

# Direction Assistance

James Raymond Davis  
and  
Thomas Frank Trobaugh

December 1987

Speech Research Group Technical Memo 1  
The Media Laboratory  
Massachusetts Institute of Technology

## Abstract

Direction Assistance is an interactive program that provides spoken directions for automobile travel within the Boston area. The program has a telephone interface which uses touch tone keypad input and synthetic speech output. Routes are both short and easily followed. The directions are given in fluent English. The program has successfully directed newcomers through Boston.

This paper tells how we built Direction Assistance, with emphasis on how the available databases are and are not useful for this application. It also talks about automatic generation of route descriptions, and compares our work to that of others.

## 1 Introduction

### 1.1 Overview

Direction Assistance consists of about 11,000 lines of CommonLisp code, runs on a Symbolics Lisp Machine, and uses a Digital Equipment Corporation DecTalk synthesizer. It was written mostly during the summer of 1985 at the Thinking Machines Corporation of Cambridge, Mass. Since then, it has undergone periodic rewrites. It is running at the Media Lab, and is also installed at the Computer Museum in Boston and as part of the Age of Intelligent Machines exhibit traveling across the United States.

Direction Assistance consists of five modules. The **Location Finder** queries the user to obtain the origin and destination of the route. A location may be specified as a street address or as a telephone number. The **Route Finder** finds a simple, short route between the two points. The **Describer** generates high quality English text describing the route. The **Narrator** recites the route to the user. In addition, there is a graphical interface for maintenance.

These modules share a set of databases. The most important is the street map, which covers an eleven square mile area of Boston centered on the Charles River. A second database is an inverted phone directory, which provides a street address for a phone number.

In this paper, we discuss the databases, the Route Finder, and the Describer. The Location Finder and Narrator are described in [2].

It would be inappropriate to continue without mentioning the pioneering work of Jane Elliot and Mike Lesk[5,4]. Our work differs from theirs in several ways. Our interface uses synthetic speech and pushbutton telephones rather than a graphics terminal. We are much more concerned with generating fluent English text than they. On the other hand, we are not much concerned with route finding algorithms. Finally, Elliot and Lesk used a Yellow Pages database in addition to the white pages and street map. We will not clutter this paper with citations to Elliot and Lesk on every point where they have made contributions. They are to be assumed.



The New Yorker, April 6 1987 p 56

We next discuss the underlying databases, and then the modules which use them. The description of the databases will by necessity refer to features of the program in order to motivate the construction of the database.

## 2 Databases

### 2.1 Streets

Our street map began as a DIME (Dual Independent Map Encoding) file distributed by the United States Geological Survey[1]. A DIME file consists of a set of straight line segments, each with a name, a type, endpoints in longitude and latitude, and some additional information. Segment types include natural features (chiefly water boundaries), railroads, town and property lines as well as streets. The latter are also labeled with address numbers on both sides of the street at each endpoint; thus it is possible to estimate the coordinates for any street address by interpolation, assuming all lot sizes to be constant.

We began with an 11 square mile subset centered roughly on the Charles River. This includes portions of Boston (Charlestown, Allston, Back Bay, South End, North End), Brookline, and Cambridge (Cambridgeport and Harvard, Inman, Central and Kendall Squares). (See figure 1.) There are about 279 miles of streets in the map, which contains 6163 segments, of which 5506 correspond to streets. The total size is about 477 kilobytes.

