DVILASER/PS EXTENSIONS TO LaTeX

Doug Maus
Bruce Baker
Textset, Incorporated

Overview

The LaTeX Document Preparation System is a collection of TeX macros designed to make it easy to produce high-quality documents. LaTeX was developed by Leslie Lamport — also the author of *LaTeX: A Document Preparation System* which is published by Addison-Wesley Publishing Company. LaTeX is in the public domain and is included in the standard TeX distribution available from Stanford University. It is also included in most proprietary implementations of TeX.

LaTeX is based on the concept of a document as a set of structures (e.g. descriptions, tables, enumerated lists) called *environments*. Environments start with a `\begin{`*environment*`}` statement such as `\begin{tabular}` or `\begin{verbatim}` and end with an analogous `\end{`*environment*`}` statement.

This article describes an upward compatible set of extensions to LaTeX's `picture` environment implemented by Textset to take advantage of the graphics capabilities of PostScript language used by the Apple LaserWriter and other printers. Textset's TeX output driver for the PostScript printers, DVILASER/PS, includes support which makes it easy to integrate user-supplied PostScript statements with the PostScript statements being generated automatically by DVILASER/PS as it converts TeX DVI files into PostScript format. The LaTeX extensions are included free of charge with all DVILASER/PS distributions.

The TeX macro package described here contains about 300 lines of code and has been dedicated to the public domain by Textset. Although the macros assume one is using Textset's DVILASER/PS program, they should be of general interest. The same strategy could be applied with other device drivers. In fact, similar extensions to LaTeX could be implemented for any printer with a reasonable graphics language.

Some of this work was funded by the University of Michigan and developed initially at the CAEN Apollo Lab, and many thanks are due to Leslie A. Olsen and our other friends at the College of Engineering.

The LaTeX Picture Environment

One of LaTeX's nicest features is the `picture` environment. With it, one can "draw" pictures. The picture in Figure 1 was created by the LaTeX commands that are to the left of it.

The "(1,1)(−150, 27)" specifies how much space the picture should take up and also allows the user to move the picture around on the page. In the example, "(1,1)" tells LaTeX to allocate hardly any space at all for the picture — just an imaginary box 1pt by 1pt, though obviously the actual picture is much larger. The "(−150, 27)" positions the picture in the desired place by moving the whole picture to the right 150pt and down 27pt from where LaTeX would have otherwise have put it, at the next place a regular character would go that has dimensions 1pt by 1pt.

Within the LaTeX `picture` environment, pictures normally are "drawn" by positioning special characters, in this case curved and straight line segments, in ways that give the appearance of circles and continuous lines. Circles are actually composed of four discrete curved segments, one for each quarter of the circle. LaTeX carefully aligns them so that

Editor's Note: this article was originally printed on an Apple LaserWriter using the Almost Computer Modern fonts, with the illustrations in Figure 1, Figure 3, Figure 4 and Figure 6 inserted by the DVILASER/PS program. For this special issue of *TUGBOAT*, it was reformatted, typeset on an Alphatype CRS using the new Computer Modern fonts, and the illustrations were reinserted manually.

```
\begin{picture}(1,1)(-150,27)
 \put(150,200){\circle{40}} % head
 \put(140,205){\circle*{10}} % left eye
 \put(160,205){\circle*{10}} % right eye
 \put(150,193){\circle{15}} % mouth
 \put(150,180){\line(0,-1){100}} % body
 \put(150,150){\vector(-1,-1){50}} % left arm
 \put(150,150){\vector(1,-1){50}} % right arm
 \put(150,80){\line(-1,-1){50}} % left leg
 \put(150,80){\line(1,-1){50}} % right leg
\end{picture}
```

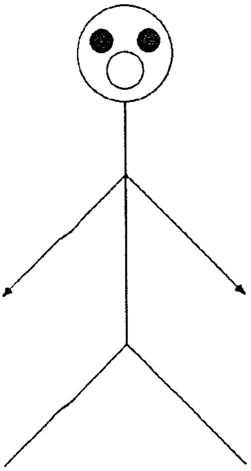


Figure 1.

