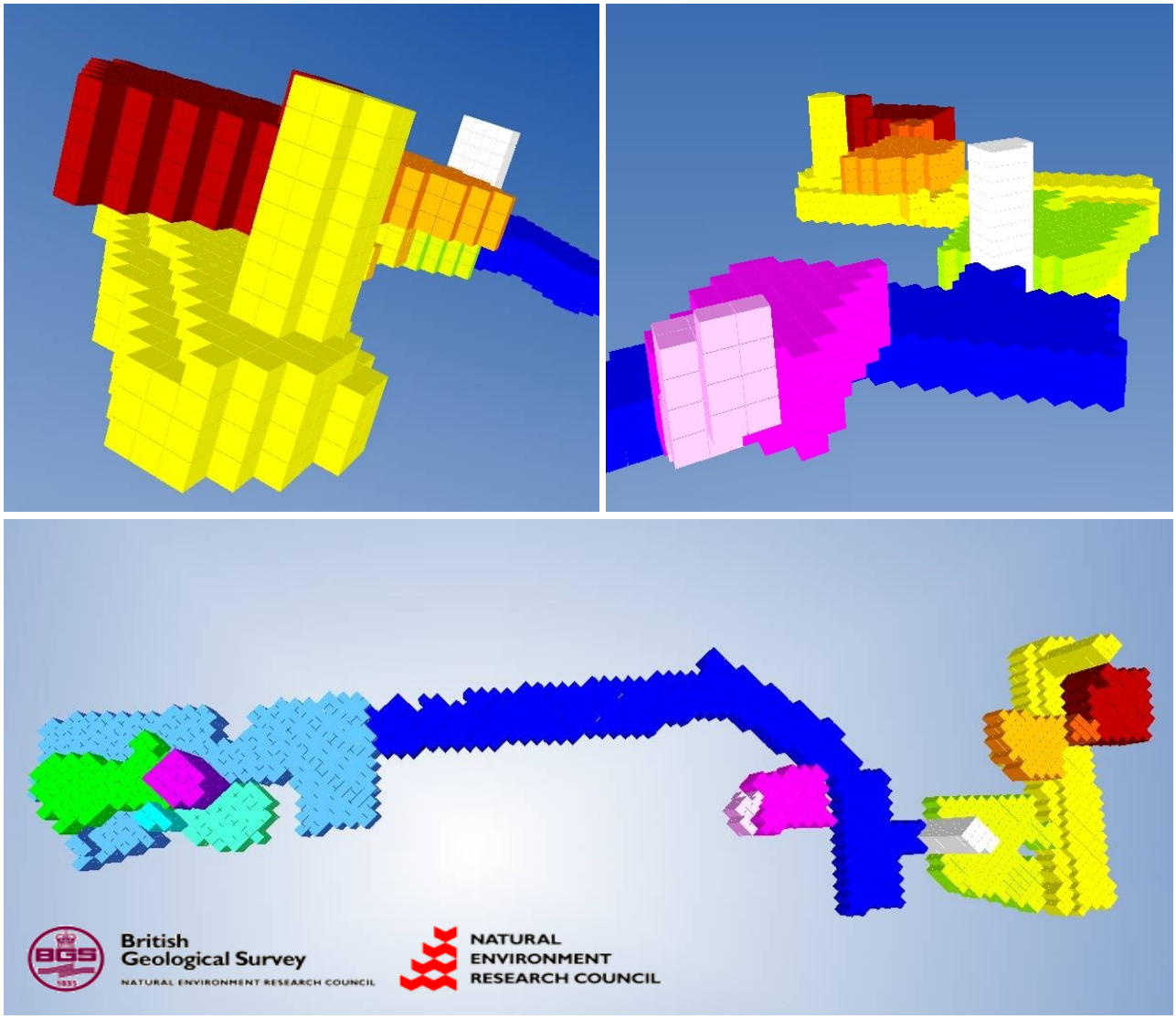
ISSN: 1361-8962

Issue 35

COMPASS POINTS

March 2006

BCRA



Storing and viewing 3-D cave survey data

Cave survey symbol font proposal

Drawing surveys with Tunnel

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, N-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 35

