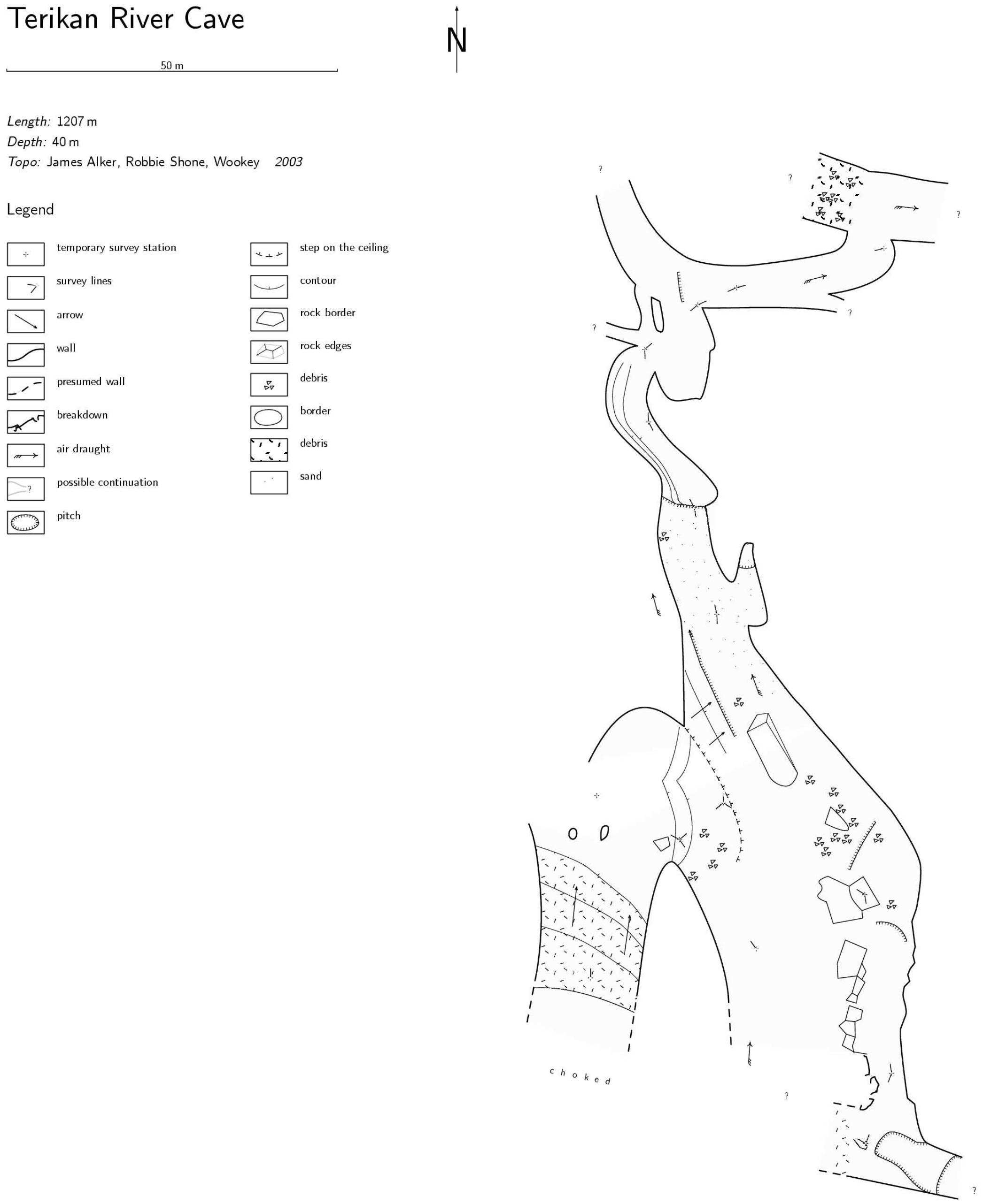
ISSN: 1361-8962

Issue 33

Compass Points

March 2004

BCRA



Drawing Surveys with Therion

An Electronic Compass/Clino

The Journal of the BCRA Cave Surveying Group

COMPASS POINTS INFORMATION

Compass Points is published three times yearly in March, July and November. The Cave Surveying Group is a Special Interest Group of the British Cave Research Association. Information sheets about the CSG are available by post or by e-mail. Please send an SAE or Post Office International Reply Coupon.

NOTES FOR CONTRIBUTORS

Articles can be on paper, but the preferred format is ASCII text files with paragraph breaks. If articles are particularly technical (i.e. contain lots of sums) then Latex, OpenOffice.org or Microsoft Word documents are probably best. We are able to cope with many other formats, but please check first. We can accept most common graphics formats, but vector graphic formats are much preferred to bit-mapped formats for diagrams. Photographs should be prints, or well-scanned photos supplied in any common bitmap format. It is the responsibility of contributing authors to clear copyright and acknowledgement matters for any material previously published elsewhere and to ensure that nothing in their submissions may be deemed libellous or defamatory.

COMPASS POINTS EDITOR

Anthony Day, Vollsveien 86A, 1358 Jar, Norway. Tel: +47 67 15 65 39
E-mail: `csg-editor@survex.com`

SUBSCRIPTION & ENQUIRIES

Andrew Atkinson, 31 Priory Avenue, Westbury-on-Trym, BRISTOL, BS9 4BZ Tel: 0117 962 3495
E-mail: `csg-secretary@survex.com`

PUBLISHED BY

The Cave Surveying Group of the BCRA. BCRA is a registered charity.

OBJECTIVES OF THE GROUP

The group aims, by means of a regular Journal, other publications and meetings, to disseminate information about, and develop new techniques for, cave surveying.

SUBSCRIPTION RATES (FOUR ISSUES)

U.K. - £4.50 Europe - £6.00 World - £8.00

These rates apply regardless of whether you are a member of the BCRA. Actual "membership" of the Group is only available to BCRA members, to whom it is free. You can join the BCRA for as little as £5.00 – contact BCRA directly for details. Send subscriptions to the CSG secretary (see "subscriptions and enquiries"). Cheques should be drawn on a UK bank and payable to *BCRA Cave Surveying Group*. Eurocheques and International Girobank payments are acceptable. At your own risk you may send UK banknotes or US$ (add 20% to current exchange rate and check you don't have obsolete UK banknotes). Failing this your bank can "wire" direct to our bank or you can pay by credit card, if overseas. In both these cases we have to pay a commission and you should pay 5% extra to cover this.

BACK ISSUES

Past issues of *Compass Points* are available from the secretary (see "subscriptions and enquiries") subject to availability. Cost is £1.25 per issue, plus postage and packing at rates of £0.50 (UK), £1.50 (Europe) or £3.00 (world). Published issues are also accessible on the Web via the CSG Web pages at `http://www.bcra.org.uk/csg/`

DATA PROTECTION ACT (1984)

Exemption from registration under the Act is claimed under the provision for mailing lists (exemption 6). This requires that consent is obtained for storage of the data, and for each disclosure. Subscribers' names and addresses will be stored on computer and disclosed in an address list, available to subscribers. You must inform us if you do not consent to this.

COMPASS POINTS LOGO

courtesy of Doug Dotson, Speleotechnologies.

CAVE SURVEYING MAILING LIST

The CSG runs an e-mail list for cave surveyors around the world. To join send a message containing the word 'subscribe' in the body text to `cave-surveying-request@survex.com`

CONTENTS
of Compass Points 33

The journal of the BCRA Cave Surveying Group

Cover image: Excerpt from a survey of Terikan River Cave, Mulu, drawn using Therion

Editorial

Much of this issue is taken up with Wookey's epic tome on the virtues of Therion, a dedicated cave survey drawing package. It is clear that the use of computers to draw surveys is becoming increasingly popular, and it would interesting to hear about the advantages and drawbacks of the drawing solutions that others have employed. The other major article in this issue concerns a project to make an electronic compass/clino. unit. Building such a device at an affordable price has been on many surveyors' wish list for a number of years, and it appears that significant progress is being made. Are there any other similar projects out there?