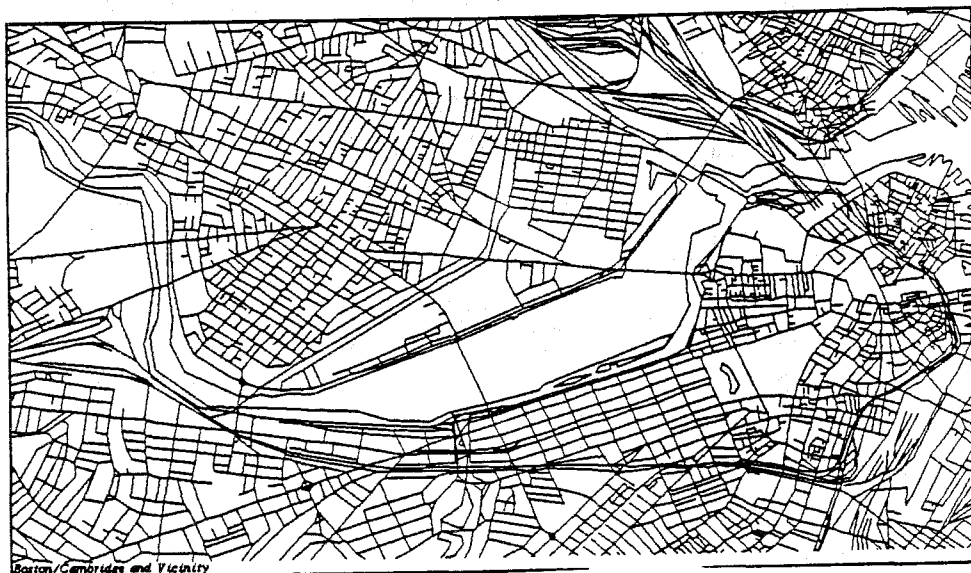


Figure 1: Street Database

The DIME file as supplied was far from suitable for our use. It contained many errors: streets were missing, mislabeled, or misconnected, and names were not spelled consistently. In some cases, more than one segment occupied the same place, and some segments were connected to themselves. We wrote a

battery of plausibility checkers to detect and remove these errors, automatically where possible.

In addition to correcting errors, we had to add new kinds of information to the database. The most important information was whether a street was one way. We also classified streets by quality, and recorded textual descriptions for some turns. We'll now describe each of these.

Segments in the DIME file are deemed to connect if they share a common endpoint. We refer to this kind of connection as *physical connectivity*. Every segment has two endpoints, and for each of these there is a list of the segments which are physically connected to that endpoint. Obviously, physical connectivity is a symmetric non-reflexive relation. Physical connectivity is not sufficient for route finding, since it may not be legal to drive from one piece of pavement to another, even though they meet, because one might be one-way, or a turn might be forbidden, or there might be a divider in the way<sup>1</sup>. To provide for the fact that one can not always drive from a segment to any other physically connected to it, we added a second kind of connection, *legal connectivity*. Two (street) segments are legally connected if one may drive from one segment to the other without breaking a law. Legal connectivity supplements, but does not replace, physical connectivity. Physically connected segments include those that can be seen in passing, and must also be retained, for they are important in forming descriptions. One cannot turn onto a railway, though the street and railroad segments are physically connected, but one may also wish to mention the crossing of the railroad as a salient detail of the tour.

Not all streets are created equal. We wanted our routes to use the widest, fastest, and most easily located streets, so we gave each street a value for *goodness* (super, good, average, or bad). By definition, most streets are average. The super streets are the expressways, interstate highways, and other limited access roads. Our rating of super is awarded more on the basis of being easy to find and to follow, since super roads are often crowded and slow. At the other extreme, the bad streets are those we know to be narrow or in poor repair. Our database contains only three miles of such streets. Unlike the taxi driver, we are not interested in shortcuts which use marginal streets.

The concept of "better than average" is a bit hard to define. We wanted to identify streets which were likely to be easy to find and follow. We decided that streets that were long were likely to be important, so we marked all streets longer than one half mile as "good", and then added a few more by hand if they seemed important. The resulting network is about 105 miles long, and forms a simplified skeleton covering our map. It appears in figure 2.

The third extension was to expand the street classification scheme. We added new segment types for bridges, underpasses, rotaries, and access ramps. This information is useful to both the route finder and to the describer, as we show below.

Finally, at every intersection in the map we can store additional descriptive information about each possible turn at the intersection, in the form of labelled items. Each item has a label telling what kind of information is stored, for instance an exit number or the text of a sign present at that intersection. This information is used by the Describer.

We made almost all of these corrections and augmentations ourselves from observations in the field.

---

<sup>1</sup>In this case, the turn is forbidden by physical obstacles, and not merely law or custom. But rather than engage in an epistemology of barriers, we use the same mechanism to represent this restriction.