The journal of the BCRA Cave Surveying Group

- **Editorial.....2**
- **Snippets.....2**
  Electronic instrument projects
  Ian Todd and Mike McCombe
  Surveying software updates
  Cave survey data archiving project: survey symbol fonts
  Harry Pearman
- **Developing a data structure framework for three-dimensional cave geo-representation.....5**
  Thomas Patterson
  The BGS is in possession of large quantities of 3-D co-ordinate data for the man-made caves beneath the city of Nottingham, but much of it remains filed and unused in paper form. This article describes an effort to make these data more easily accessible in an electronic format.
- **Drawing up cave surveys by computer: the Tunnel software suite.....11**
  Dave Loeffler
  Tunnel is a dedicated drawing package for creating cave surveys. Here, Dave describes the capabilities of the software and provides a step-by-step tutorial.

Cover image: VRML visualisation of the St James Street caves, Nottingham.

Editorial

This issue is so jam-packed with goodies that I scarcely have room for an editorial. I shall therefore restrict myself to a reminder about the forthcoming field meet on 18-19 March at the SWCC headquarters, Penwyllt – contact Allan Richardson for details (Tel: 01772 783194 eves, email: allanr@caving.demon.co.uk).

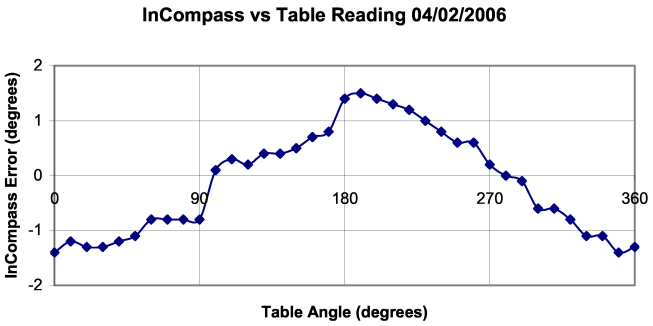
Snippets

Electronic instrument projects

Ian Todd and Mike McCombe

In issues 33 and 34 of Compass Points, Dave Edwards and Mike McCombe described their respective projects to build electronic surveying instruments.

Dave reported that South Wales Caving Club had developed a working clinometer which was in regular use, and had made progress towards building a working compass. The combined prototype unit is called "InCompass". Since that time, Ian Todd has been struggling with problems of non-orthogonality of the X/Y sensors and how to compensate for this in calibration procedures, and also the effects of soft iron. Progress is being made, and the results of the latest test are shown opposite. The compass was placed on a rotating table, and the figure shows the difference between the compass reading and the table. The single cycle error of ±1.5º suggests that the calibration routine is not quite right yet.



InCompass calibration test

Mike's instrument was also in a similar state - the clinometer but there were problems with compass calibration. He reports that there are still a few things on the "to do" list but the instrument is substantially complete and has seen action in Ogof Ffynnon Ddu and Dan yr Ogof. As intended, it really comes into its own in tight, wet, awkward passage where it would be very difficult to sight with a conventional compass and clinometer. Mike's usual survey partners are delighted with the reduction in angst and shivering that it provides, reportedly referring to it as "the dog's bollocks!"

So far, they have not measured many underground closed loops, but during surface trials were typically obtaining closure errors around 2%. As expected, the clinometer accuracy is superior to that of the compass. Overall, the results are not as good in "easy" passage as can be obtained with conventional Suunto compass/clinometer. It is noticeable, though, that the nature of the errors is different. Usually, with conventional instruments, a reasonably good set of observations are obtained with the occasional wildly wrong "blunder" caused by human error - reading the wrong scale on the clinometer, getting the sign wrong or holding the compass so far off level that it sticks. With the digital instrument, the error distribution seems to be more "Gaussian". In addition, since taking readings is so quick, each shot can be recorded several times to check for consistency. Arguably, then, the digital instrument is a better match for least-squares loop closure methods used by most cave surveying software.

Surveying software updates

Here is a round-up of some recent developments in the world of cave surveying software.

First up **Compass** [1]. The latest version includes a number of new features, some of which are highlighted here (see the website for a full list of changes). The Viewer now supports the option of locking bitmap images to the cave. This is useful for tying topographic maps, aerial photographs and other maps or drawings to the cave. Once the image has been registered, the image will expand, contract, shift and rotate in synchrony with cave. CaveX can now generate 3-D Rose-Depth diagrams that can be panned, rotated and zoomed, which allows them to be viewed from any angle. 3-D Rose Diagrams are much more useful than ordinary Rose Diagrams because they allow you to see passage trends that only exist at certain depths in the cave. On the processing side, several improvements have been made to the Blunder detection routines. These include displaying the original Azimuth, Length and Inclination with each blundered shot candidate, thus making it easier to select the likely blunder.