Forthcoming Events

Summer Field Meet

The CSG field meet advertised in the last issue has been postponed to the Summer and diverted North. It will now be held at the headquarters of the Red Rose Cave and Pothole Club, Bull Pot Farm in Cumbria, on 19th/20th June. We will be doing an expedition surveying workshop, suitable for giving those who haven't surveyed at all a crash course in how to do it, as well as comparing notes with the more experienced on the most appropriate techniques and the latest hardware. Discussion of options for base-camp computing/printing/drawing-up/GPS management will also be covered.

All welcome, particularly those new to expeditioning who realise that it's a good idea to have a go at surveying before you end up at the top of a 100m shaft in a foreign country with only the vaguest idea what to do.

There will also be a demonstration of the Therion and TunnelX cave-drawing software packages.

Further details about the meet can be obtained from Wookey, email: `wookey@aleph1.co.uk`

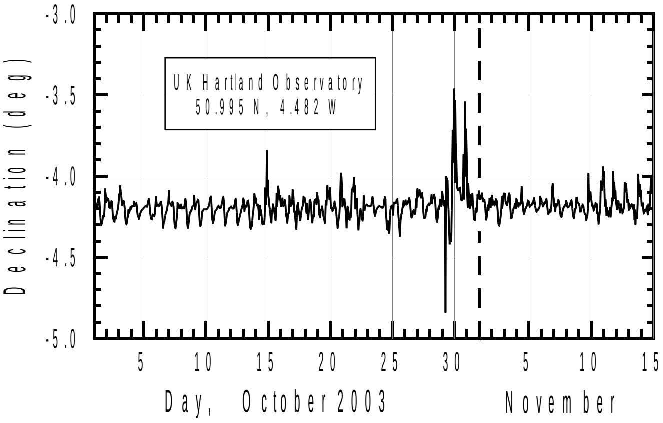
Snippets

Magnetic Storms

Bob Thrun

This is a followup of my article Hourly Variation of Magnetic Declination that appeared in Compass Points No. 18 and Compass & Tape No. 43. In that article I mentioned that the declination, the difference between true north and magnetic north, is affected by sunspot activity. At the end of October 2003, there were two large solar flares that caused geomagnetic storms on Earth. The first of these was reported in the newspapers to be the largest in many years and it was followed by another large storm about a day later. I decided to see how much a large magnetic storm could affect a compass reading.

I got hourly means of magnetic declination measured at the Hartland magnetic observatory from a British Geological Survey website `http://www.geomag.bgs.ac.uk/gifs/hourly_means.html`. These data are in a more convenient form than I could get from an American observatory. These data are plotted opposite. Note that October 31 goes from 31.0 to 32.0. The IGRF magnetic model predicts a declination of –4.243°at the site.



Full Tilt Ahead?

Dave Edwards

Over the course of the past three years, members of the South Wales Caving Club have been involved in a project to design and build an electronic compas/clino. unit. At present they have succeeded in building a clinometer, which is in regular use, and have made good progress towards building a working compass. Here, Dave Edwards describes how they did it.

Most of this story can be blamed on Ian Todd. It was his idea that an underground surveying unit ("InCompass", a compass/clinometer) could be made, affordably, by using some increasingly available electronic gizmos. He had found a magnetic field sensing chip which, when combined with a tilt sensor and microcontroller, could sense compass direction accurately [1]. The tilt sensor would both allow the compass to be used while tilted, and supply clino. data. He intended to use a laser pointer as a pointing device. The idea was to aim the laser pointer at the remote survey station and read both heading and inclination data at the push of a button. The tilt sensor also required a microcontroller to process its output. This was the first of many problems; neither of us had ever played with a microcontroller!

Here then was the first step, to learn what made a microcontroller tick. Essentially, a microcontroller is a single chip computer with a limited range of instructions (35 in our case). Ours is capable of processing 1 million instructions per second! Each instruction is very limited in capability, being no more than, for example, get a piece of information from one location and put it into the working register; or add/subtract two numbers. This means that even the simplest job can take hundreds of separate instructions. This sequence of instructions constitutes the program that will do the job required.

The early learning was rather slow and painful. We purchased several PIC16F84 processors. These are amazing little things, and fortunately very inexpensive. Physically programming them turned out to be fairly easy as designs for the programmer hardware, that can be driven from an ordinary pc, are available in the public domain via the internet. We built our own, but they are also available pre-built or as kits from many outlets [2]. Armed with a sheaf of Application Notes from Microchip's website [3] we spent many hours writing and testing our assembled code to perform some simple tasks, like lighting LED's in a particular sequence!

The next step was to obtain some tilt sensors. We chose the ADXL202 accelerometer from Analog Devices [4]. This is micro-machined in silicon, so the whole device fits into a small surface mount Integrated Circuit package. It consists of two sensors mounted at right angles to each other, each of which is effectively just a thin beam with a suspension at each end. When this beam is perpendicular to the gravity field, i.e. horizontal, there is no lateral movement, but when it is not horizontal the beam moves under gravity against the suspensions. The degree of movement is proportional to the sine of the angle to vertical. We originally purchased an "evaluation" board which required only a simple power supply. The X and Y output signals it generated are simple square waves of a frequency which can be preset by external components. When the sensor is horizontal, the square wave has a nominal 50% duty cycle (ratio of high to low signal times) and, as the sensor is tilted, the duty cycle either increases or decreases depending on whether the sensor is tilted up or down. The maximum variation is not very great, being only plus or minus 12.5% about the centre (horizontal) point.

We chose a frequency of 1 kHz and programmed the microcontroller to continuously check the X and Y outputs to determine whether they were high or low and used the controller's timer function to determine the pulse widths. To improve accuracy, by reducing signal noise, the signal was sampled 64 times and the readings averaged. As the duty cycle is proportional to the sine of the angle we decided on using a look up table stored in a (non-volatile) memory chip external to the processor, to translate the duty cycle to the angle being measured. To talk to the memory required learning to program the PIC for I^{2}C communication. This turned out to be a useful skill, as many of the chips we would subsequently use in the InCompass project also use this communication protocol.

Initially a liquid crystal display was used as the display device, and some considerable time was spent in learning how to get the microcontroller to talk to it. When we started to calibrate the tilt sensor against known angles we found, to our surprise, that the sensors did not appear to conform to the specifications in the data sheet. After some considerable head scratching, and contact with America to check our algorithms, we discovered that our units pre-dated a known problem and were defective! A further set of samples were despatched from the USA and were also defective. At the third attempt, we received the correct units. The algorithms which we had used were derived from an American Application Note and were designed to make the mathematical calculations as easy as possible to apply to a microcontroller program. They did not, however, fully compensate for the effects of temperature changes on the sensor. It soon became apparent that this was a potential problem. While this would not have proved too bad if the device was calibrated and used in the nominally constant underground temperature of a cave, it did preclude calibration and use at surface temperatures and subsequent, accurate, use in a cave without underground recalibration. Dave came up with an algorithm which overcame this temperature problem and allowed accurate results even when calibration was done at a different temperature.

At this time, cheap laser pointers started to appear on the high street and so several were obtained and seemed ideal for the job. Running from three 1.5 V button cells at about 30 mA, they were easy to cannibalise and incorporate into our unit. The laser diode was removed from the tubular body, the push-button switch removed and the button cells discarded. By incorporating a single diode into the supply line, the laser was successfully run from the surveying units' 5V supply. Although claiming to be “safe”, normal precautions regarding eye safety must be taken with these devices!

As our expertise with microcontrollers increased, Dave branched off and started to play with LED displays. Although requiring more battery power, their brightness level could be increased and decreased through software and they were more robust and, being self luminous, more easily read underground.

It was decided that it would be a good idea to try out the tilt sensor in a cave environment whilst proceeding with the compass separately. With input from Brian Clipstone, a fellow member of the South Wales Caving Club, as to what he would like a clinometer to do, and compromises with reality, a basic unit was thrashed out. It was decided that the unit would only measure tilt in one direction (i.e. the “X” axis or “pitch”) to an accuracy of 0.25°. The LED display could show five digits so a downward tilt could show, for example, “-15.75”. Upward tilt showed the angle without the minus sign. Above 75°, the accuracy fell off somewhat so above this angle the display was made to show “- - - - -” so that inaccurate readings would not be recorded.

Other than the main on/off switch, a single push button switch was all that was required. It would, in three consecutive operations: 1) switch on the laser and the LED display; 2) switch off the laser and hold the current reading on the display; and 3) switch the display off again. All these requirements were successfully incorporated into the microcontroller program and work was started on bringing all the elements together into a working unit.