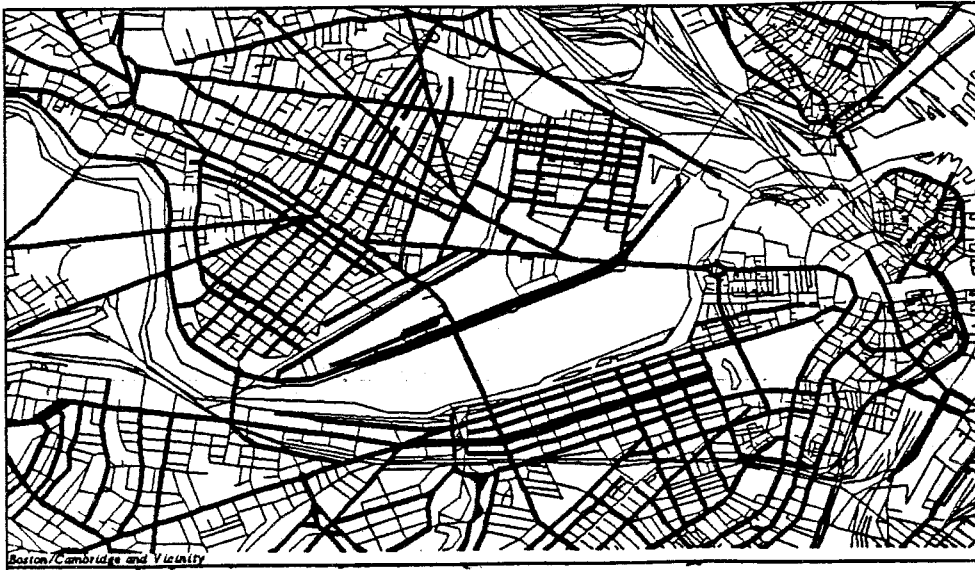


Figure 2: Network of good streets

We could not find a paper map listing all the one way and restricted turning streets of Boston, so we had to drive around looking for them. This investment in time and effort is a major cost of the system, but needs to be done only once. The graphic database editor was extremely useful, as it permitted rapid editing of the database. We commend the many designers of the Lisp Machine window system for making this easy.

## 2.2 Neighborhoods

A related database lists the neighborhoods of Boston, with their associated zip codes. We need this database because a given street might occur in several different towns. For instance, there are three distinct streets named "Washington" in our map, in Boston, Cambridge, and Somerville. Even worse, Cambridge contains two different streets named "Elm".

The Location Finder uses this database to disambiguate street names. When the user supplies a name that could designate more than one street, it is necessary to ask for further information, e.g. "Do you mean Beacon Street in Cambridge or in Boston?". To make this as easy as possible, it is best to use the names of the most general locations that still distinguish the streets<sup>2</sup>. If the street occurs in two neighborhoods of the same city, the neighborhood name is used. If the street occurs in different cities, the city name is sufficient. We determine neighborhood from the Zip code of the street. The mapping from Zip to neighborhood is imperfect, but good enough for our purposes. For the most part, the neighborhood names are those used by the local post offices. We think it is very likely that these names are also familiar to the local residents, and intelligible to visitors, but we have no evidence.

<sup>2</sup>This assumption could be tested. If people represent locales hierarchically, and if there is a preferred level of representation, it might be more difficult to determine inclusion in a too-general region.

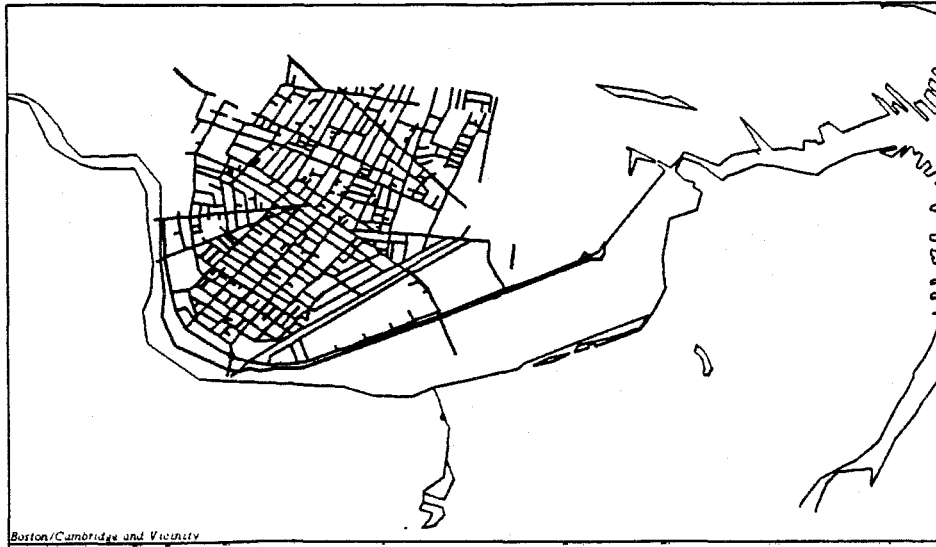


Figure 3: Central Square, 02139

### 2.3 Inverted Phonebook

The inverted telephone directory allows us to map telephone numbers to street addresses. We built this database ourselves, by inverting a "white pages" database. This required parsing the street addresses in the white pages, which was difficult for several reasons. The white pages have a great variety of spelling and abbreviation. We found, for instance, 23 variations of "Massachusetts". In addition, the format is not consistent. Sometimes listings contain professions ( "atty" or "archt"), or a second phone number ("If No Answer"), or other information (e.g. "toll free", "children's phone"). We did not have the typographic information that helps separate names from locations and phone numbers. Finally, addresses are often incomplete, listing only a city, or road, or some a name which does not correspond to a street, such as a shopping center or an office park.

Even after parsing, it can be hard to determine locations from a a phone book listing. Even the best entries have at best a street, number, and city. But as we said above, streets occur in more than one place within a city. There is a rough correspondence between exchange and locale, so we can sometimes determine a unique location with this extra information. But when we can not, the Location Finder must ask the user to choose a location, as it does for street names.

