The extensible drawing editor Ipe*

Otfried Schwarzkopf
Department of Computer Science, Utrecht University

A paper in Computational Geometry typically contains quite a number of figures, and many of us spend a significant amount of time on the creation of these figures. Usually, we use some tool such as `xfig` or `idraw` to create a Postscript file, which is then included into a LaTeX-document. This works quite well for simple figures, but for more difficult drawings the available tools are just not very satisfying. Ipe is a drawing editor that has specially been written to be useful for the kind of figure that a computational geometry paper might contain. It runs under X11, and currently requires a color screen. In the following, I am trying to list the properties that make Ipe different from other drawing programs.

1 Support for LaTeX-text

If you write your papers using LaTeX or some other TeX-based system, you probably like to be able to put TeX-text into your drawings. This not only means that you can type `$x_i$` to get "x_i" instead of having to manually place a little letter "i" next to the "x", but also that you can use the macros from the main paper in the figures as well, and finally the fonts in the figure will nicely match those in the document. However, entering the text in the figure in LaTeX format usually means that you are left to trial and error to figure out whether the text fits in a box or circle, or doesn't overlap other text. Ipe overcomes those problems because it runs TeX in the background. Whenever you enter a piece of text, Ipe simply asks TeX how big the text is, and will show you a bounding box of the proper size. Furthermore, Ipe can LaTeX the whole figure for you, convert it to Postscript, and show it to you in a second window using Ghostscript.

2 Snapping

Ipe is capable of making the objects that you have already drawn "magnetic." This means that once the cursor comes near the object, it automatically "snaps" to the object. This makes it very easy to draw objects that share endpoints, that lie on other object's boundaries, or even to use intersection points between objects. You can easily draw c-oriented polygons, or you can move objects in a direction parallel to another object's edge. A typical example is demonstrated in Figure 1. To draw the two vertical extensions through the endpoints p and q of segment e, you use vertex- and edge-snapping and turn on the 90°-oriented mode. As if by magic, you then only have to click once near p and once near the segments s_1 and s_2 to get the vertical extension of p.

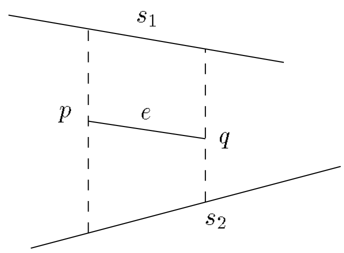

Figure 1:

3 Extensibility

The problem with most drawing programs is that they have a fixed set of commands, whileas in a computational geometry paper you often wish that you had a special function that, say, draws the circle from which a given segment is visible under a certain fixed angle. Ipe allows you to extend its functionality by adding your own code. Your self-defined functions can be used like any other Ipe function, they can appear in menus and can have keyboard shortcuts. The interface is rather simple to use, and allows you to incorporate your own geometric code in Ipe. Among the Ipe user macros that have already been written by me and other Ipe users are the following:

- Importing `idraw`-files, such that they can be saved and used as Ipe-files.
- Importing Postscript files, turning them in to normal Ipe files. This works for the Postscript output generated by many packages, such as `gnuplot`, `showcase`, etc.
- Clipping Ipe objects to a given polygon.

*This work was supported by the Netherlands' Organization for Scientific Research (NWO).

- Computing convex hulls, Voronoi diagrams, Delaunay triangulations, higher order Voronoi diagrams and medial axes. (This is based on the `qhull`-code by C. Bradford Barber and Hannu Huhdanpaa.)

- Computing the Minkowski sum and difference of polygons.

Using these functions, making an illustration for Fortune's algorithm such as Figure 2 takes only a few minutes. Figure 3 was created by importing a Postscript file

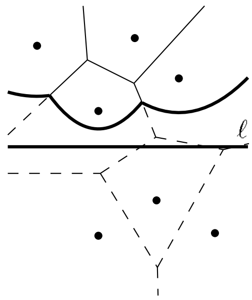

Figure 2:

with a map of the Netherlands, marking the provinces' capital cities, computing their Voronoi diagram and clipping that to some auxiliary polygon, which has been deleted afterwards. Again, this takes only a few minutes. The tiger in Figure 4, finally, has been created by

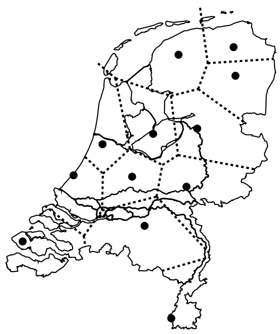

Figure 3:

importing a Postscript file in the Ghostscript distribution and some slight editing.

4 Making transparancies

Ipe can also be used to prepare transparencies for a presentation. For this purpose, it has a multi-page mode where you can have a template that appears on every page. There is also a special "minipage" text object that can contain `itemize` and other LaTeX-environments. In effect, using Ipe this way is very much like a WYSIWYG version of SliTeX.

5 Ipe as a visualization tool

It is very easy to create an input file for Ipe. In fact, most of the geometric software currently written at our department has a button to save the current image as an Ipe file. This makes it a matter of seconds to create a figure for a paper, a figure that can still be edited and labeled. In the same spirit, it is possible to use Ipe to create test sets for your implementations.

6 More on Ipe

Ipe is fully documented and comes with a 50-page manual. The manual is also available on-line, and there is a "mouse explainer" that indicates the function of the mouse button at any given moment. Ipe can undo a certain number of operations.

The object types available in Ipe are chosen to be convenient for users in computational geometry: Circles can be entered by specifying center and radius, a diameter, or three points on the boundary. Similarly, there a different input modes for the other objects, and a selection of marks that can be used to indicate point sets, mark vertices differently, and the like.

Ipe stores drawings in a single file. This file is at the same time a valid Encapsulated Postscript file, a TeX source file, and it contains the information that Ipe needs to read it back. You do not need to carry around three different files—a master file, a LaTeX-version and a Postscript version—as some other drawing programs require.

Ipe is highly configurable. Several options can be set in a special configuration window, others are determined by Ipe's X resources.

Figure 4: Find the differences!

More information about Ipe is available on the world wide web.[1] This includes the full Ipe manual, a list of all currently available Ipe user macros, and the Ipe Frequently Asked Questions.

[1] The resource locator for the Ipe page is `http://www.cs.ruu.nl/people/otfried/html/ipe.html`. If you have no access to the world wide web, you can ftp to `ftp://ftp.cs.ruu.nl/pub/X11/Ipe/`.