The latest stable version of **Survex** [2] is 1.0.39, and **Winkarst** [3] is now at version 12.3. Recent updates to both packages have been mainly minor enhancements and bug fixes. The latest build of **Walls** [4] includes updates to the SVG-related features to support the latest version of Adobe Illustrator.

A "new" (two years old) surveying package – **Speleoliti** [5] has been brought to my attention. This is Windows program for spatial modelling of caves, and includes a special module for easily constructing extended profiles. A new version has recently been released, and it is available in Slovenian and English.

In the world of drawing packages, **Therion** [6] has seen several updates and is now at version 0.3.9. New features include support for inputting passage dimension information as LRUD measurements. The current capabilities of **Tunnel** [7] are described at length in an article by Dave Loeffler elsewhere in this issue.

[1] Compass: `http://www.fountainware.com/compass/`
[2] Survex: `http://www.survex.com/`
[3] Winkarst: `http://www.resurgentsoftware.com/winkarst.html`
[4] Walls: `http://www.utexas.edu/tmm/sponsored_sites/tss/Walls/index.html`
[5] Speleoliti: `http://www.speleo.net/speleoliti/`
[6] Therion: `http://therion.speleo.sk/`
[7] Tunnel: `http://www.goatchurch.org.uk/tunnelx/index.html`

Cave survey data archiving project: survey symbol fonts

Harry Pearman

Compass Points 34 included an article making the case for the establishment of a national cave survey data archive. Following these proposals, Andrew Atkinson has set up an experimental web site to explore the ideas further. This reproduces for permanent reference much of the original material and, in addition, Andrew has refined the original proposal. The website can be found at:

`http://seagrass.goatchurch.org.uk/~andrew/Home`

Another part of the project is to investigate a new approach to cave survey symbols. Cave survey symbols or logos are are very useful shorthand in conveying survey information. Part of the data archiving proposal is that, instead of a one-size-fits-all approach to publishing cave surveys as at present, in future, survey plots should be tailored to the needs of specialist users. By inference this means that instead of a single standard set of cave survey symbols there should in be several of them.

There are two options for plotting symbols as part of downloaded survey data. One is to download the symbols to plotting programs as graphics so that they become part of the plotting process; the other is for the printing PC to hold symbol libraries and to select and print symbols at appropriate places on surveys in response to embedded instructions in the downloaded data. Both of these carry development and maintenance overheads, to say nothing of extensive re-writing of plotting programs.

There is a third option which simplifies the whole process and this is offered as a solution. This is to devise cave symbol fonts. The symbols are then transmitted as text in the same way as notation data similar to, say, cave passage names. The plotting PC only has to hold the fonts in its font library and to use the appropriate one according to survey type.

With this in mind some draft work has been carried out on designing a set of cave symbol fonts and the results are shown overleaf. These are drafts and anyone with comments should send them to Harry Pearman at `hape@compuserve.com`. Harry will also supply the symbol fonts themselves on request.

Mines and mining activities symbol font (CAVE-M)

Keyboard Symbol	***ASCII Decimal Code***	***Symbol***	***Description***
q	113		Open shaft - descending
w	119		Shaft with removable cover - descending
e	101		Closed shaft including blocked and capped shafts - descending
r	114		Open adit or drift entrance
t	116		Closed adit or drift entrance which can be opened.
y	121		Permanently closed adit or drift entrance
u	117		Boundary of walled section
i	105		Boundary of traversable collapsed area
o	111		Blocked passage
p	112		Boundary of run of roof props
a	97		Unsurveyed extension
s	115		Unexplored extension
d	100		Significant graffiti or carvings
f	102		Artefact(s)
g	103		Work face
h	104		Borehole in roof
j	106		Borehole in floor
k	107		Borehole in roof and floor
l	108		Borehole in wall
z	122		Open shaft - ascending
x	120		Shaft with removable cover - ascending
c	99		Closed shaft including blocked and capped shafts - ascending
v	118		Open shaft – ascending to surface

Cave exploration symbol font (CAVE-E)