Having described the databases, we now turn to the modules of Direction Assistance.

### 3 Route Finder

The Route Finder finds a route subject to three constraints. The route must be easy to follow, reasonably short, and it must be found before the user loses patience<sup>3</sup>. These constraints conflict. Rarely is there a straight line route - the shortest route may require devious shortcuts. We are biased towards simplicity, since we want our users not to get lost.

The output of the Route Finder is a *path*, an ordered list of street segments, such that the origin is on the first segment, the destination on the last, and each segment is legally connected to the next. The real time requirements of the system rule out exhaustive, breadth first search<sup>4</sup>. The current implementation uses a best first search that provides reasonably good routes in a moderate time. A sample route appears in figure 4.

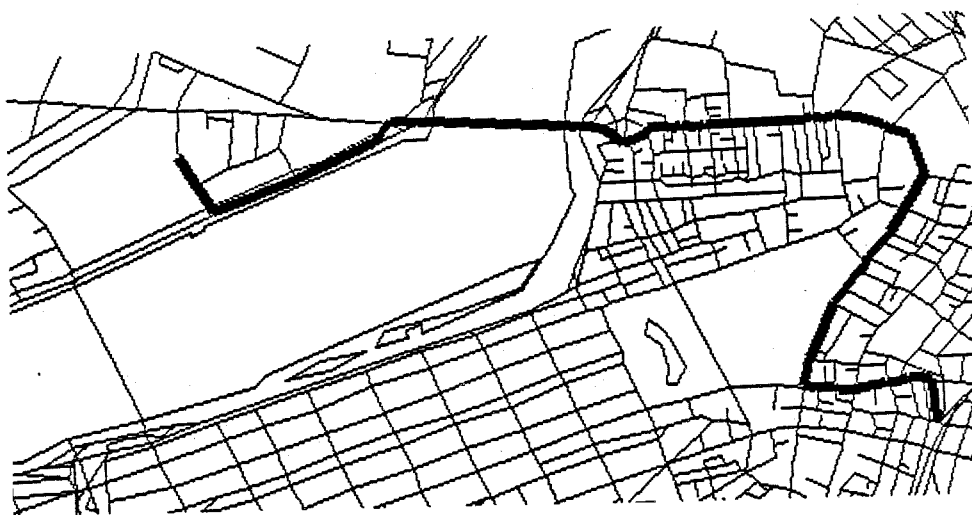


Figure 4: A sample route

Best first search is an improvement on breadth first search. Search is conducted in (simulated) parallel on a list of candidate partial paths. For each path, there is a cost which is the sum of the known cost for the current path and an estimate which is a lower bound on the cost for (as yet undetermined) remainder of the path. At each step of the search, we consider the path of least cost, and expand it by considering all segments legally connected to its terminal end. The estimation function is just the Cartesian distance, since no route can be shorter than a straight line. Figure 5 shows every segment visited by the search in finding the route shown above.

As Elliot and Lesk point out, it is not desirable to find minimum distance routes, for these have too

---

<sup>3</sup>A fourth constraint which we do not consider explicitly is that the route must be easy to describe. We are familiar with situations where a person asks for a route to a familiar place, but we can not describe the route because it is a "felt path": we no longer remember (or do not know) the names of the streets, only a list of subtle cues we can't describe.

<sup>4</sup>on a serial machine, anyway. An experimental version on the Connection Machine[6] works in just this way.





Figure 5: All segments touched by search

many turns. Such routes are hard to describe and hard to follow. Elliot and Lesk impose a cost of  $1/8$  mile for a right turn, and  $1/4$  mile for a left turn. We extend their system of costs in several ways. First, we consider street goodness. Travel down a "super" street is not as "expensive" as travel down an average street, and travel down a "bad" street incurs a surcharge. Second, we consider sharp right turns to be as bad as left turns, since they are harder to execute. Third, we reduce or waive turn costs in some cases. For example the turning cost is halved for a turn on to or off a one-way street, and waived altogether for a forced turn ("left turn only"). A turn onto a bridge is also free, since bridges are major landmarks, and contribute to ease of following the route. We have not studied the effect of these routes on the routes found, nor have we attempted to determine whether the routes are better where different. Such a study would require a model of driver's errors, both of understanding and of execution.

## 4 Describer

The Describer generates a set of text instructions for following the route. (An example of its output appears in figure 6.) We generate text instead of a map for two reasons. First, the system is used by telephone, which limits the output to voice. But even if our users had portable graphics terminals with modems, we would prefer text to graphics, because some people can not read maps. In a survey of map reading abilities Streeter and Vitello recommend text as a "lowest common denominator" [9].