When completed as a simple prototype (mainly perspex and glue – see Figure 1) it was necessary to calibrate the unit for an accurate “zero”. This was achieved with an 18 metre garden hose with clear plastic tubes at each end for a water level; a garden fork with a strip of reflective tape at the far end, and a pile of bricks at the other. By keeping the laser beam level with the water level at each end, the tilt sensor could be adjusted to read “00.00”. It turned out that cheap laser pointers frequently had a beam which was not co-axial with the body. For the clino., it was only necessary to rotate the laser until the discrepancy was purely horizontal, but for the compass we will need to build an adjustable holder, so the beam can be aligned both vertically and horizontally with the compass body.

Since completing the prototype and handing it to Brian, it seems to have been in surveying use nearly every weekend for about 18 months now. He says he has found that it reduces the vertical error in survey loops by a significant amount. Just try prising it away from him!

In the meantime, the compass has been steadily progressing. Ian had upped the spec. of the microcontroller to a Microchip PIC16F876 for its considerably increased working memory. It also has a “capture and compare” function to enhance the reading the of the ADXL202. This, with a larger look-up table, has permitted the tilt resolution to be improved to one tenth of a degree. The compass circuit was put together from a mixture of a Philips data sheet for their KMZ51/52 compass chips [5], and various application notes from Honeywell for their HMC1001/1002 magnetoresistive sensors [6]. We settled for the Honeywell devices as they are more readily available in small quantities.

The Compass/Clino. is currently at the position of having a completed PCB with all components in surface mount. The analogue X, Y, Z outputs of the compass look most promising and the conversion to digital form, and subsequent storage in the PIC memory, works well. The software is designed to allow the compass to self-calibrate following the taking of six random compass readings. This should allow the determination of X,Y,Z channel gain and offset errors so that corrections can be applied. This involves using matrix maths to solve five simultaneous equations (matrices probably weren’t invented when Ian was at school!). Whilst these routines are written, they are still not working properly, so our current position is trying to track down the bugs. We feel that we are getting very close to a working unit and then we will be able to gauge the degree of accuracy which might be reliably expected. This will be reported on in greater detail at a later date.

This project has now involved us for over 3 years and we still seem to have a way to go. Still, it beats football.....

This article is based on a similar article entitled “No Sense of Direction” that appeared in the most recent issue of the South Wales Caving Club Newsletter.

References

[1] Caruso, M.J. & Withanawasam, L.S. Vehicle detection and compass applications using AMR magetic sensors, online at: `http://www.ssec.honeywell.com/magnetic/datasheets/amr.pdf`

[2] Dontronics home page, online at: `http://www.dontronics.com`

[3] Microchip graphic explorer homepage: microchip website, online at: `http://www.microchip.com`

[4] Analog devices: home, online at: `http://www.analog.com`

[5] Electronic compass design using KMZ51 and KMZ52, online at: `http://www.semiconductors.philips.com/acrobat/applicationnotes/AN00022_COMPASS.pdf`

[6] Honeywell SSEC – magnetic sensors features components, online at: `http://www.ssec.honeywell.com/magnetic/features_comp.html`

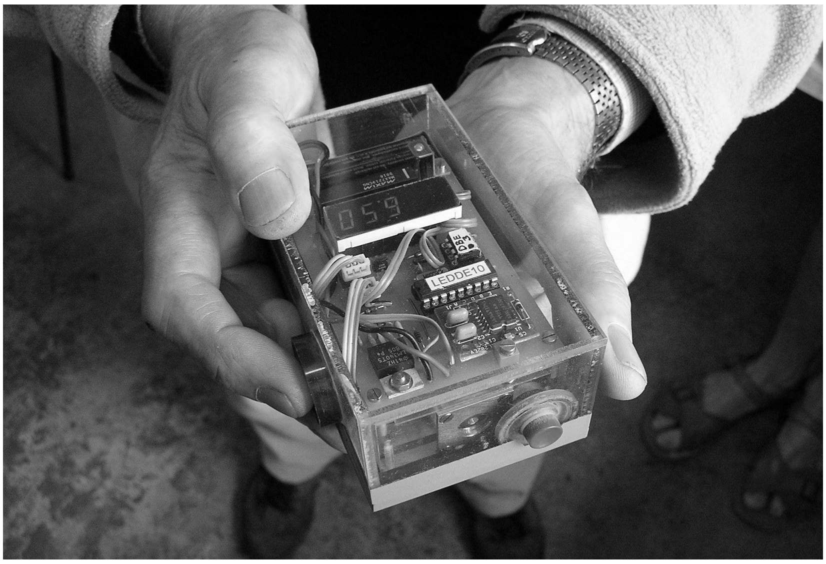


Figure 1: The prototype clino unit.

Therion - State of the Art Cave-Drawing Software

Wookey

Using a computer to draw cave surveys is becoming increasingly popular. Therion is one of the specialist cave survey drawing software packages currently available. Here, Wookey describes its capabilities and presents a worked example of how it may be used to create a complete survey.

Drawing up on computer instead of paper has been steadily increasing over the last few years as drawing software improves and computers become more widespread and faster, and the software gets fancier. However, whilst it has some advantages, particularly in terms of labelling and printing out at different scales, there are still some significant disadvantages, particularly for larger caves that are still being surveyed or explored. The fundamental problem is that standard drawing software has no mechanisms to cope when new surveys, loops or entrance fixes change the shape and relative positions of passages. For example if an upper level passage crosses in a slightly different place after some new surveys are added then the dotted and non-dotted sections need to be changed (see Figure 1).

There are also numerous detailed aspects of cave-drawing that standard software is not ideal for: dashed pitch-head symbols, boulders, and splitting of surveys into multiple sheets are some examples. In Compass & Tape issue 49, John Ganter described some techniques for drawing pitch edge ticks in Aldus FreeHand [1], and whilst his techniques worked reasonably well he concluded that the optimum solution was to use postscript.

Postscript is a language to describe graphics, most familiar as the basis of PDF files, and is a fundamental aspect of Therion. Pitch ticks is just one of the many cave-drawing problems that Therion's graphics description approach solves well. It also deals with updated data adjusting station positions, producing multiple layer surveys, and internationalization issues like different cave symbols and character sets in labels.

Therion is at its heart a language for describing cave-surveys which can be processed to produce finished surveys. This separation of the survey description from the survey representation is fundamental to the way Therion works and whilst it means that Therion can never be an exact WYSIWYG program, it does make a very versatile system. For example it is easy to change the symbol set used so the survey can be plotted using BCRA, UIS, or ASF symbols. Similarly the plot can be a single-sheet or the multi-page atlas form which is popular in much of Europe. Sheets can be up to 5m square (the maximum size of a PDF file).

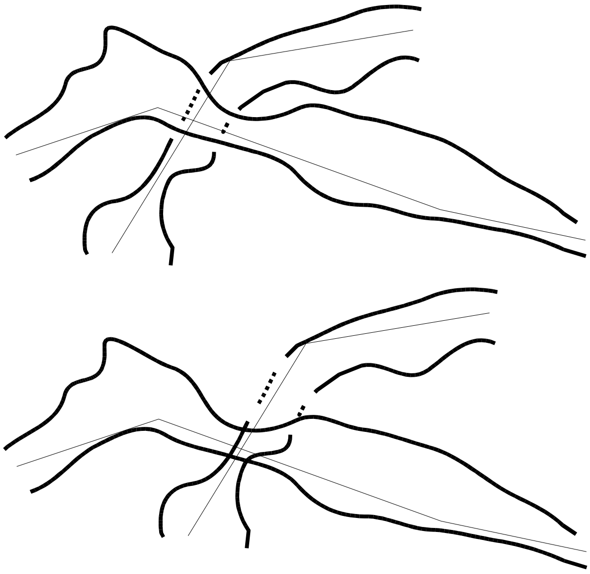


Figure 1: The dashed section denotes a passage that passes beneath a higher passage (top), but after loop closure adjustments the dashed section may no longer be in the right place (bottom).

The Therion language is just text, much like Survex data files, so in principle it is possible to enter all the survey description using a text editor. However as a great deal of this data is co-ordinates describing wall positions and passage shapes, this would clearly be an almost impossible thing to do in practice. A graphical editor is the only sensible way of describing the passage walls, boulders, pitch edges and the relationships between them, and so one comes with Therion: Xtherion. It is only one possible graphical front-end; other designs are possible and encouraged so long as they produce valid Therion data files.