Sample LaTeX picture commands, and the resulting picture.

they print as a complete circle. Oblique lines, ones that are neither vertical nor horizontal, are composed of a number of shorter segments which are placed end on end and form what looks like a continuous line. These special characters are obtained from a set of special fonts (`circle10`, `line10`, `lasy10`, etc.) that are distributed with LaTeX. Using these special fonts, LaTeX "fools" TeX into drawing pictures even though TeX doesn't really have any graphics capabilities.

LaTeX's `picture` environment is somewhat restrictive since "pictures" must be put together from the limited number of "pieces" available in the special LaTeX fonts. LaTeX carefully figures out the position of each circle or line segment. Longer lines, since they are made up of many shorter segments, cause longer execution and printing times. Even worse, creators of complex pictures often get the error "TeX capacity exceeded — sorry." This happens because TeX runs out of memory since there are too many "characters" on the page to handle. In addition, the user must be sure to use lines having only certain slopes, and be willing to accept circles of only certain diameters since in the standard LaTeX distribution there are only 36 available slopes for straight lines and 10 different diameters for circles. Also, LaTeX can only print hollow circles of diameters that are multiples of 4pt in size. If a user selects an in-between size like 21pt, LaTeX will pick the closest available size: 20pt. The biggest hollow circle it can do is 40pt — a little over half an inch — in diameter. The biggest solid circle is 15pt in diameter. The steepest non-vertical slope available is $\pm\frac{5}{6}$. LaTeX checks the slope arguments to make sure neither Δx nor Δy exceeds ±6, and that they are both integers. For vectors, because the arrowhead is drawn from a font too and slope availability in its case is even more limited, neither Δx nor Δy can exceed ±4.

```
\begin{picture}(300,300)
 \put(150,200){\circle{70}} % head
 \put(140,205){\circle*{1}} % left eye
 \put(160,205){\circle*{1}} % right eye
 \put(150,193){\circle{10}} % mouth
 \put(150,165){\line(0,-1){85}} % body
 \put(150,150){\vector(-1,-6){15}} % left arm
 \put(150,150){\vector(2,-1){50}} % right arm
 \put(150,80){\line(-1,-1){50}} % left leg
 \put(150,80){\line(1,-7){15}} % right leg
\end{picture}
```

Figure 2. *This example exceeds standard LaTeX's capabilities. Slopes specified for the left arm and right leg are too steep.*

Figure 2 demonstrates these problems by drawing the same picture with different slopes and diameters. No picture was produced when this was run under standard LaTeX because first the "−6" for the left arm exceeded ±4, caused a LaTeX error message, and halted execution; then the "−7" for the right leg exceeded ±6 and LaTeX stopped again.

When these two simple problems were "corrected" by substituting "$(-1,-4)$" for the left arm and "$(1,-6)$" for the right leg, the picture still did not look as desired — the head was 40pt in diameter instead of 70pt. This messed up the body as well, causing the head to "hover" above the body. In