The Describer creates a new representation of the route, instead of using the path itself. There are two reasons for this second form of representation. First, the elements of a path (segments) are too fine grained for useful textual description. Recall that a segment reaches from just one intersection to the next. This is smaller than our sense of a "street", which continues as a unity past many intersections. In addition, segments are straight lines: so a street with no intersections might be still represented as a

If your car is on the same side of the street as 20 Ames Street, turn around, and start driving. Drive all the way to the end, about one eighth of a mile. Make a left onto Memorial Drive. Drive about one eighth of a mile. After you pass Wadsworth Street on the left, take the next left. It's an easy left. Merge with Main Street. Stay on Main Street for about ninety yards, and cross the Longfellow Bridge. You'll come to a rotary. Go about half way around it, and turn onto Cambridge Street. Drive all the way to the end, about three quarters of a mile. Make a right onto Tremont Street. Drive about one half of a mile. After you pass Avery Street on the left, take the next left onto Boylston Street. Stay on Boylston Street for about one eighth of a mile. After you cross Washington Street, it becomes Essex Street. Keep going. Drive about one eighth of a mile. After you pass Ping On Street on the right, take the next right onto Edinboro Street. Number 33 is about one eighth of a mile down on your right side.

Figure 6: sample of directions

sequence of segments if it made a broad turn. We want to describe the entire stretch of a street as a single object. A second reason is that a path is just a topological structure, but natural instructions should be expressed in terms of geometry and of types of streets. Consider the difference between a "fork", a "T", and an "exit", as shown in figure 7. All have the same topology - a branch in the road. But they must be described differently. The Describer's structure is a *tour*, which is a sequence of *acts* to be taken in following the path.

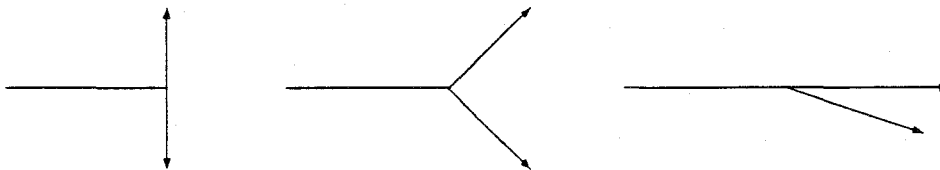


Figure 7: T, fork, and exit all have same topology

#### 4.1 Acts

Acts are things a driver does (or notices) while following a route. Figure 8 shows our taxonomy of acts.

Each of these acts must be *recognized*. The route finder works only with segments, and the Describer builds acts which describe motion from segment to segment. We now describe each of these acts, and how we recognize them. We describe the text generated for each below.

- Boundaries
  - Start
  - Stop
- Straight
  - Name Change
- Turn
  - Enter
  - Exit
  - Merge
  - Fork
  - U Turn
  - Rotary
  - Ordinary

Figure 8: Act Taxonomy

The first act is necessarily START, and the last STOP. They are trivial to recognize. The NAME CHANGE act requires the driver to notice a change in name, but nothing further. We include it only to avoid confusion. The difference between a NAME CHANGE and a TURN is that the former consists of a two streets meet within 10 degrees of straight, and where there is no other segment at the intersection with the same name as either of them. These two criterion are almost correct, but not quite right. There are streets which seem (to us) to be name changes, but have more extreme turns (at least, as represented in the map). For the present, we have caused these to be treated as name changes by changing the map, slightly altering the positions to make the turns more gentle. This would be intolerable were we using the map for, say, surveying, but is of no consequence for route description.

There are several types of TURN acts. The ENTER and EXIT acts refer to limited access roads. In this case, some of the travel will often be on "nameless" segments - access ramps. This shows one reason for the additional classification of street segments. We want to recognize entrances and exits, and we want to describe access ramps in different terms than other streets.

A MERGE and a FORK are similar in that they are different actions that might be taken at the same intersection, depending upon the direction one is driving. A Merge has the following characteristics:

1. Old and new streets have different names.
2. Only one street is legally possible.

3. The angle of turning is small.
4. There is at least one other street going to the destination street.
5. All streets make only small turns onto the destination.

At a FORK on the other hand, there are at least two ways to go, though all are shallow turns. Note that a "fork" onto an exit ramp is recognized as an EXIT.

There are two types of U turn known to drivers in Boston. The first kind is made in the middle of the street (within a single segment, in our representation). Our routes never include such turns. Not only are they illegal, such moves never shorten the path. The second kind of U turn is the sort one makes to reverse direction on a divided road. Typically one makes a left onto a nameless piece of road, which is often very short, and then makes a second left. This double turn is what we call a U TURN act. It is very important to recognize this act, because describing it as two successive lefts is very confusing. It is a single entity in the minds of drivers. We recognize a U TURN as a pair of turns where the intermediate segment is less than 165 yards long, the total angle is within 20 degrees of 180, and the name of the street is unchanged after the two turns.