Whilst Therion is currently useful for 2-D drawing, its authors intend to extend it to also cover 3-D cave modelling in the future. Currently it produces Survex .3d or Compass .plt files, as well as PDF survey files, but full modelling will be introduced as the software develops. The software is written by Stacho Mudrák and Martin Budaj (from the Czech Republic and Slovakia respectively).

Software

Therion is entirely free software (that's free as in "free market" not free as in "free offer" for those of you still unfamiliar with the concept), licensed under the GNU General Public License (GPL). That means you can copy and change the software so long as you pass on any of your own changes if you distribute it further. This means anyone can easily help improve Therion.

It uses other major free software components to do it's work. TeX document processing software and the associated MetaPost graphics description language, Postscript and PDF page description language, and Survex for centreline processing. The fact that all these programs are licensed as free software means that they can be used together in this way, and are also widely available at little or no cost - either included in your GNU/Linux distribution or downloadable for Windows, MacOS and other more obscure platforms.

How the components fit together internally is complicated, and we'll come to that later, but the process of using it is reasonably straightforward.

Xtherion

Xtherion provides a way of editing Therion files by viewing scanned images (see Figure 2 for an example) and drawing over the top of them. It also provides a survey data editor to ease the entry of survey data. It is written in the Tcl/Tk language and thus runs on Unix, Windows or MacOS and for most of us is the primary interface to Therion. It tries to make the drawing of caves as easy as possible.

Using Xtherion

First you enter the centreline data, either within Xtherion or by referring to existing Survex data files (the format is extremely similar but not identical), then you scan in each page of survey notes, or (if the sketches are too vague or distorted to use directly) sections of drawn-to-scale plan and elevation along with cross sections from the notes. Then you draw around the scans, describing which parts are plans, elevations or cross sections, where the stations are, pitches, rocks, floor details etc., and specifying scaling points for bits of picture that don't have two stations on.

Each little bit of picture (cross-section, segment of plan) is called a "scrap" and these are the building blocks Therion uses. The fundamental thing about a scrap is that it mustn't overlap itself. This is one of the things that lets you decide where to put your scrap joins. The other considerations are that joins at simple areas of low detail are a good idea, and that you don't want to make your scraps too big,

Finally you describe which scraps join to which other scraps, or if necessary which walls join up, and what text to place where, then let Therion process all this to produce a set of images - normally a plan and elevation.

You have complete control over what things are collected into one image and how things are placed so almost any kind of final output is possible.

For amusement value Xtherion also includes a blood alcohol calculator that takes your age, gender, what you drank, how much of it and when and tells you if you are too pissed to draw decent surveys or not.

Not drawing to scale?

This process is really designed for drawing-to-scale survey techniques - i.e. the original sketches are quite close to being accurate, so not too much stretching/distortion is needed. If you don't draw to scale then entering your sketches directly into Xtherion may not produce the best results. Drawing up a correctly scaled version before scanning is best. Exactly how good a job Therion makes of inaccurate sketches is something that still needs testing.

When scanning drawings remember not to make your scanned scraps too large. Large scans use up a lot of memory when uncompressed into Xtherion's memory, and if your scan is too big Xtherion will be unable to zoom in. On my 64Mb machine a 4Mb scan could not be zoomed beyond 100% without Xtherion crashing. In practice if you keep each scrap to being less than a few hundred metres of cave passage and less than 5% of the available memory size then you should be fine. Don't forget that this is the uncompressed size of the scans – if they are GIF format then 200K files will uncompress to about 4Mb.

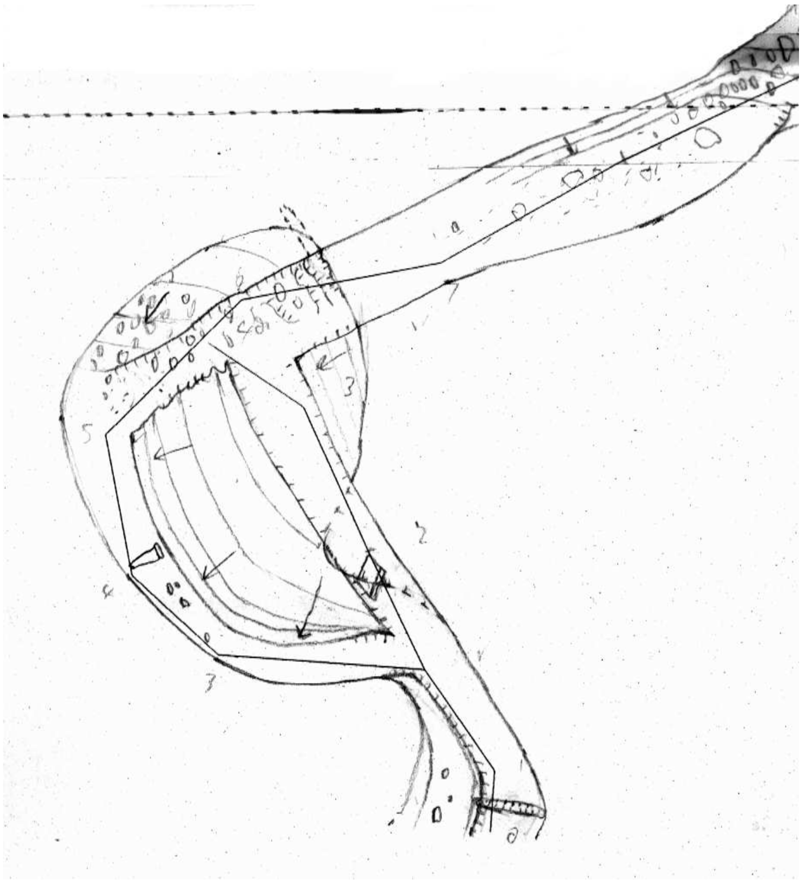


Figure 2: A scan of a scale drawing of the soundriver plan survey.

Documentation

The documentation has improved enormously in the last year, as Martin has written "The Therion Book", which provides fairly comprehensive coverage of the Therion language and capabilities as well as a useful tutorial on how to draw a cave. However, I have found from experience that it is not easy to get started by reading that document so this article tries to fill in the missing info by giving a step-by-step guide. The Therion Book will no doubt be updated soon to improve this aspect.

An example

The best way to explain how the software works is to work through an example. I will use the "soundriver" area of cave from from the Mulu 2003 expedition as I'm currently drawing that up. I'll cover how to use the software and also review the things I did and didn't like about it.

Installation

As I'm using Debian GNU/Linux which has Therion and Survex included in the distribution I just type `apt-get install Therion`, and it installs Therion and any other software it needs (Survex, Tcl, TeX) if it's not present already. On Windows you'll need to download and install the major components individually. Therion itself can be downloaded from the Therion home page [2]. Other URLs are given in the docs.

First you need to scan your notes or drawings. 200 dpi greyscale works well – see Figure 2 for an example. Scanning at higher resolution just makes unmanageably large pictures. Save these images as gifs or pnms (portable anymaps).

Entering survey data

If you haven't yet entered your survey data then you can enter it using Xtherion's built-in editor (see Figure 3). This is the same editor as "survex-edit" supplied with Survex. You can enter data just as in a text editor or you can enter it into "from" "to" etc. boxes provided. The "scan format" button will set up these boxes in the correct order from a data command entered in the main window. The editor will also fill in station names for you (attempting to predict them from the previous station).