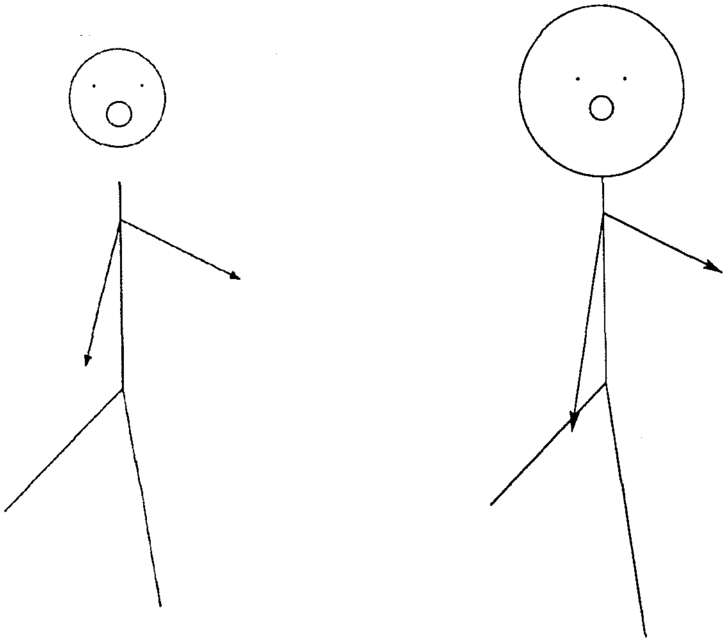


Figure 3.
The left picture was produced by standard LaTeX, the right picture by extended LaTeX.

Figure 3, the left picture is how standard LaTeX did it. On the right is the Extended LaTeX version of the original example.

The LaTeX Picture Environment Extensions

The PostScript printer language is very powerful. It has a very general and flexible set of capabilities enabling the creation of practically any textual or graphic image. It can draw letters, lines, and circles of virtually any size and slope. It can shade the inside of any closed figure it draws. And it allows lines of text to be drawn at any angle. The extended version of LaTeX uses these PostScript capabilities to provide a number of features not normally possible with LaTeX. All extensions are modelled closely after the way LaTeX normally does things.

Extensions to Existing LaTeX Commands

Line Thickness Standard LaTeX provides two line thicknesses which are selected by typing either `\thinlines` or `\thicklines` inside the `picture` environment. These two declarations simply tell TeX which font to use, either `line` or `linew`, respectively. LaTeX version 2.05 also provided a `\linethickness{}` command with which the user could vary the thickness of *non-oblique* (vertical or horizontal) lines.

The latest release of LaTeX does not document the use of the `\linethickness{}` command, probably because of the confusion which arose because it could only do non-oblique lines. With the LaTeX extensions, though, it has been brought back. By typing something such as `\linethickness{5pt}` you can set the line thickness of a PostScript-drawn line or circle to 5pt.

`\thinlines` is equivalent to the command `\linethickness{0.4pt}`. `\thicklines` is equivalent to typing `\linethickness{2pt}`. The Extended LaTeX version of `\thicklines` draws a much thicker line than regular LaTeX `\thicklines` does. Lines of intermediate thickness may be drawn with the `\linethickness` command.

Circles Circles of practically any diameter may be drawn. For hollow circles, use `\circle{`*diam*`}`. For solid circles, use `\circle*{`*diam*`}`.

Lines and Vectors Length and accuracy are virtually unlimited. Line lengths and \put coordinates need not be restricted to integer values as is the case normally with LaTeX. However, whole numbers are still required for Δx and Δy. This might seem restrictive, but if you want a line 34.5 points long with a slope of, say, 0.66, try \line (100,66){34.5}. A negative length causes the line to project opposite to the specified direction.

The \vector command uses the same syntax as the \line command and is just as versatile. The arrowhead points away from the location of the \put command. If you choose to use a thick line, you'll find the arrowhead may not look very good since it was designed to be used with thinner lines. In this case, use the \PSarrowhead command (described later) to custom-make your own vector with a bigger arrowhead.

New LaTeX Commands

PSoval If you've used LaTeX's picture environment, you have seen that the \oval command doesn't really produce an oval at all; it's actually a rectangle with rounded corners. The corners are made up of the same fonts that make circles. Because \oval produces this unique-looking figure, the original LaTeX macro was not redefined. Instead, a new command, \PSoval, was created. The syntax of \PSoval is as follows:

\PSoval{*width*}{*height*}

The \put command specifies the location of the center of the ellipse. Again, width and height are virtually unlimited.

You can get a solid oval by typing \PSoval* with the same syntax as \PSoval.

Two drawbacks — you can't use \PSoval to put text inside the oval. Use an extra \put command to do that if you need to. Also, this release of \PSoval has no provision for printing a portion of an oval — something LaTeX *can* do with \oval.