Keyboard Symbol	***ASCII Decimal Code***	***Symbol***	***Description***
q	113		Too tight
w	119		Horizontal squeeze
e	101		Vertical squeeze
r	114		Climb Up
t	116		Duck
i	105		Environmental hazard
o	111		Telephone point
p	112		Log book
a	97		Lock
s	115		Bivouac
d	100		Loose rock
f	102		Fixed ladder
g	103		Traverse wire
h	104		Belay point
j	106		Dig
k	107		Accident black spot
l	108		Emergency dump
z	122		Traversable sump
x	120		Thick mud
c	99		Liable to flood
v	118		Bad air
b	98		Climb down

Cave geology symbol font (CAVE-G)

Keyboard Symbol	***ASCII Decimal Code***	***Symbol***	***Description***
q	113	D3	Dibunophyllum D3 base of millstone grit
w	119	D2	Upper Dibunophyllum D2
e	101	D1	Lower Dibunophyllum D1
r	114	S2	Upper Seminula S2
t	116	S1	Lower Seminula S1
y	121	C2	Upper Syringathyris C2

Keyboard Symbol	*ASCII Decimal Code*	*Symbol*	*Description*
u	117	C1	Lower Syringathyris C1
i	105	Z2	Upper Zaphrentis Z2
o	111	Z1	Lower Zaphrentis Z1
p	112	K2	Upper Cleistopora K2
a	97	K1	Lower Cleistopora K1
s	115	Km	Cleistopora Km passage beds to Old Red Sandstone

Cave hydrology symbol font (CAVE-H)

Keyboard Symbol	*ASCII Decimal Code*	*Symbol*	*Description*
q	113		Running water 090 degrees
w	119		Running water 270 degrees
e	101		Running water 000 degrees
r	114		Running water 180 degrees
t	116		Water exits 090 degrees
y	121		Water exits 270 degrees
u	117		Water exits 000 degrees
i	105		Water exits180 degrees
o	111		Water enters 090 degrees
p	112		Water enters 270 degrees
a	97		Water enters 000 degrees
s	115		Water enters 180 degrees
d	100		Sound of water
f	102		Heavy drip
g	103		Sump

Developing a data structure framework for three-dimensional cave geo-representation

Thomas Patterson

This article describes an effort to develop procedures for storing and visualising geographical information about caves, with particular reference to data from the man-made caves of Nottingham held by the British Geological Survey (BGS). Since the 1970s it has been an ongoing project for the BGS to record, survey and map each subterranean find. In recent years, this information has been stored in digital form within a 2D GIS shapefile and database query system. Despite these efforts, however, much information in the form of surveys (containing 3D coordinate data for a number of caves) still remains filed and unused in paper form. Nevertheless, a variety of interested parties would prefer to see this information stored and made retrievable in a three-dimensional format, which suggests that a method for containing this valuable data must be found which is easy to create and manage, undemanding on computer power, and useful as a 3D cave product.

Introduction

Approaches to the 3-D modelling of subterranean caverns are heavily influenced by the initial state of the primary survey data. At present, ground-based LIDAR is the most commonly used method of data acquisition in the archaeological field, although non-digital survey techniques are still frequently exercised. Traditionally, the two most important approaches adopted for communicating the appearance and shape of an underground space are cave surveying and photography [1]. Although effective in allowing people to envisage these often inaccessible areas, the limited two-dimensional appearance and lack of depth permitted by these approaches can reduce realism and hinder the recognition and exploration potential of a site. These assets are essential requirements if the models are to be used within the fields of education, archaeology, geophysics or planning [2]. In recent years, computer-aided techniques have been adopted to harness the information from traditional techniques for the production of clearer, three-dimensional viewing and querying experiences, with potential for direct viewer interaction in an electronic space, through media such as VRML (Virtual Reality Modelling Language) and 3D interactive environments such as CAVE [3].

Successful visualisation of caves is heavily dependent on the quality and quantity of available spatial data. Previous efforts have predominantly incorporated either historical or accurately measured survey data, although photography is another major source of original information through photogrammetric procedures. Techniques vary greatly in the amount of computational power that is required both to produce and display models, with levels of

realism often suffering in the larger or less detailed caves [3]. However, although the principles of geo-representation have been adopted for geological applications [4], little research has been undertaken into their employment for cave data representation. This approach could be useful in a variety of applications, from geophysical analysis, to querying cave locations for urban planning implementation, and even for digital archiving, storage and modelling of survey data held by an organisation such as the British Geological Survey (BGS).