Survex data conversion

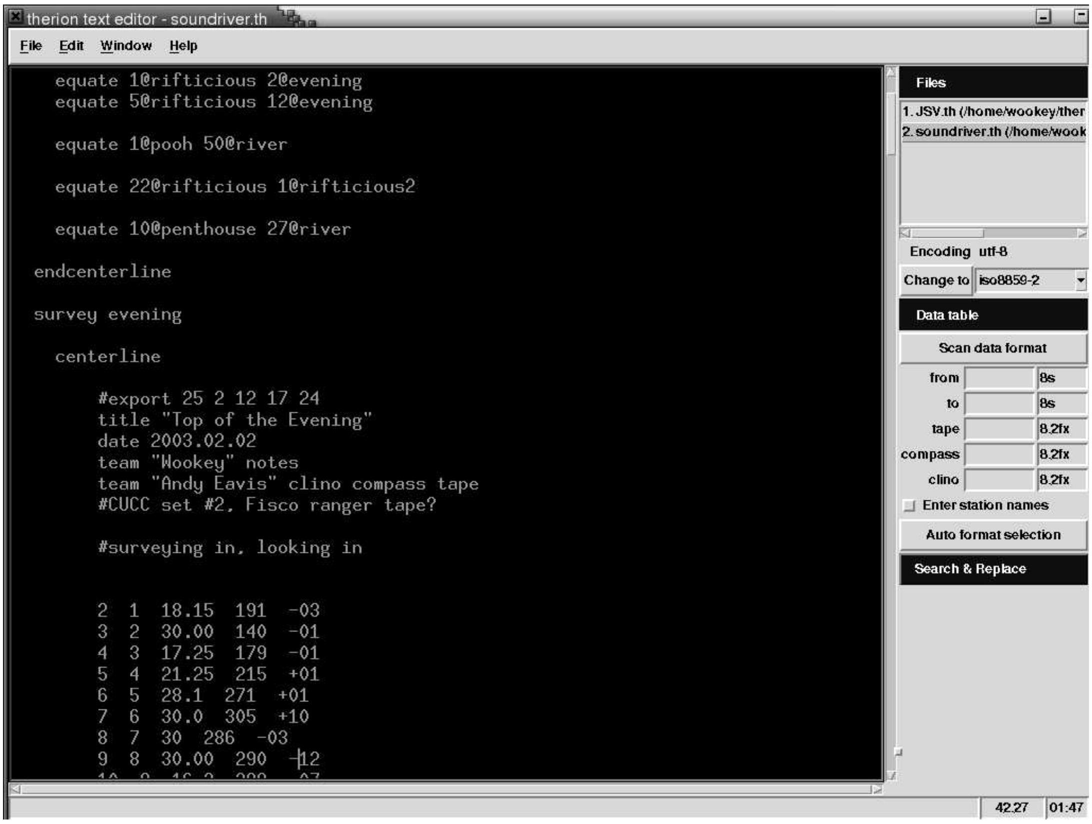
If you already have your data entered in Survex format then you need to modify it slightly for use with Therion. This is a rather tiresome feature. The Therion data format is very much like Survex's (deliberately), but it's not quite the same so you can't just use data in both Survex and Therion at the same time. Direct reading of Survex files is likely to be added quite soon to make data-sharing easier.

To convert Survex data you need to do the following:

- comment out `*export` commands;
- change the comment character from ';' to '#';
- remove the '*'s from the start of commands (i.e. `*data` -> `data`);
- change `begin` to `survey` and `end` to `endsurvey`;
- surround all survey data with `centreline`/`endcentreline` commands;
- (optionally) move `*title` from inside the `centreline` commands to an option on the `survey` command;
- put all equates inside `survey`/`endsurvey` and `centreline`/`endcentreline` pairs and change their syntax..

Therion uses @ to refer to the data item at the bottom of the hierarchy (e.g. a station or a scrap) and the order is reversed so `cobweb.soundriver.34` in Survex becomes `34@soundriver.cobweb` in Therion. And finally Survex currently allows anything as the role in a `*team` statement, whereas Therion checks for a set of valid roles.

Table 1 shows the same data in both Survex and Therion formats to illustrate the differences between them.



```
    equate 1@rifticious 2@evening
    equate 5@rifticious 12@evening

    equate 1@pooh 50@river

    equate 22@rifticious 1@rifticious2

    equate 10@penthouse 27@river

  endcenterline

  survey evening

    centerline

        #export 25 2 12 17 24
        title "Top of the Evening"
        date 2003.02.02
        team "Wookey" notes
        team "Andy Eavis" clino compass tape
        #CUCC set #2, Fisco ranger tape?

        #surveying in, looking in


        2  1  18.15  191  -03
        3  2  30.00  140  -01
        4  3  17.25  179  -01
        5  4  21.25  215  +01
        6  5  28.1  271  +01
        7  6  30.0  305  +10
        8  7  30  286  -03
        9  8  30.00  290  -12
```

Figure 3: The Xtherion editor.

Survex data	*Therion data*
  *begin evening *export 25 *export 2 12 17 24 *title "Top of the Evening" *date 2003.02.02 *team "Wookey" notes *team "Andy Eavis" Insts tape *calibrate declination 0.511 ;CUCC set #2, Fisco ranger tape? ;surveying in, looking in 2 1 18.15 191 -03 3 2 30.00 140 -01 *end evening *equate evening.17 river.53 *equate evening.24 river.1 *begin river	 encoding utf-8 survey soundriver centerline equate 17@evening 53@river equate 24@evening 1@river endcenterline survey evening -title "Top of the Evening" centerline #export 25 2 12 17 24 date 2003.02.02 team "Wookey" notes pics team "Andy Eavis" clino compass tape #CUCC set #2, Fisco ranger tape? #surveying in, looking in 2 1 18.15 191 -03 3 2 30.00 140 -01 endcenterline endsurvey evening survey river

Table 1: Comparison of Survex and Therion data formats.

Therion data files

At this point we need to consider how Therion's data files are arranged. Therion actually has two types of files - config files and data files. They look quite similar but config files contain information about the data, such as which parts should appear in the final drawing, which data files should be read and so on - we'll come to this in more detail later. The data files contain the actual data.

All these files can be called anything you like but normally there is just one config file for a particular cave, and by default it is called thconfig. If you just run the `therion` command it will automatically use this config file. To use a different one, specify it on the command line.

The data is conventionally split into .th files for the centreline data, and .th2 files for the drawing information. Xtherion selects these extensions by default in the text editor and map editor respectively so it's easier to use them than not, but remember that they do not have different formats - anything in a .th file can go in a .th2 file and vice-versa.

Almost any data arrangement in these files is possible but the common convention is to have one of each for each survey. So in this case we will save the above data as soundriver.th and we'll save the drawing data as soundriver.th2.

Xtherion use

Now I need to describe how to use Xtherion in general to save mentioning tedious details continually in the rest of the text. Xtherion has 3 main screens - the data editor, the map editor and the compiler. These are selected with F1, F2, and F3 respectively, or using the “Window” menu. F9 is like F3 but also runs the compiler “therion” on the data.

Figure 4 shows an Xtherion window containing a drawn up scrap (the drawing process is described in the next section). There is a main screen area and a panel down the right hand side that contains a set of headings in blue. Each of these headings can be expanded or collapsed by clicking on it. Sometimes there are key shortcuts to get to them. Especially in the map editor there are a lot of these headings and when expanded they all take up much more space than there is on the screen so it's usually best just to keep open the ones you are using a lot.

We'll use *italics* to refer to these headings in the description below.

The text editor has been described above and is quite simple. The compiler is also fairly straightforward; it is the map editor where most of the interesting stuff lies and where you will spend most of your time.

Starting to draw

OK, so now we get to the interesting part - drawing the cave. Switch to the Xtherion map screen (F2), and create a “New File”. It's a good idea to save this before loading any images so that loaded images will have pathnames relative to the .th file, so do that, and give it a name. Now load your first scan using *Background Images*. Click “insert” and use the browse window to select your first scan. The image can be scaled and zoomed with *Drawing Area*. It's normally easiest to just click “auto adjust” which will centre and scale it appropriately.

You can move the picture around by right-button dragging it.