PSarrowhead Extended LaTeX allows the user to control the shape of arrowheads. The user can make almost any size or shape of arrowhead desired. Add a line and get a custom-made vector as well. The syntax of the command is:

\PSarrowhead($\Delta x, \Delta y$){*length*}{*width*}{*depth*}

where *length*, *width* and *depth* are defined in the summary. Keep Δx and Δy as whole numbers as in the \line and \vector commands. You can use almost any values you want for the length, width and depth. Use \PSarrowhead* with parameters as above to get a solid arrowhead.

The Extended LaTeX \vector command uses a solid arrowhead that has a length of 8pt, a width of 4pt, and a depth of 2pt.

PStilt Extended LaTeX, because it can use PostScript capabilities, allows the user to temporarily "rotate" the coordinate system of the page for individual lines. This means that text no longer has to be horizontal. In fact, it can be at any slope desired. The syntax of \PStilt is

\PStilt($\Delta x, \Delta y$){*object or text*}

The user can put text or a picture object like a \framebox inside the third (brace enclosed) argument to \PStilt. This addition is very useful for labelling lines and rotating LaTeX picture objects such as \oval's and \framebox'es.

PSpath Extended LaTeX allows a much easier way of drawing straight lines — by naming the coordinates of the endpoints. In fact, a series of connected lines can be drawn with just one command. The syntax of \PSpath is

\PSpath(x_0, y_0){(x_1, y_1)(x_2, y_2)$\cdots$(x_n, y_n)}

You don't even need the \put command, since the starting point is defined to be (x_0, y_0) and the n points are then connected in sequence. The number of points that can be connected using just one \PSpath varies from system to system. If you try to use a lot of points and get "capacity exceeded" errors, use more \PSpath commands.

The map example of Figure 6 demonstrates a series of points connected as one \PSpath.

Conclusion

The LaTeX extensions described in this article represent one way that TeX and PostScript can be used in a combination that is more powerful than either one alone. While the implementation described here is available only with Textset's DVILASER/PS program, other individuals or organizations might make use of a similar strategy to design their own individualized LaTeX extensions.

Note added in press

Leslie Lamport read a preprint of this article and had several much appreciated comments. He correctly guessed that we were not, at the time the article was written, aware of the new option that draws quadratic Bezier splines. He suggested for portability's sake that the extensions be enabled with a document-style option; the extensions in fact already must be enabled by a command and will work within any document style. We incorrectly stated that the \linethickness command was

Figure 4.

Summary of Syntax of LaTeX Extensions.

Control Macros

\PSextensionsOn Enables LaTeX extensions.

\PSextensionsOff Disables LaTeX extensions and reverts to regular LaTeX processing.

Redefined Macros

\linethickness{*dimen*} Sets the thickness of all lines, including sloped lines, circles, and other geometric shapes. *dimen* is something such as 0.01in or 3pt.

\thinlines Sets line thickness to 0.4pt.

\thicklines Sets line thickness to 2pt.

\circle{*diam*} Puts a hollow circle at the current location with diameter *diam*, which can be to almost any size and accuracy desired.

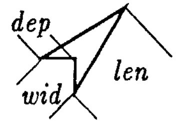
\circle*{*diam*} Just like \circle, only the circle is solid.

\line(Δx,Δy){*length*} Puts a line starting at the current point with a slope $\Delta y/\Delta x$ and a length defined as described in Lamport's book, where *length* is how far *over* the line is to go in terms of x-units, unless the line is vertical, where *length* then means the length of the line up or down. Use whole numbers for Δx and Δy. *length* can be almost any number to virtually any accuracy.

\vector(Δx, Δy){*length*} Just like \line above, except in addition, a solid arrowhead is added at the other end of the line, pointing away from the \put point.

New Macros

\PSarrowhead(Δx, Δy){*length*}{*width*}{*depth*} Puts a hollow arrowhead with its tip at the position of the \put command, pointing at a slope $\Delta y/\Delta x$ and where the arguments are as shown:



When *depth* is positive, the figure produced is a concave arrowhead as shown. But when *depth* is negative, the figure is convex, and other shapes can then be created, like diamonds and rotated squares.