Perhaps the most insidious feature of Boston's streets is the ROTARY. For those not familiar with the term, a rotary is a one way street in a circle. Traffic enters the rotary on roads which are (usually) tangent to the circle, moves counterclockwise around the circumference, and exits on another tangent. Rotaries are difficult to traverse because they cars enter and exit within a very short distance, without much room to maneuver. Recognition of a rotary is trivial, but only because we label all rotary segments explicitly in the street map.

An ORDINARY turn is anything not handled by one of the above cases.

## 4.2 Cues

While the Describer is collecting the acts of the tour, it also collects cues. A cue helps the driver follow the tour. We distinguish four kinds of cues. *Action* cues tell when to do an act. *Confirmatory* cues describe things that will be seen while following the route. *Warning* cues caution the driver about possible mistakes. A warning successfully heeded also serves as a confirmatory cue. *Failure* cues describe the consequences of missing an act, e.g. "If you see this, you have gone too far".

The most common action cue is just the name of the street. An instruction such as "Turn right onto Tremont Street." tells the driver what to do and when to do it. This cue may be hard to follow, since street signs may be missing. A very strong action cue is coming to the end of a road. No one is likely to forget to turn under this circumstance, since the alternative is to leave the road. We refer to this as a "forced turn" cue.

Distance traveled is also a cue, but hard to use. People have a vague sense of distance, but not an accurate one. Still, we use distance as a secondary cue, because we can compute it easily and it helps some people. We express distance in yards when less than 1/16 of a mile, and other distances in approximate

fractions of a mile because people are accustomed to seeing distances expressed this way. We do not use tenths of miles, because some people do not know how to use odometers, and because using an odometer to calculate distance requires doing mental arithmetic, which might prove distracting while driving.

We never use blocks, since a block is not a clearly defined concept. We do not know whether a block is bounded by an intersecting street, or only by streets that cross and continue. Figure 9 illustrates this. In

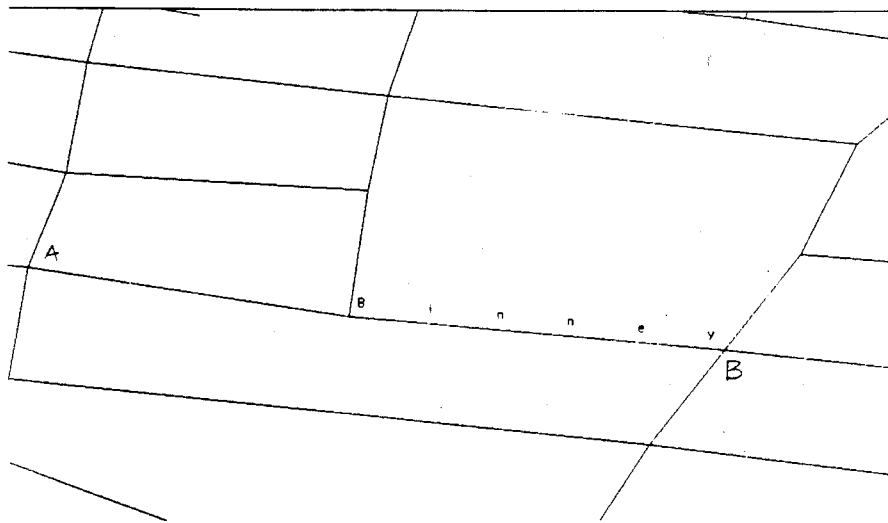


Figure 9: Is the distance between "A" and "B" one block, or two?

any event, we do not expect our drivers to be able to drive more than two or three blocks without losing count. Since we don't want to rely on distance or counting blocks, we use as a cue for an act the name of a street immediately preceding the act. This is a risky cue, since the driver who misses the cue may keep looking for it and miss the destination street as well. To make this less likely, we use only streets on the same side as the turn for a cue. This way, a driver need attend to only one side of the road while looking for street signs, so if the cue street is missed, the target street may still be seen. This same strategy is adopted in [10].

The confirmatory cues are crossing major streets or railroads, or going through an underpass. The only warning cue currently is a warning about left exits from limited access roads. We assume drivers will not take the wrong exit, but if they keep in the left lane they may be surprised by an unexpected left exit. We have not implemented failure cues.