Start by creating a scrap: “Insert Scrap” should be the default in *File Commands* - if not change it with the “Action” button. Click on “Insert scrap”. The top window in *File Commands* shows a summary of the .th2 data file content:

```
Scrap – scrap1
endscrap
```

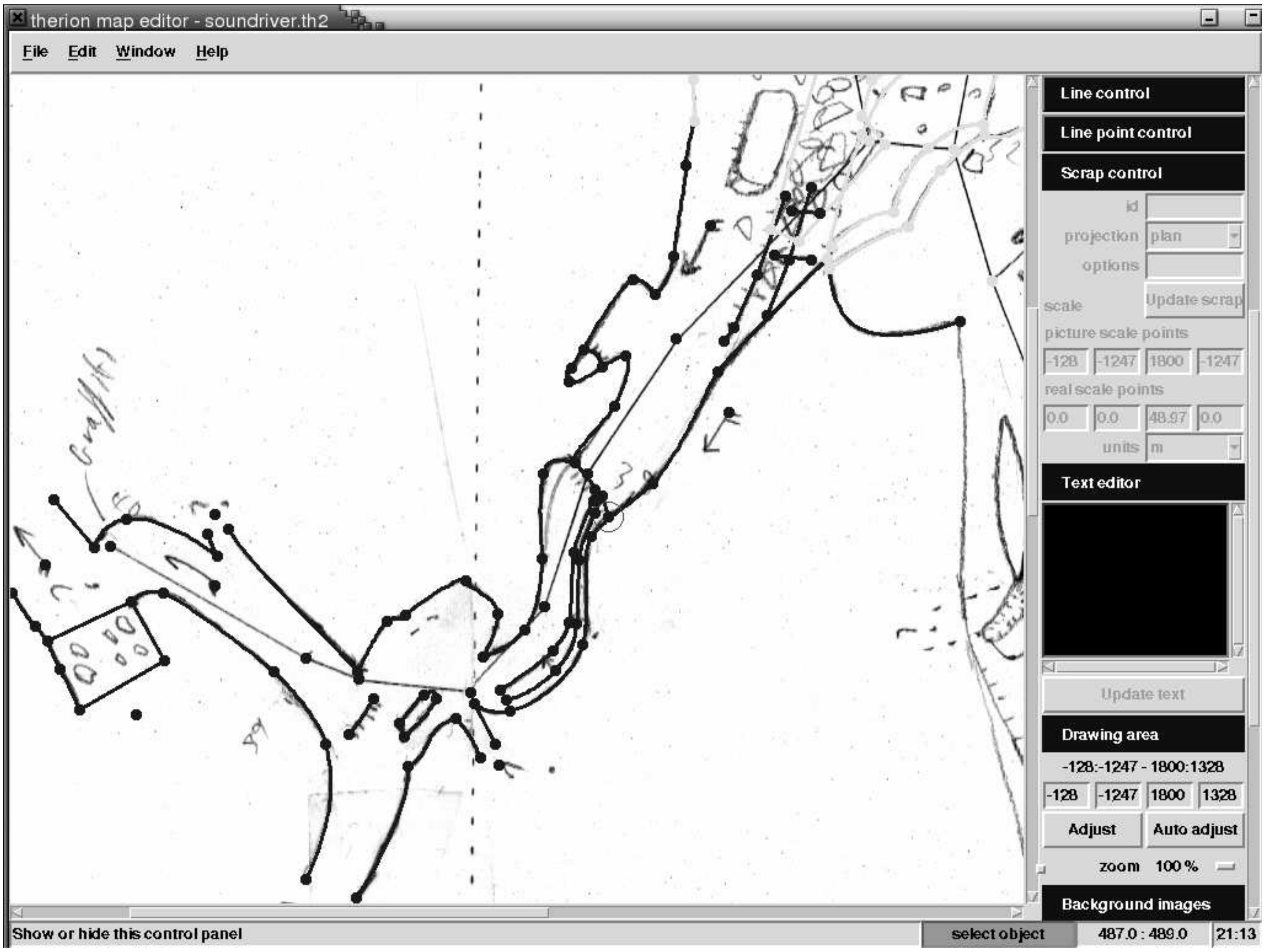
appears in the window. Every scrap has an id, and this one has been set to “scrap1” by default. Scraps need to have ids. just like stations have numbers so that they can be referred to in other commands. Each scrap normally has a projection allocated - plan, elevation, extended or none. Set the scrap to the appropriate type (“plan” for bits of plan, “none” for cross sections). Scraps of the same projection type will be brought together later.

You should remember, whilst using the map editor, that what you are actually doing is creating a text file full of `scrap`/`endscrap` commands containing lots of line, point and area commands. The map editor only creates a subset of all the Therion commands and you will sometimes need to edit the file after entering the basic drawing data to get a complete map. Understanding how the drawing data is laid out in the .th2 data file is helpful if you need to modify it later.

New data will be entered above the underlined item in this list, and you can move items up and down using the “move up” and “move down” buttons, which move the currently-selected item - illustrated by a grey highlight bar. When you select an item graphically this bar shows you which item is selected.

Now we have a basic structure we can start entering detail. Start with some lines - change the action button to “insert line” by selecting from the “Action” menu again, then click on that (or use ctrl-L) to start entering a line.

Line entry is like many other vector drawing programs. Points are either simple ones with straight lines between them, or Bezier curves with control points. To get a straight line just click when adding the next point; to get a curve click-and-drag, with the drag controlling the position of the control points. The Bezier curves



id
projection: plan
options
scale
picture scale points: -128 -1247 1800 -1247
real scale points: 0.0 0.0 48.97 0.0
units: m
-128:-1247 - 1800:1328
-128 -1247 1800 1328
zoom 100 %
Show or hide this control panel
select object
487.0 : 489.0
21:13

Figure 4: The Xtherion window containing a drawn-up scrap.

mean that you don't need many control points to accurately represent a section of wall - putting them at the ends, sharp corners and in the centre of each curve is generally sufficient. There are a number of shortcuts and shift- and control- click options for drawing conveniently. I won't cover them all here, but they are in The Therion Book, and you quickly get the hang of drawing lines.

Each line can be controlled in the *Line Control* and *Line Point Control* panels. Each line has a type - wall, pitch (pit), contour, border, rock-edge, ceiling step etc., which is set in *Line Control* - which is also the place where you reverse a line (this is necessary for things such as getting the ticks on the right side of a line). One of the less obvious things is that rock-border is used for the outline of large boulders, whilst rock-edge is used for internal lines on such boulders.

Once you've entered the walls, pitch lines and contours, you can move on to the points. This covers stations, draught indicators, question marks, heights, sinks and so on. Every scrap must have at least 1 station so that Therion can position it and either a second station or a scale so that it can be scaled and/or rotated.

Ctrl-P enters a point and *point control* allows the entering of details such as point type, id, orientation and other options. For each station you need to specify the station name so that Therion knows which station in the centreline data it corresponds to. Enter this in the “options” box as “-name 34” for station 34. You can avoid typing this repeatedly by entering all the stations in one go – leave the cursor in the “options” box, and then you just click to add each station and type in the relevant number. Some symbols need an orientation (water and draught arrows for example). Once this box is ticked arrows appear on the point which can be dragged to the correct orientation. All point symbols appear as a simple blob in the editor - the correct symbol is inserted later when Therion processes the data.

You'll quickly find that it's a good idea to enter all the lines of points of a given type at the same type as it's quicker and less error-prone. So do all the rock-borders, all the rock-edges, all the plain walls, all the QMs, all the contours etc. together.

The end result is a complete scrap as illustrated in Figure 4.

Preparing a thconfig file

This file controls the way Therion processes data if no other config file is specified on the Therion command line. It specifies which files to use, what results are required (.3d files, plans, elevations, atlases) and which surveys are to be used (if none are specified then every survey is used).

A straightforward file will look like this:

```
encoding utf-8
source soundriver.th
export model -fmt survex
export map -proj plan -o soundriverpln.pdf
```

If you want an atlas as well as a single sheet then add this line:

```
export atlas -proj plan
     -o soundriveratlas.pdf
```

Note that the default scale is 1:200. Add a base-scale command here to set a suitable scale for your map, like this:

```
export map -proj plan -layout-base-scale
1 500 -o soundriverpln.pdf
```

If you look up the layout command you'll find there are *lots* of options available to control what appears in the title bar, the symbol-set in use, where things go on the page and so on. If entering just a few layout options they can go in the export command with the layout- prefix, as I have done above for the base-scale option, but for more complex layouts its best to just specify a layout id in the export command, and then put all the options in a separate layout command. For the most complex layouts Therion allows you to write your own macros to do whatever you want. The Therion Book has a chapter describing how to do this using the TeX typesetting language.

Creating a map

The final part of the jigsaw is a map command specifying which scraps are included in it. This would look something like:

```
map   elevator   -title   "Terikan:   Elevator
Entrance"
      farside2@river
endmap
```

Map commands need to live inside a "survey" container, so in this case putting it near the start of the soundriver.th file makes sense. For larger surveys with numerous .th and .th2 data files it is usually best to have an overall "cave.th" file which lists all the files to be included (input command) and the maps to be created.