\PSarrowhead*{*same as above*} Puts a solid arrowhead at the position specified by the \put command.

\PSoval{*width*}{*height*} Puts a hollow ellipse with its center at the position specified by \put.

\PSoval*{*width*}{*height*} Puts a solid ellipse with its center at the position specified by \put.

\PStilt(Δx, Δy){*object*} Puts *object* at the position specified by \put, but at a slope $\Delta y/\Delta x$. *object* can be text, or any picture environment object such as \oval, \PSarrowhead, or \framebox with their appropriate arguments included.

\PSpath(x_0, y_0){(x_1, y_1)$\cdots$(x_n, y_n)} Connects the points (x_i, y_i) with straight lines starting at the point (x_0, y_0) and ending with (x_n, y_n). Can have an arbitrary number of points, but the exact number varies from system to system.

no longer documented; see page 199 of the LaTeX manual. He pointed out that our "oval" shape is actually an ellipse, while his, though not an oval, is a convenient shape in which to insert text. Textset plans to implement several other of his suggestions into the program.

```
\setlength{\unitlength}{0.125in}
\begin{picture}(56,80)(11,-12)
\PSextensionsOn
\put(0,77){\framebox(31,3){\huge
  Directions to \bf TEXTSET}}
\put(34.5,61){\huge $\star$}
\put(31.5,58){\vector(1,1){3}}
\put(27.7,57){\bf TEXTSET}
\put(27.3,56){\sf P.O.~Box 7993}
\put(27.3,55){\sf 416 Fourth St.}
\put(25.4,54){\sf Ann Arbor, Mi. 48107}
\put(27,53){\sf (313) 996-3566}
\put(25,48){\Huge Ann Arbor}
\linethickness{5pt}
\PSpath(0,74){(3,72)(5,68)(5,40)(40,3)(56,3)}%I-94
\thicklines% Jackson/Huron
\PSpath(0,71){(10,67)(24,69)(40,68)(56,68)}
\PSpath(17,72){(24,69)}  % Dexter
\PSpath(9,72){(9,63)(24,42)(46,42)(56,38)}% Stadium
\PSpath(13.3,57.1){(41,65)(51,65)}  % Liberty
\put(51,0){\line(0,1){75}}  % State
\PSpath(21,11){(41,31)(41,52)(45,80)}  % Main St.
\put(42.5,61,8){\line(12,-12){13.6}}  % Packard
\put(41.4,54.5){\line(1,0){14.5}}  % Hill
\put(36,62){\line(1,0){15}}  % William
\put(36,63.7){\line(0,-1){7}}  % Fourth St.
\put(51,63.5){\line(1,0){5}}  % N. University
\thinlines \put(56,23){\line(-23,57){23}}% RR Track
\multiput(54.7,25.5)(-0.5,1.24){44}%
{\line(57,23){0.5}}% Ties for RR Track
\put(44,5){\framebox(6,4){\parbox{0.6in}%
  {\sf Briarwood\\Mall}}}
\put(5,70){\framebox(3,4){\sf Plaza}}
\put(34,37){\framebox(6,4){\parbox{0.4in}%
