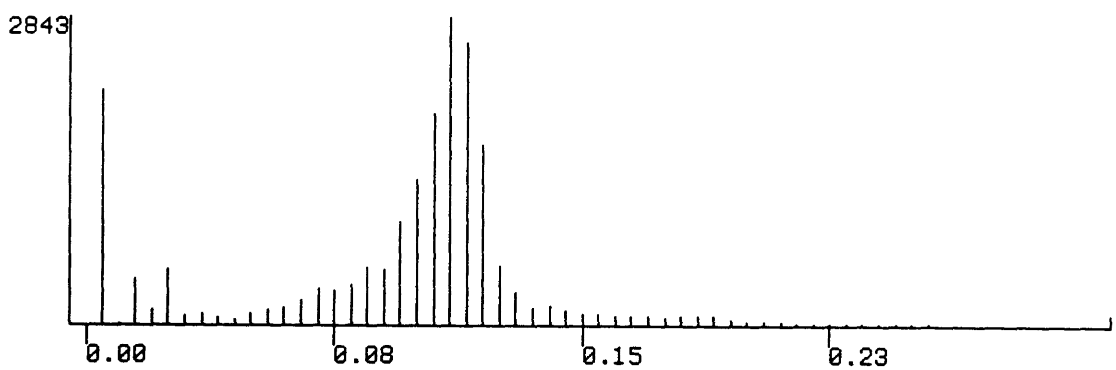


Figure C-2: Histogram of durations of odometer sequence

per character, or 32 characters per second. Tests by Fontana[24] found better results, between 75 to 80 characters per second.

[3]The "Bridge"

Even with the cellular modem, calls are sometimes dropped. Figure C-3 shows the probability of a call being dropped plotted against time. The measure of probability is obtained by comparing the number of calls dropped at or before time t with the number of calls that lasted at least that long. The call durations are usually long enough for a successful trip with the Back Seat Driver. Voice calls are dropped at about the same rate as data calls.

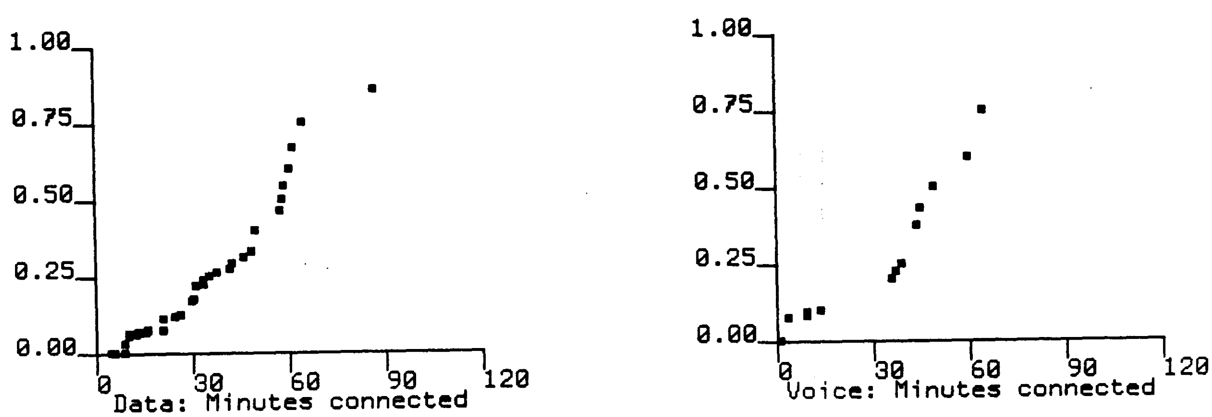


Figure C-3: Probability of cellular call termination increases with time

C.0.3 Retransmission introduces latency

The protocol used by the Spectrum modem acknowledges all data transmitted. If the acknowledgment is not received, it retransmits the data until acknowledged. Under adverse conditions, this can result in an arbitrarily long delay. This is a problem when real-time data is transmitted. Observation of the Back Seat Driver shows that sometimes the system will "freeze" for from one to ten seconds. During this time, the car of course continues to move. If these freezes occur near decision points, the driver may go past the intersection without being told what to do. At

20 miles per hour a car travels nearly 45 meters in five seconds. Figure C-4 shows a closeup histogram of the average arrival rate of odometer packets. The navigation system in the car sends a packet once every second. Most packets arrive within a second, but a few are delayed, some by up to ten seconds. (These delays may also arise from delays at the workstation. Lisp Machines are not noted for real-time response.)