When creating maps with multiple levels and/or atlases the map command can get a lot more complex. Multiple levels are done by listing the scraps in each level separated by the text "break". The preview option controls how the small navigation panes in atlas maps are arranged.

Note on editing files whilst using Xtherion

Xtherion is good for entering primarily visual data, but when it comes to other items it can be easier to use a plain text editor. However you need to be careful if running Xtherion and editing files using an external editor at the same time, because it's easy to get your changes overwritten. You need to edit files within Xtherion, or close them before editing them elsewhere, because Xtherion saves files in the map editor automatically whenever you close them, quit the program, or run the compiler, and will thus overwrite any changes you just made in an external editor. The text editor (F1) and map editor (F2) do not co-ordinate the editing of a .th2 file, so don't try to edit a file loaded in both windows at once - it has very confusing effects. For .th2 files in the map editor I've found it best to enter all the graphical data, then close the file and do further editing in a text editor. However before you do this you need to make sure that all your areas work properly (see "Drawing Areas", below) as it's not practical to fix these outside the graphical editor. The "Open XP" option in the map editor, which opens files without the background scans, is very handy if you want to open a file quickly to check the structure, as it's the image loading that takes most of the time.

Processing the data

Finally we are ready to produce a survey. In Xtherion just hit F9 – that will switch to the compiler screen and run Therion in the current directory, displaying any log or error information in the lower half of the screen. This is exactly equivalent to running `therion` yourself in the relevant directory.

If everything goes to plan then you will get a PDF file for each `export map` command specified in the thconfig file, and a .3d file for the "export model" command. If not you'll get complaints about whatever is wrong.

Drawing areas

Areas are needed for pools, floor sediments and areas of rocks. An area is specified by a set of intersecting lines to bound each area - these lines can be visible or invisible. Some of these are often walls that you will have already drawn, but usually a couple of other lines are needed too. Each line involved in bordering an area needs an id so that it can be referred-to in an `area`/`endarea` command. The command specifies the type of fill for the area and then lists each line segment bounding the area, one per line, in the order they intersect. e.g

```
area debris
 wall1
 rbr1
 wall2
 rbr2
endarea
```

Figure 5 shows the area defined by this.

By far the easiest way to enter areas is using the *Area Control*. Click on "Insert Area" then click on the lines defining the area to insert them in the file. IDs are automatically added if not present. Note that you need to select them in order so that each intersects the next. You need to take care to ensure that the lines really do intersect. The easiest way to do this is to draw a continuous line round the area, making the last point the same as the first to "close" the loop, but this isn't always appropriate. If there are convenient existing points on the walls then you can connect lines across the passage to them (as has been done in the example above), but there often isn't a point on the wall in the right place. Lines don't need to be joined at points, but they do need to intersect. It's difficult to draw a line that exactly touches the wall, but fortunately there is usually no need to try because any line appearing outside the wall is clipped on drawing and will not appear in the final plot, so you simply extend the line beyond the wall to ensure intersection.

If you do manage to make an area that does not connect, you get cryptic errors about "paths 2 and 3 do not intersect" when you compile the data.

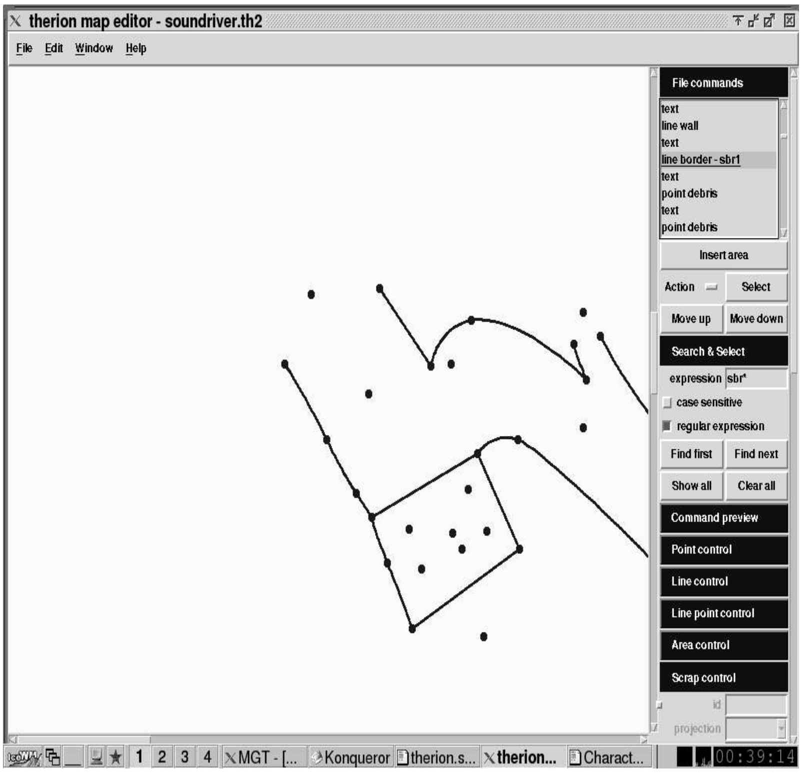


Figure 5: Defining an area: the rectangular region at the bottom has been defined as a distinct area to allow for the different floor detail (see Figure 4).

The "paths" mentioned refer to the order given in the area command - so in the above example paths 2 and 3 are rbr1 and wall2 respectively. However if you have several areas defined in a file you can't easily tell which one it is complaining about. The best way to avoid this problem is to do a quick compile each time you add a new area and check that it creates no problems

If you need to review text items in the file (such as areas) you can do this by clicking on the "text" item in *File Commands* and the content of the text will appear in *Text Editor* where it can be edited.

Joining multiple scraps

To make more scraps appear on the map, simply list them in the map command (as well as drawing them in Xtherion, of course). However this does not deal with making sure that the scraps join up smoothly. To do this you need to use the join command. The simplest form of this is:

```
join scrap1 scrap2
```

which connects the closest ends of the given scraps together. Adding -count 2 makes it look for two places to join the scraps. This works quite well for simple passage joins, but for more complicated stuff you need to specify lines or points to join. e.g.:

```
join farsidewest@river farsidewest2@river
```

connects the line "farsidewest2" (in scrap "farside2" in the "river" survey) to "farsidewest" (in scrap "farside", also in the "river" survey). Note that scrap names do not appear in these specifiers, only object ids and survey names. More explicitly, this:

```
join [farsidewest:0]@river [farsidewest2:end]
@river
```

connects point 0 (the first point) in the first line with the last point in the second line.

The join command can be used inside a scrap or inside a survey.

Figure 6 shows a close-up of a join between two scraps. However you can see that the join has not worked as we want it to – part of the farside scrap (right) is obscured by the farside2 (left) scrap. It's best to try and make the joins in areas of passage with as little detail as possible for an easy life, but to deal with this sort of problem you can add "-clip off" to any items which you need to "show through" at a join. Making those changes to the contours, ceiling step, and rock edges which were obscured leads to the version of joined scraps shown on the front cover of this issue.

If you look carefully you will see that I also fixed a number of other small problems at this stage, e.g. some of my `wall –subtype blocks` sections were "the wrong way round". All lines in computers have a way round, unlike the way we normally think of lines, so you need to draw them the "right" way round to get things like contour ticks on the down side, pitch ticks on the pitch side, wall block indents going into the wall, not out, and so on. Fortunately it's very easy to reverse a line using the "reverse" button in the Line Control panel. It was also at this stage that I set the -layout-base-scale correctly, so now the text, and wind arrows are the right size, and the line weights have increased.

How does Therion work?

Figure 7 (overleaf) is a diagram showing how the parts of Therion fit together.

Everything is controlled by the "therion" program which calls the other programs as necessary. It starts with the specified thconfig file, loading the datafiles given in there and processing them. The centreline data is passed to Survex to process to get correct station positions, then this position information is combined with the scrap data which describes the survey, Therion then scales and rotates the scraps and generates MetaPost code. MetaPost processes this to generate all the lines and symbols and thus effectively draws the survey. TeX is called to typeset any text. Each scrap becomes a few EPS files which therion turns into almost-PDF which is processed by pdftex to generate the final map. In practice all this is rather complicated, but you don't need to understand it to use the software.