  {\sf Pioneer High\\School}}}
\put(51.5,58){\framebox(4,4){\parbox{0.4in}%
  {\sf Univ of Mich}}}
\thicklines
\put(11,16){\circle{2}}  % Ann Arbor
\put(17.5,14){\circle{4}}  % Detroit
\put(13,8){\circle{3}}  % Toledo
\put(4,4){\framebox(16,16){}}  % Inset Frame
\linethickness{5pt}
\put(42.5,44.5){\PSoval{2}{3}}  % Michigan Stadium
\put(44,46){\sf Michigan} \put(44,45){\sf Stadium}
\thinlines
% Inset to follow
\PSpath(18,14){(10,14)(9,16)}  % I-94
\put(9,16){\vector(-1,0){4}}  % I-94
\put(13,7){\line(0,1){13}}  % US 23
\put(17,7){\line(-1,0){13}}  % Ohio Turnpike
\put(17,7){\vector(1,-1){2}}  % Ohio Turnpike
\PSpath(10,4){(16,10)(20,18)}  % I-75
\thinlines  % Ohio-Michigan Border
\multiput(4,9.5)(0.5,0){32}{\line(1,0){0.25}}
\put(1,59){\framebox(3,2){\large \bf I-94}}
\put(53,4){\framebox(3,2){\large \bf I-94}}
\put(45,1){\large \it Exit 177}
\put(29,17){\large \it Exit 175}
\put(-1,67){\large \it Exit 172}
% street names to follow
\put(28,19){\PStilt(1,1){\small \sf
  ANN ARBOR - SALINE ROAD}}
\put(27,42.4){\small \sf STADIUM BLVD}
\put(8,63){\PStilt(0,1){\small \sf STADIUM}}
\put(14,66.5){\PStilt(14,2){\small \sf JACKSON}}
\put(20,71){\PStilt(7,-3){\small \sf DEXTER}}
\put(29,69.4){\PStilt(16,-1){\small \sf HURON AVE}}
\put(29,62.4){\PStilt(28,8){\small \sf LIBERTY ST}}
\put(37,56){\PStilt(0,1){\small \sf FOURTH ST}}
\put(44,62.5){\small \sf WILLIAM}
\put(52,64){\small \sf N UNIV}
\put(52,55){\small \sf HILL ST}
\put(46.3,58.4){\PStilt(1,-1){\small \sf PACKARD}}
\put(54.3,50.4){\PStilt(1,-1){\small \sf ROAD}}
\put(50,23){\PStilt(0,1){\small \sf STATE}}
\put(50,43){\PStilt(0,1){\small \sf STREET}}
\put(40.5,46){\PStilt(0,1){\small \sf MAIN}}
\put(43,70){\PStilt(4,28){\small \sf STREET}}
\put(5,18){\bf ANN} \put(5,17){\bf ARBOR}
\put(5,15){\small \it Chicago}
\put(5,14){\small \it 225 Mi}
\put(10,13){\small \sf I-94}
\put(12.5,16.5){\large \bf DETROIT}
\put(10.2,11){\small \sf US 23}
\put(16.5,10){\small \sf I-75}
\put(4.4,6){\small \sf OHIO TURNPIKE}
\put(11,4.5){\small \it Cleveland 100 Mi}
\put(14.5,8){\bf TOLEDO}
\put(6,9.8){\tiny Michigan}
\put(6.8,8.8){\tiny Ohio}
\put(52,1.8){\it To Detroit}
\put(0.5,74.5){\it To Chicago}
\put(14.5,13){\PSarrowhead(0,1){0.5}{1}{0.35}}
\put(14.45,12.2){\framebox(0.1,1){}}
\put(13.3,11.4){\tiny Detroit}
\put(13.5,10.8){\tiny Metro}
\end{picture}
```