The data for this study were taken from the caves of Nottingham. The city probably has more man-made caves than any other in Britain [5]. More than 400 artificial underground spaces have currently been discovered in the Sherwood Sandstone underneath the city centre, excavated through the years with hand implements to serve many purposes, including dwellings, storage rooms, sand mines and air raid shelters [6]. Using Nottingham cave data obtained from the BGS, this paper aimed to provide a scoping study of established geo-representation data structures for the development of a model framework for the storage, retrieval, querying and visualisation of three-dimensional cave survey data. It identified the importance of data structure for the graphical representation and exchange of information across different disciplines, and considered the future implications for cave data storage and representation as a research area by reviewing and critically appraising key aspects of the construction process.

Three-Dimensional data structures

Of the myriad solid modelling solutions available, this study has used a "voxel" approach, whereby the modelled object (the cave) is broken down into volume primitives [7] - regular cubic cells called voxels [8]. The voxel approach is a space subdivision technique that is often preferred over other methods due to its capability to store characters at each point (object occupancy), and the relative complexity of its vector model counterparts [9]. Representing a cubic unit of volume centred at an integral grid point, the voxel is the 3D counterpart of the 2D raster pixel, which signifies a unit of area [10]. Each voxel can have either binary or numeric values associated with it, which can represent either some measurable properties or independent variables.

The advantages of using voxels instead of vector-based surfaces include their insensitivity to scene and object complexity, their viewpoint independence, and their ability to represent sampled and simulated datasets [11]. Disadvantages with this method include the fact that voxels store data in discrete form, causing a loss of geometric information and accuracy, and that the memory and processing power required is habitually very large relative to other approaches. However, recent developments in volumetric visualization (particularly for LIDAR representation) have reduced the memory and computing power needed to render a scene, and in fact some software can render scenes more efficiently than surface representations [12]. The voxel data structure would therefore appear to be a sensible and useful choice as an appropriate data structure for geo-representation of a static three-dimensional cave with volumetric characteristics.

Methodology: constructing the voxel array

The method proposed for the design and construction of a data structure framework suitable for storing and representing three-dimensional cave information was broken down into a number of steps. The first step was to acquire and prepare the test cave data (obtained as surveyed plans and cross-sections from the BGS, Figure 1) ready for processing into the appropriate format.