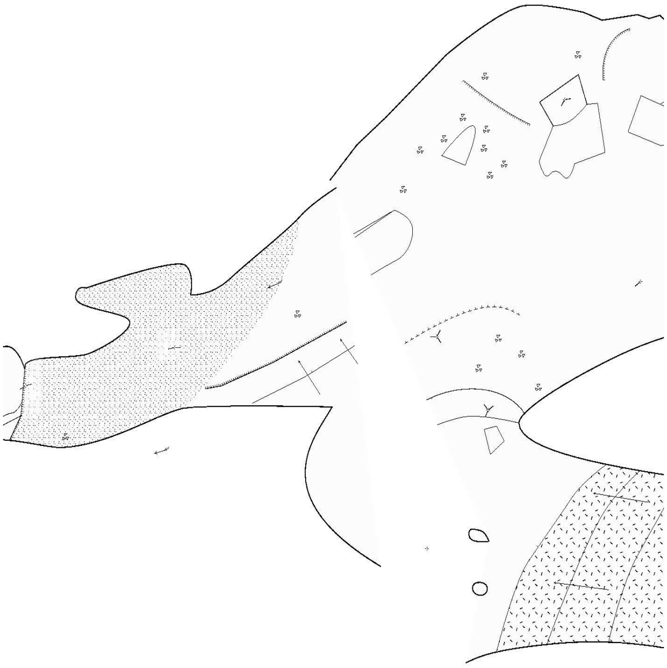


Figure 6: View showing the join between the two scraps farside and farside2

Adding your own Symbols

If you don't like the standard symbol sets supplied with Therion you can change them – UIS, ASF (Australian), CCNP (Carlsbad Caverns), and SKBB (Martin and Stacho's Slovak Club) sysmbols sets are included so far. The Therion Book explains how to add new symbols and define a new set. However this does require you to learn MetaPost which is not entirely trivial.

Tips and Tricks

Here are a few things which caught me out whilst learning to use this software, but which don't really fit within the rest of the text.

- Rock-border/Rock-edge
- The Search and Select control, which I didn't notice until it was pointed out, is very useful for finding things – e.g. all stations, or all rock-edge type lines, or an area, or a named item, e.g. if you enter "station" and click "Show All", then all the stations will be highlighted in red. This is very useful for checking that things are recorded as the correct line type. Currently if you want to do complicated searches you need to know how to write regular expressions, like the complex searches in Aven. e.g `rock*` will find both rock-edge and rock-border lines. `rock*|contour` will find those and contours too.

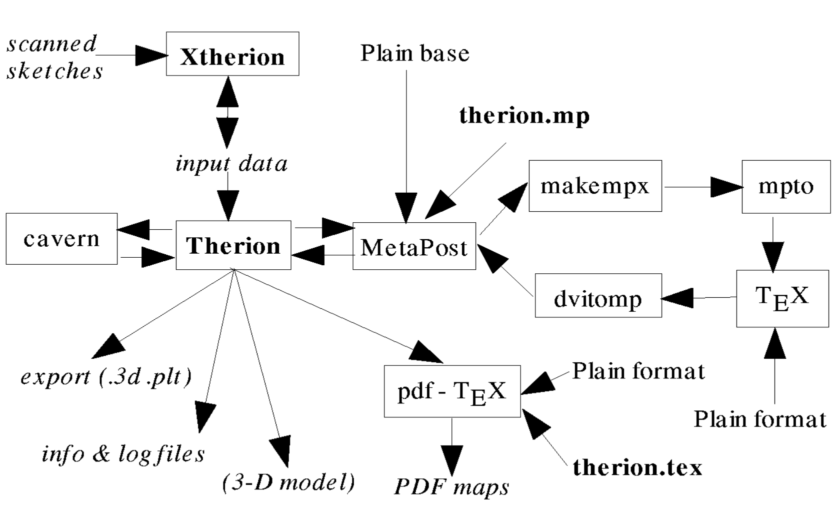


Figure 7: Diagram of how the components of Therion fit together. Programs are in boxes, other items are data files, arrows show data flow between programs. Files/programs related to Therion are shown in bold, input files created by the user and output files generated by Therion are in italics.

- There is both a scale and a base-scale command, and they interact in a flexible way, but you need to understand what each does to get the desired effect. Base-scale describes the scale at which the original drawings were done (approximately), and thus determines the width of the lines and sizes of the text relative to the overall drawing. Scale is the optical scale at which the final output is produced. You can think of it as the scale of the “final photocopy” with a real drawing, which of course often isn't the same as the scale printed in the legend or title block. With these two options it is possible to get any desired combination of “original size”, and thus line weight, and “final printed size” that you desire.

Conclusions

Therion does particularly well on cost in comparison to the alternatives. Illustrator and the like cost around $/€300-400, or £200-300, per computer. Therion costs the price of the distribution medium (CD or download) and you can put it on as many computers as you like. Now of course a lot of cavers don't pay for their software anyway, but it's satisfying to do it without breaking any regulations or going against the author's wishes.

Therion also wins on portability. Illustrator, Freehand and Coreldraw are all only available on Windows and MacOS (although the latest Coreldraw (12) is not available for the Mac). Therion is available for all major platforms (Windows, GNU/Linux, MacOSX), and could be used on minor ones too if anyone can be bothered to make it work.

On the cave-drawing front things are closer. Therion is more flexible in its output forms and doesn't have various limitations that the drawing packages do, because it is designed for the task at hand. But the interface is less slick, and this, combined with the need to specify a full logical model of scraps as opposed to just drawing a picture, makes Therion more complex to use and thus slower. Where it really comes into its own is when you need to add the next year's data to the survey, or a massive data cock-up is discovered. With Therion you just add the data, or fix the mistake. With the others you may need to draw large chunks again as well as adding data. For quickly drawing up a small survey Therion may well be overkill.

On the other hand, Therion also helps if you need to make a set of surveys use consistent symbols (e.g for a book or website). You just run all the data through again with the new symbols and fonts, so in this case using it for every cave, no matter how small, makes sense.

I'm very impressed with Therion's underlying model - it seems to me to be the right answer to the problem of drawing caves. I've resisted drawing caves on computer for several years because I didn't see sufficient advantage in it. I was also very wary of expending a great deal of time and effort putting a lot of data into something that was only a partial solution, and I might never be able to get that data out again in a useful form, so I'd end up doing it all again a few years later. However I'm now drawing a couple of quite significant surveys in Therion because I think it's a good answer, and I won't get my data stuck in a drawing package I can't extract it from, because the data format is entirely open. Computer drawing is still a lot slower than paper drawing, but the lettering is much faster, and when you need to do updates huge amounts of redrawing can be saved.

In the original draft of this article I complained that the area interface was tedious and error prone. Within a couple of days of seeing the article (and supplying some helpful comments and corrections) the authors wrote the *Area Control* to solve the problems I found. You don't get that with Coreldraw! I can only commend them on their eagerness to take account of user feedback, although the response won't always be that fast. Promised developments include more keyboard shortcuts, and options for different or improved distortion algorithms. Using Therion is satisfying because you can get involved yourself in how the software is developed: adding new symbols, updating the documentation, or giving feedback on how the user interface can maximise drawing efficiency.

The actual process of drawing caves still has plenty of room for improvement; some aspects of the interface are tedious, and there is a steep learning curve to getting going with it. However this improvement is happening quickly with a new version out every couple of months, and I hope this article has made a significant dent in the difficulties of getting started, by explaining how the parts fit together, and what you need to do to use the software to create a straightforward survey. There is plenty more to learn as you get onto more complex surveys, and when I've finished this one I'll perhaps write some more about what I've discovered on the more complex uses.

Therion is still developing rapidly. In many ways it's currently like Survex was a few years ago: the underlying software was great but it was hard to install for the average Windows user, the documentation wasn't perfect and understanding how to get the best out of it required a fair amount of effort. If that doesn't put you off then I suggest you take a look now. If it does, then give it a year or two and come back to see how things have moved on.

References

[1] Ganter, J., 200?. <TITLE OF ARTICLE>, *Compass and Tape*, 49, ??-??

[2] Therion Home Page, online at: `http://therion.speleo.sk/`

Acknowledgements

Thanks to Stacho Mudrák and Martin Budaj for technical review of this document and much help understanding Therion. Copyright Wookey 2004, licensed under the GNU Free Documentation Licence.