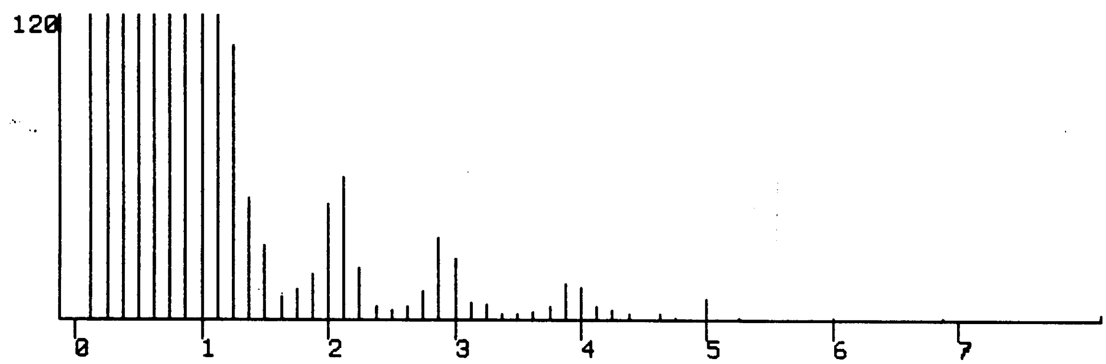


Figure C-4: Histogram of inter-arrival times of packets

It would be better to have a protocol which guarantees to deliver data intact and free of errors, if it delivers it at all, but does not guarantee to deliver the data. Real time data is only valuable in real time, and time spent retransmitting old data is taken away from never, more valuable data. Such a protocol modification is feasible for the Spectrum product (personal communication), and might be required for future work.

Index

Bibliography

[1] Federal Highway Administration. Economic assessment of potential solutions for improving motorist route following. Technical Report RD-86/029, Federal Highway Administration, June 1986.

[2] IEEE Aerospace and Electronic Systems Society. *IEEE Position and Location Symposium.* IEEE, 1986. IEEE reference 86CH2365-5.

[3] IEEE Aerospace and Electronic Systems Society. *IEEE Position and Location Symposium.* IEEE, 1988. IEEE reference 88CH26757.

[4] Ron Alexander. A Way to Go From A to B. *New York Times*, 1989. March 11, page 52.

[5] Douglas E. Appelt. Some pragmatic issues in the planning of definite and indefinite noun phrases. In *Proceedings of the 23rd conference of the Association for Computatonal Linguistics*, pages 198–203, 1985.

[6] G. A. Arrdondo, J. C. Feggeler, and J. I. Smith. Voice and data transmission. *Bell System Technical Journal*, 58(1):97–122, Jan 1979.

[7] Javad Ashjaee. Differential GPS with Ashtech XII. In *IEEE Position and Location Symposium*, pages 318–322, 1988. IEEE CH2675-7.

[8] Herman J. G. M. Benning. Digital maps on compact discs. Technical Paper Series 860125, Society of Automotive Engineers, 1986.

[9] Earl G. Blackwell. Overview of Differential GPS Methods. *Navigation: Journal of The Institute of Navigation*, 32(2):114–125, Summer 1985.

[10] Matthew M. Blizard. Update: Differential loran-c system. Technical report, USCG R&D Center, Groton, CT 06340, 1988.

[11] A. Borelli and S. Sklar. An integrated approach to Automatic Vehicle Monitoring and Mobile Digital Communications. In *Proceedings of the 28th IEEE Vehicular Technology Conference*, March 1978. IEEE publication 78CH12971.

[12] Mark Bosworth and Rodman Low. Taxi!: Dynamic cartographic software for training cab drivers. Technical report, Hunter College Department of Geology and Geography, (212)-772-4000, 1988. paper presented at the Annual Meeting of the Association of American Geographers.

[13] Gerard Boudriault. Topology in the TIGER file. In *AUTO CARTO 8 International Symposium on Automation in Cartography*, April 1987.

[14] Peter Braegas. Function, Equipment, and Field Testing of a Route Guidance and Information System for Drivers (ALI). *IEEE Transactions on Vehicular Technology*, VT-29(2):216–225, May 1980.

[15] Donald F. Cooke. Vehicle Navigation Appliances. In *AUTO CARTO 7 International Symposium on Automation in Cartography*, pages 108–115, March 1985.

[16] S. A. Dale and P. Daly. Recent Observations on the Soviet Union's Glonass Navigation Satellites. In *IEEE Position and Location Symposium*, pages 20–25, 1986.

[17] S. A. Dale, I D. Kitching, and P. Daly. Position Fixing Using the USSR's Glonass C/A Code. In *IEEE Position and Location Symposium*, pages 13–20, 1988.