### 4.3 Generating Text

For each act there is a corresponding routine which generates one to three sentences describing it. The routine selects appropriate cues from all those gathered. Now we'll describe some aspects of generating text.

```

(defun disc-seg-rotary (act)
  (list
    (make-sentence
      "You'll" "come" "to"
      (make-np-constituent '("rotary") :article :indefinite))
    (make-conjunction-sentence
      (make-sentence
        "Go" (rotary-angle-amount (get-info act 'rotary-angle))
        "way" "around" (make-anaphora nil "it"))
      (make-sentence
        "turn" "onto" (make-street-constituent (move-to-segment act) act))))))

(defun rotary-angle (angle)
  (selector angle <=
    (45 '("just" "a" "little"))
    (135 '("about" "a" "quarter"))
    (225 '("about" "half"))
    (315 '("about" "three" "quarters" "of" "the"))
    (360 '("almost" "all" "the"))))

```

Figure 10: Generator for rotary

Generating text for a START is tricky because it is hard to specify an initial direction. We do not use absolute directions, because most people do not know them. If we had a landmark database we might sometimes use relative direction (e.g. "towards the river"). Instead, we use the initial address, since that also determines a side of the street, and thus a direction to drive. We might have used "If your car is on the same side of the street as ... start driving the way it is facing.", but that sounded clumsy. Instead, we chose to give a negative instruction, either "If your car is on the same/opposite side of the street from ... turn around, and start driving." For one way streets we mention that the street is one way, and say "just start driving." We think (but do not know) that drivers would not have confidence in the instruction ("just start driving.") if it did not indicate that the system knew about the one way street.

One of the simplest generators is for rotaries. It appears in figure 10. Rotaries are hard to describe and hard to follow, because there are no good references for distance around a rotary. We can not expect people to measure angular distance around the rotary, and there may not be signs. The segments of a rotaries may or may not be nameless, or there may be several names involved. The rotary itself may have a name, e.g. Leverett Circle, but this name does not appear in the database and usually does not appear on any street signs either.

Output from this generator appears in figure 6. The generator produces two sentences, the second of which is a conjunction of two sentences. The distance around the rotary is converted from an absolute angle, as measured on the map, to an approximation in English.

The instructions generated have syntactic structure only for sake of exploiting generality in text generation. Thus the function `make-np-constituent` handles agreement between the article and the noun. The function `make-sentence` ensures that capitalization and punctuation are correct. Text is sent directly to the synthesizer, and punctuation is required to achieve proper intonation. The function `make-anaphora` serves no purpose at present, but in planned future research will allow us to convey intonational features of discourse[3].

#### 4.4 Comparison

We can compare our descriptions with those generated by Streeter and colleagues[10].

Streeter's descriptions are intended to be understood and acted upon in real time, as if uttered by a navigator in the next seat. (In fact, they are recorded on a tape, and the driver pushes a button to play the next instruction.) This interface imposes a new requirement on the form of the directions. Since they are to be heard and acted on in real time, it is important to repeat essential information so that it can be remembered. In our interface, we assume people are writing down the directions before they begin to drive, so repetition is not crucial. (The user can ask the Narrator to replay an instruction if it is not understood.)

They classify turns into ordinary turns, "T" turns, complex intersections, turns in short succession, and continues. Their "T" turn is our "forced turn" cue. The difference between an ordinary turn and a "T" turn is that the latter needs no failure cue. So our treatments are similar. We do not distinguish complex intersections, though we should. The Route Finder should avoid them, and the Describer should warn about them.

Their instructions are sometimes more structured than ours. They cluster turns which occur close to each other into a single instruction block, and their "continue" is just our "name change", but is also incorporated into the following turn. We recognize the importance of providing higher levels of structure, and wish to remind the reader that Streeter and company were working by hand, not with a program, and were in a better position to form hierarchies than we.

We claim that our directions are more natural than those of Elliot and Lesk, but have no proof for this. We leave it to the reader to judge.

Are our directions clear? We know that people have been able to follow them, but we have not made any systematic test. Christopher Riesbeck wrote a program (MCMAP) which judged the clarity of directions. Our directions would not be acceptable to it, to judge from its published description. Partly this is because we talk about features the program does not know, for example rotaries, but also because his program explicitly rejects use of miles for distance as inherently unclear. We use mileage only as an approximation, as a cue for when to look for a landmark, but the weak syntactic powers of MCMAP would not notice this. Also, we use "drive all the way to end", which Riesbeck terms a "procedural operator", and did not implement. Since people accept our directions, this suggests that Riesbeck's rules are too strict, or perhaps not powerful enough.

## 5 Discussion

Products like Direction Assistance are beginning to appear in the marketplace. It is reported that ETAK, of Sunnyvale California, has a product (the Navigator) which, installed in a car, estimates the car's position by counting wheel rotation and turning angle and comparing results with a stored map. A display in the dashboard displays the local area and the position of the car. The Navigator does not supply driving directions, but surely could be made to do so.

A more similar product is DriverGuide, made by Karlin and Collins, also of Sunnyvale, which is reported to produce printed directions for travel in the Bay Area[8].