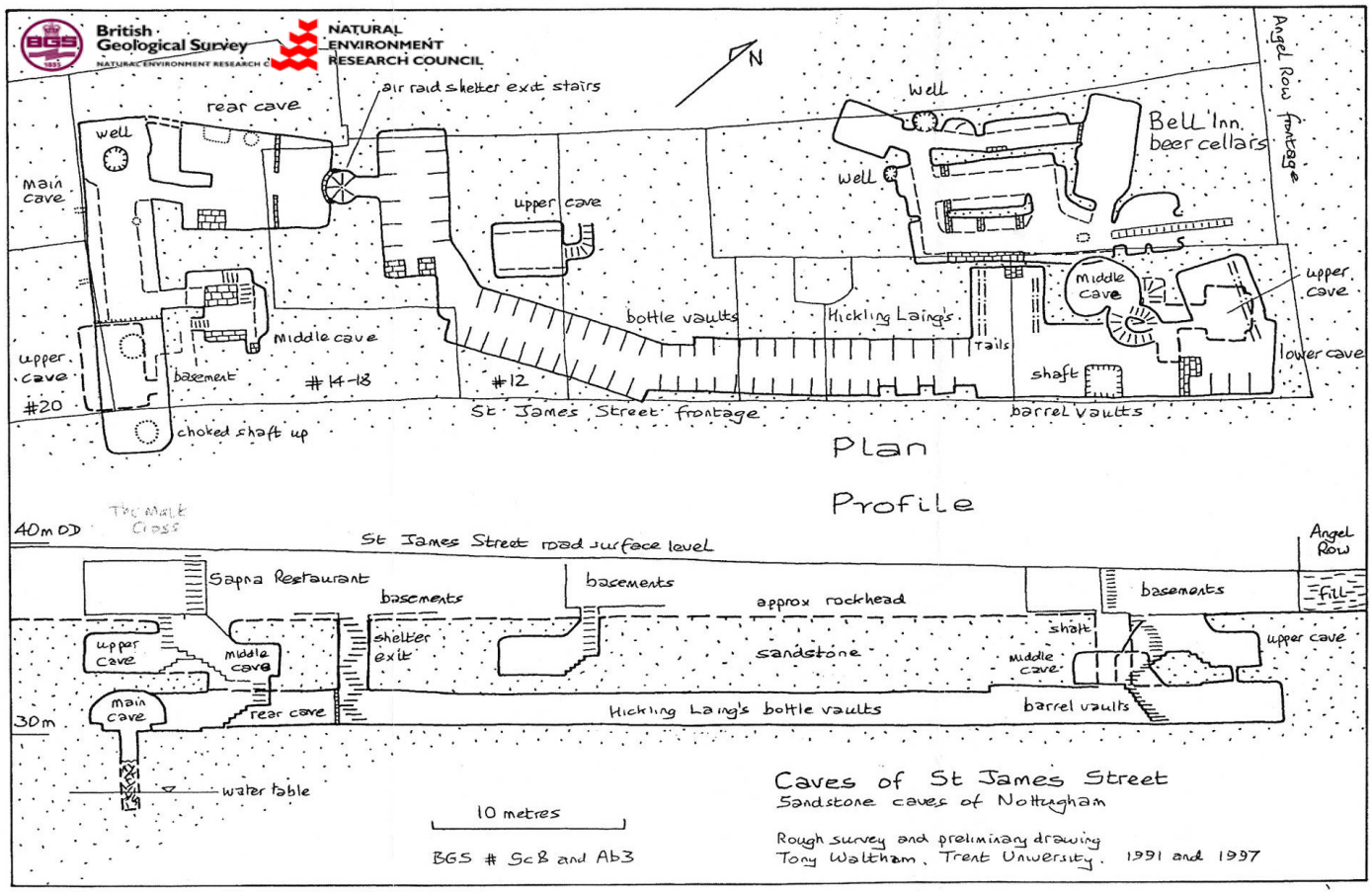


Figure 1. Primary data: an example of a rough sketch survey of St James Street Caves (by A. Waltham, 1991 and 1997).

Additional data sources such as photographic and laser-scanned information were overlooked in this study due to the lack of available data. Using Visual Basic 6.0, a program was then written which allowed all of the necessary formatting functions to be performed, including data input, creation of the 3D output array, combination of different spatial entities, as well as additional tasks including data reduction and compression, attribute linkage and visualisation. The final stage permitted the data to be displayed in its new structural form. and allowed the recommendation of data structure formats for future applications.

The first step in the approach to designing a data structure framework involved the choice of a fitting subject to model. With a limited number of cave surveys containing suitable three-dimensional information, the St James Street caves (Figure 2), close to the centre of Nottingham were chosen as detailed survey diagrams and height information were readily available in paper form (Figure 1). The survey plans were then geo-rectified into the MapInfo Professional GIS system, creating a digitized 2D plan of the cave system to allow for the unproblematic creation of input data for the eventual Voxel program.

The St James Street caves can also be divided into 15 separate rooms or "objects" (Figure 2). This allowed the 3D voxel array to be built up as each individual spatial entity is added. This dissection is also advantageous as it allows the input data for the cave system to consist of 15 separate data files, containing the ceiling and floor height (z) values for each point in a room. This permitted the final program to read each room separately, and ensured that caves, which overlapped on a 2D plane, were stored accurately into the 3D block array as different vertical levels (for example, the Barrel Vaults (area 6) and St James Street Upper Cave (area 9)).

Once each object had been saved as an individual tablet file, a 2D vector grid of points were placed over the entire cave area at a chosen resolution of 0.5m (Figure 3). Each point represented the integral centre of a 0.5m x 0.5m voxel cube, which were used to ascertain whether a voxel was positioned over the location of a cave object, and whether the voxel lay within the space between the ceiling and floor of that cavern space. Despite the relatively coarse resolution of this initial grid, the voxel approach still provided a multitude of blocks for manipulation within the structure. The 3D array at a 0.5m resolution contained 349,890 voxels (109 x 107 x 30) for the St James Street Caves, which was unquestionably enough for this preliminary appraisal of the data structure approach to be sufficiently tested and analysed.