[18] James R. Davis. A voice interface to a direction giving program. Technical Report 2, MIT Media Laboratory Speech Group, Apr 1988. Replaces the paper in the 1986 AVIOS proceedings titled "Giving Directions: A voice interface to an urban navigation program".

[19] James R. Davis and Thomas F. Trobaugh. Direction assistance. Technical Report 1, MIT Media Laboratory Speech Group, Dec 1987.

[20] Arthur R. Dennis. STARFIX. In *IEEE Position and Location Symposium*, pages 251–255, 1986.

[21] R. J. Elliot and M. E. Lesk. Let your fingers do the driving: Maps, yellow pages, and shortest path algorithms. Technical Report unpublished, Bell Laboratories, 1982.

[22] R. J. Elliot and M. E. Lesk. Route finding in street maps by computers and people. In *Proceedings of the National Conference on Artificial Intelligence*, pages 258–261, 1982.

[23] Amr A. El-Sawy et al. LORAN C Tracking of Land Vehicles using micro computers. In *Proceedings of the 28th IEEE Vehicular Technology Conference*, pages 157–161, March 1978. IEEE publication 78CH12971.

[24] Gregory M. Fontana. Transmission of Data over the Cellular Telephone Network. In *Proceedings of the 37th IEEE Vehicular Technology Conference*, pages 528–531, June 1987. IEEE publication 87CH24299.

[25] Seiichi Fujimura et al. A vibration gyros and their applications. In *Digest of Technical Papers*, pages 116–117. IEEE ICCE, 1989. IEEE publication CH2724-3.

[26] F. K. Ganjon. Liability aspects of elctronic charts. In *IEEE Position and Location Symposium*, pages 303–305, 1986.

[27] David K. Gifford, John M. Lucassen, and Stephen T. Berlin. The application of digital broadcast communication to large scale information systems. *IEEE Journal on Selected Areas in Communication*, 3(3):457–467, May 1985.

[28] H. P. Grice. Logic and conversation. In Cole and Morgan, editors, *Syntax and Semantics: Speech Acts*, volume 3, pages 41–58. Academic Press, 1975.

[29] George W. Gruver and Otto A. Reichardt. The Huntington Beach Automatic Vehicle Monitoring System using overlapping RF signposts. In *Proceedings of the 28th IEEE Vehicular Technology Conference*, pages 538–542, March 1978. IEEE publication 78CH12971.

[30] Peter Haeussermann. On board computer system for navigation, orientation, and route optimization. Technical Paper Series 840483, Society of Automotive Engineers, 1984.

[31] Robert G. Hajovsky. An Update on the Huntington Beach Automatic Vehicle Location System. In *Proceedings of the 31rst IEEE Vehicular Technology Conference*, April 1981. IEEE publication 81CH1638-6.

[32] Herbert H. Hall. Alternate approaches to automobile navigation. Technical Paper Series 861057, Society of Automotive Engineers, 1986.

[33] Clyde B. Harris, Laurie A. Klesh an Edward J. Krawkiwsky, Hassan A. Karimi, and Ness S. T. Lee. Digital Map Dependent Functions of Automatic Vehicle Location Systems. In *IEEE Position and Location Symposium*, pages 79–87, 1988. IEEE CH2675-7.

[34] Ted Harris and David Kerridge. The geomagnetic storm of 13 March 1989. *Radio Communication*, 65(5), May 1989.

[35] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on SSC*, 4:100–107, 1968.

[36] Stanley K. Honey, Marvin S. White Jr., and Walter B. Zavoli. Extending Low Cost Land Navigation Into Systems Information Distribution and Control. In *IEEE Position and Location Symposium*, pages 439–444, 1986. IEEE 86CH2365-5.

[37] Tom Hunter and Javad Ashjaee. Land Navigation and Fleet Management with GPS, LORAN, and dead reckoning sensors. In *IEEE Position and Location Symposium*, pages 54–60, 1988. IEEE CH2675-7.

[38] Toshiyuki Itoh, Yasuhiko Okada, Akira Endoh, and Kenji Suzuki. Navigation systems using gps for vehicles. Technical Paper Series 861360, Society of Automotive Engineers, 1986.

[39] Robert V. Janc. Consideration of the Various Error Sources in a Practical Automatic Vehicle Location system. In *Proceedings of the 34th IEEE Vehicular Technology Conference*, pages 277–284, 1984. IEEE publication number CH19513.

[40] Rudolph M. Kalafus, Janis Vilcans, and Norman Knable. Differential Operation of NAVSTAR GPS. *Navigation: Journal of The Institute of Navigation*, 30(3):187–204, Fall 1983.