### 5.1 Better databases are required

Any serious use of Direction Assistance requires further improvements to the street map. The area covered is too small, and even the small region covered is not fully mapped. More significantly, there are additional facts that the current street database format can not represent.

Among these are time-dependent legal restrictions (e.g. "no left turn during rush hour"), restriction of height, weight, and prohibition of commercial vehicles, multiple names of streets, presence of stop lights, and landmarks. In addition, the representation of addresses is not sufficient. We have seen addresses with fractions and with letters, and there are also streets where both even and odd numbers are on the same side of the street.

A practical system must account for multiple names. When Route 93 passes through Boston, it is also Route 3, the Fitzgerald Expressway, and the Southeast Expressway. When Massachusetts Avenue turns north at Harvard Square, only the southbound lane is "really" Massachusetts Avenue. The other direction is officially Peabody Street. We do not know which name to use when naming these streets, but we should at least be able to accept all synonyms on input.

Boston, like any city, changes its configuration of streets daily. Some changes, e.g. for construction, are temporary, although they may persist for years. Others are permanent. Streets are built and removed, and sometimes they change names or directions. A practical system requires accurate and timely corrections to the database.

We could give better directions with a better database, giving, for example the location of traffic lights or landmarks such as gas stations. Elliot and Lesk were able to capture business locations from an online Yellow Pages. To be more ambitious, we might hope for a representation rich enough to capture the qualities of image and orientation described by Kevin Lynch[7]. We have no proposal for how to do this at present.



## 5.2 Applications

We initially designed Direction Assistance with tourists in mind. Boston's confusing streets often lead the visitor astray. A tourist's direction guide could be provided by the city, or as a profitable venture. But a tourist may not know the street address or phone number for the destination. In fact, there may not be one, for the destination might be a general area, such as a neighborhood or park. Tourists would probably prefer to identify locations by name. It might be difficult to add this feature without making the interface more complicated.

Direction Assistance could direct people to services. Given the caller's location and the type of service desired, Direction Assistance could select the closest, and provide a route. This service might be dedicated to a single vendor (e.g. for banking machines) or as an advertising service for many customers.

Routing delivery vehicles pose special problems. Some of the most useful routes in Boston are closed to commercial vehicles, either for legal reasons or because they have such low underpasses that even scofflaws can not get through. We could extend the street database to record such facts.

We also feel compelled to mention the implications of Direction Assistance for privacy. Should a public Direction Assistance include home telephone numbers? People may want to keep the ability to give out their home phone numbers without also revealing their addresses to callers. One can hang up on an annoying caller. A visitor may be harder to dispose of.

## Acknowledgments

A prototype version was written by Dinarte R. Morais during the winter of 1985. We are indebted to him for decoding the DIME files, the initial window system interface, and the proof of concept. We made extensive use of a database package and string matcher written by Craig Stanfill. Charles Lieserson made major improvements to the search algorithm of the Route Finder. Fletch McCellan of the PhoneBook Corporation loaned us the raw phone book database. This work would not have happened without the guidance and persistence of Brewster Kahle. This paper was much clarified by the comments of Janet Cahn, Mike Hawley, Margaret Minsky, and Chris Schmandt. We thank them all.

This work was supported at MIT by the DARPA Space and Naval Warfare Systems Command, under contract numbers N00039-89-C-0406 and N00039-86-PRDX002 and by the Nippon Telegraph and Telephone Public Corporation. Hardware support was provided by Symbolics and Digital Equipment Corporation.

This paper bears the names of two authors, for the program was joint work. But though it is written in the plural, it is the work of only one of us. I dedicate it to Tom, who did not survive to see his work described. Though too small a memorial, it is the best I can manage at this time.

## References

- [1] *Geographic Base File GBDF/DIME: 1980 Technical Documentation*. U.S. Department of Commerce, Data Users Services Division, 1980.
- [2] James R. Davis. Giving directions: a voice interface to an urban navigation program. In *Proceedings of 1986 Conference*, pages 77-84, American Voice I/O Society, Sept 1986.
- [3] James R. Davis and Julia Hirschberg. Automatic generation of prosodic support for discourse structure. In *Proceedings of the Association for Computational Linguistics*, page (submitted), 1988.
- [4] 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.
- [5] 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.
- [6] W. Daniel Hillis. *The Connection Machine*. MIT Press, 1985.
- [7] Kevin Lynch. *The Image of the City*. MIT Press, 1960.
- [8] Ronald Rosenberg. Mapping out a new idea. *The Boston Globe*, 39, 1987. February 17.
- [9] Lynn A. Streeter and Diane Vitello. A profile of drivers' map reading abilities. *Human Factors*, 28:223-239, 1986.
- [10] 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.