Figure 5. *This extended LaTeX code and the DVILASER/PS system produced the map in Figure 6 (facing page).*

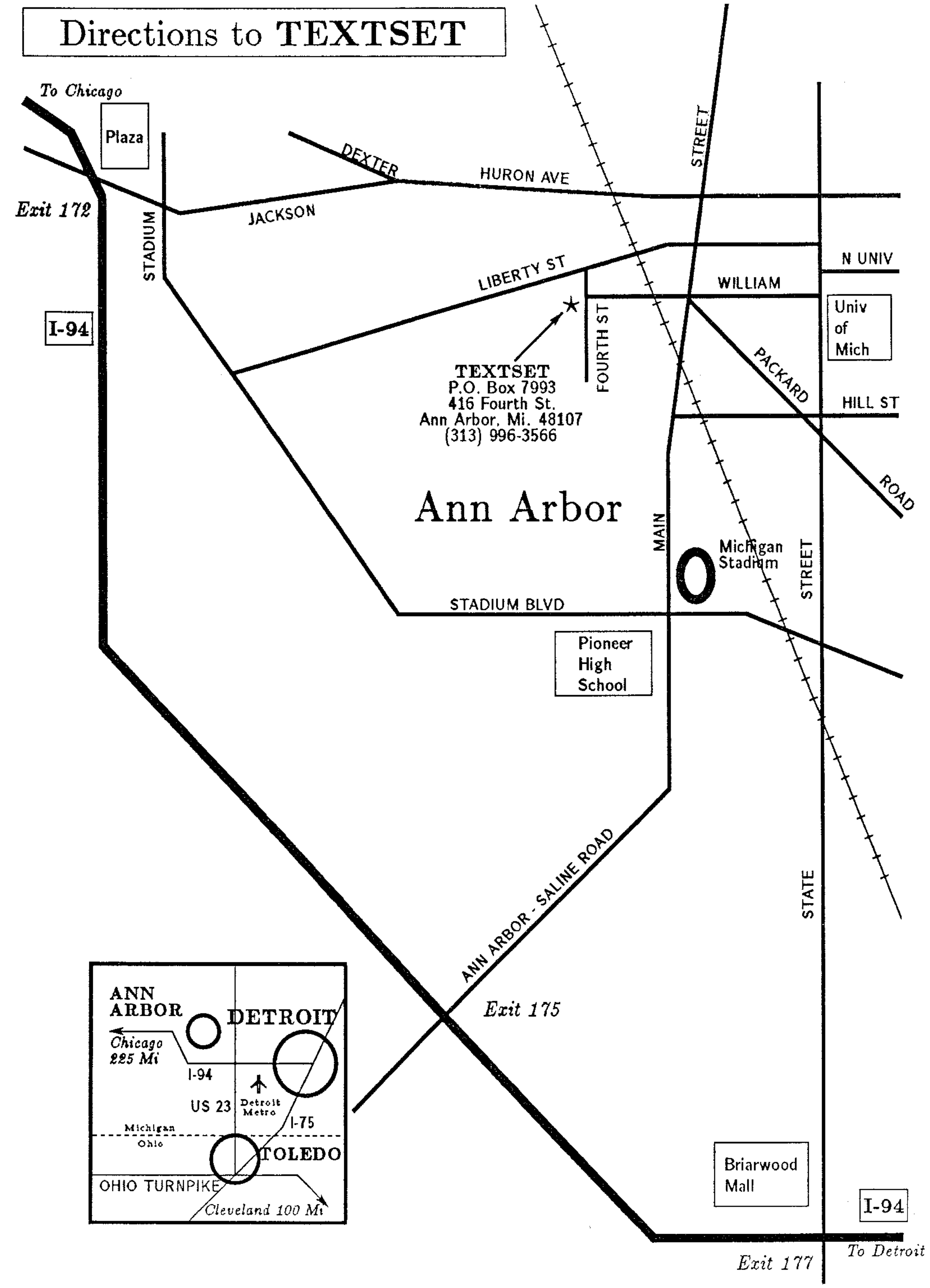

Directions to TEXTSET
To Chicago
Plaza
Exit 172
I-94
STADIUM
JACKSON
DEXTER
HURON AVE
STREET
LIBERTY ST
WILLIAM
N UNIV
Univ of Mich
FOURTH ST
PACKARD
HILL ST
ROAD
TEXTSET
P.O. Box 7993
416 Fourth St.
Ann Arbor, Mi. 48107
(313) 996-3566
Ann Arbor
MAIN
Michigan Stadium
STREET
STADIUM BLVD
Pioneer High School
ANN ARBOR - SALINE ROAD
STATE
Exit 175
ANN ARBOR
DETROIT
Chicago 225 Mi
I-94
US 23
Detroit Metro
I-75
Michigan
Ohio
TOLEDO
OHIO TURNPIKE
Cleveland 100 Mi
Briarwood Mall
I-94
To Detroit
Exit 177

Figure 6.