[41] Gerald S. Kaplan and Andrew D. Ritzie. An X-Band System using Semi-passive Signpost Reflectors for Automatic Location and Tracking of Vehicles. *IEEE Transactions on Vehicular Technology*, VT-26(1):18–22, February 1977.

[42] Christine Kinnear. The TIGER Structure. In *AUTO CARTO 8 International Symposium on Automation in Cartography*, April 1987.

[43] M. D. Kotzin and A. P. van den Heuvel. Dead Reckoning Vehicle Location using a solid state rate gyro. In *Proceedings of the 31rst IEEE Vehicular Technology Conference*, pages 169–172, April 1981. IEEE publication 81CH1638-6.

[44] E. J. Krakiwsky, C. B. Harris, and R. V. C. Wong. A Kalman Filter for Integrating Dead Reckoning, Map Matching, and GPS Positioning. In *IEEE Position and Location Symposium*, pages 39–46, 1988. IEEE CH2675-7.

[45] L. R. Kruczynski et al. Global positioning system differential navigation tests at the yuma proving ground. *Navigation: Journal of The Institute of Navigation*, 32(2):126–138, Summer 1985.

[46] Benjamin J. Kuipers. *Representing Knowledge of Large-Scale Space.* PhD thesis, MIT, July 1977. Issued as Technical Report 418.

[47] Benjamin J. Kuipers. Modelling human knowledge of routes: Partial knowledge and individual variation. In *Proceedings of the National Conference on Artificial Intelligence*, pages 216–219, 1983.

[48] W. Thompson Lawrence. A Magnetic Signpost AVM System with Limited Dead Reckoning. *IEEE Transactions on Vehicular Technology*, VT-26(1):23–29, February 1977.

[49] M. Liberman and J. Pierrehumbert. Intonational invariance under changes in pitch range and length. In *Language Sound Structure*, chapter 10. MIT Press, 1981.

[50] Kevin Lynch. *The Image of the City.* MIT Press, 1960.

[51] Peeder Ma. An algorithm to generate verbal instructions for vehicle navigation using a geographic database. *The East Lakes Geographer*, 22:44–60, 1987.

[52] Robert W. Marx. The TIGER system: automating the Geographic Structure of the United States Census. *Government Publications Review*, 13:181–201, 1986.

[53] Robert W. Marx. GIS, TIGER, and Other Useful Acronyms. In *National Conference of Geographic Information Systems.* Canadian Institute for Surveying and Mapping, March 1989.

[54] Robert W. Marx and Alan J. Saalfeld. Programs for assuring map quality at the bureau of the census. In *Fourth Annual Research Conference*, Geography Division, Room 3203-4, Bureau of the Census, U. S. Department of Commerce, Washington DC 20233, March 1988.

[55] Clare D. McGillem et al. Experimentally determined accuracy and stability of loran c signals for land vehicle location. *IEEE Transactions on Vehicular Technology*, VT-31(1):15–21, Feb 1982.

[56] Matthew McGranaghan, David M. Mark, and Michael D. Gould. Automated provision of navigation assistance to drivers. *The American Cartographer*, 14(2):121–138, 1987.

[57] Douglas L. Milliken. personal communication, Dec 1988.

[58] Ernst Peter Neukirchner and Wolf Zechnall. Digital Map Data Bases for Autonomous Vehicle Navigation Systems. In *IEEE Position and Location Symposium*, pages 320–324, 1986. IEEE 86CH2365-5.

[59] Osamu Ono, Hidemi Ooe, and Masahiro Sakamoto. CD-ROM Assisted Navigation System. In *Digest of Technical Papers*, pages 118–119. IEEE ICCE, 1988.

[60] Harry M. O'Sullivan. Cellular telephone data communciation system and method. United States Patent 4,697,281, September 1987.

[61] Lee Ott. STARFIX: Commercial satellite positioning. In *IEEE Position and Location Symposium*, pages 8–12, 1988.

[62] Georges Pachiaudi, Monique Vernet, and Anne Foti. Noise and speech interference in light automotive vehicles. Technical Paper Series 885118, Society of Automotive Engineers, 1988.

[63] Cécile L. Paris. Tailoring object descriptions to a user's level of expertise. *Computational Linguistics*, 14(3):64–78, September 1988.

[64] Timothy J. Peters. Autombile navigation using a magnetic flux gate compass. *IEEE Transactions on Vehicular Technology*, VT-35(2):41–47, May 1986.

[65] Otmar Pilsak. Eva: An electronic traffic pilot for motorists. Technical Paper Series 860346, Society of Automotive Engineers, 1986.

[66] Alexander H. Riccio. Automatic Vehicle Location Dallas Test Results. In *Proceedings of the 27th IEEE Vehicular Technology Conference*, pages 144–152, March 1977. IEEE publication 77CH11767.

[67] Robert T. Richards and Leslie O. Snively. GEOSTAR positioning analysis. In *IEEE Position and Location Symposium*, pages 13–19, 1986.

[68] Christopher K. Riesbeck. "You Can't Miss It": Judging the Clarity of Directions. *Cognitive Science*, 4:285–303, 1980.

[69] Mortimer Rogoff. Electronic Charts - At the Heart of 21rst Century Navigation. In *IEEE Position and Location Symposium*, pages 88–94, 1988. IEEE CH2675-7.

[70] Dan A. Rosen, Frank J. Mammano, and Rinaldo Favout. An electronic route-guidance system for highway vehicles. *IEEE Transactions on Vehicular Technology*, VT-19(1):143–152, Feb 1970.

[71] Ronald Rosenberg. Mapping out a new idea. *The Boston Globe*, 1987. February 17, page 39.

[72] Harvey Sacks, Emanuel A. Schegloff, and Gail Jefferson. A simplest systematics for the organization of turn-taking for conversation. *Language*, 50(4):696–735, 1974. Reprinted in*Studies in the Organization of Conversational Interaction*, J. Schenken, ed., Academic Press 1978.

[73] Richard C. Sagers. A LORAN-C based receiver for automatic vehicle location. In *Proceedings of the 35th IEEE Vehicular Technology Conference*, pages 387–391, 1986. IEEE CH2308-5.

[74] Stephen R. Sampson. A survey of commercially available positioning systems. *Navigation: Journal of The Institute of Navigation*, 32(2):139–148, Summer 1985.

[75] C. Schmandt and B. Arons. A conversational telephone messaging system. *IEEE Trans. on Consumer Electr.*, CE-30(3):xxi–xxiv, 1984.

[76] Chris Schmandt. Employing voice back channels to facilitate audio document retrieval. In *Proceedings*, pages 213–218. ACM Conference on Office Information Systems, 1988.

[77] Christopher Schmandt. Speech synthesis gives voiced access to an electronic mail system. *Speech Technology*, 2(3):66–69, 1984.

[78] A. J. Sedlock. Mid-continent LORAN-C expansion. In *IEEE Position and Location Symposium*, pages 520–524, 1986.

[79] Joel Sobel. Principal components of the census bureau's tiger file. Research in Contemporary and Applied Geography Discussion Series 3, Department of Geography, SUNY Binghamton, 1986.

[80] G. J. Sonnenberg. *Radar and Electronic Navigation.* Butterworths, sixth edition, 1988.

[81] Lynn A. Streeter and Diane Vitello. A profile of drivers' map reading abilities. *Human Factors*, 28:223–239, 1986.

[82] Lynn A. Streeter, Diane Vitello, and Susan A. Wonsiewicz. How to tell people where to go: comparing navigational aids. *International Journal of Man/Machine Systems*, 22(5):549–562, May 1985.

[83] Katsutoshi Tagami et al. Electro gyro-cator: New intertial navigation system for use in automobiles. Technical Paper Series 830659, Society of Automotive Engineers, 1983.

[84] C. Tarriere et al. Some ergonomic features of the driver vehicle environment interface. Technical Paper Series 885051, Society of Automotive Engineers, 1988.

[85] Martin L. G. Thoone and Ron M. A. M. Breukers. Applications of the compact disc in car information and navigation systems. Technical Paper Series 840156, Society of Automotive Engineers, 1984.

[86] T. Tsumura, M. Hashimoto, and N. Fujiwara. A Vehicle Position and Heading Measurement System Using Corner Cube and Laser Beam. In *IEEE Position and Location Symposium*, pages 47–53, 1988.

[87] Lawrence A. Whitcomb. Using Low Cost Magnetic Sensors on Magnetically Hostile Land Vehicles. In *IEEE Position and Location Symposium*, pages 54–60, 1988. IEEE CH2675-7.

[88] Walter W. Wierwille et al. Strategic use of visual resources by the driver while navigating with an in-car navigation display system. Technical Paper Series 885180, Society of Automotive Engineers, 1988.

[89] Walter B. Zavoli and Stanley K. Honey. Map Matching Augmented Dead Reckoning. In *Proceedings of the 35th IEEE Vehicular Technology Conference*, pages 359–362, 1986. IEEE CH2308-5.