In order for the 3D voxel group to be created, an automated program capable of first producing a blank array, and secondly filling up the data structure with appropriate object content, was produced. Using Visual Basic 6.0, *Voxel Creator* was written as a soft-coding solution for the facilitation of these tasks, and provided a structural foundation for future cave data management. The application allowed the user to create a blank array that contains the entire cave

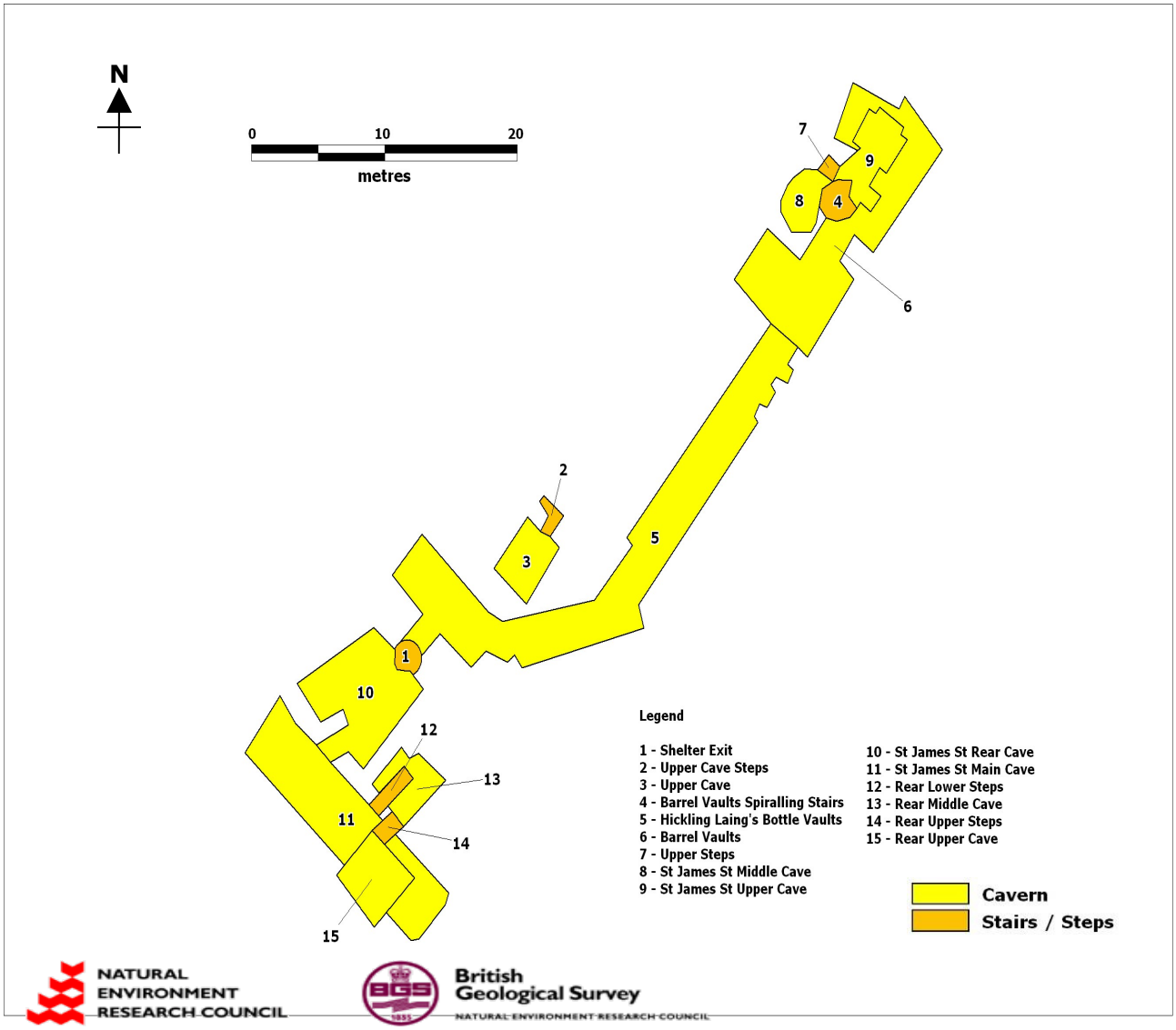


Figure 2. Digitized map of St James Street Caves.

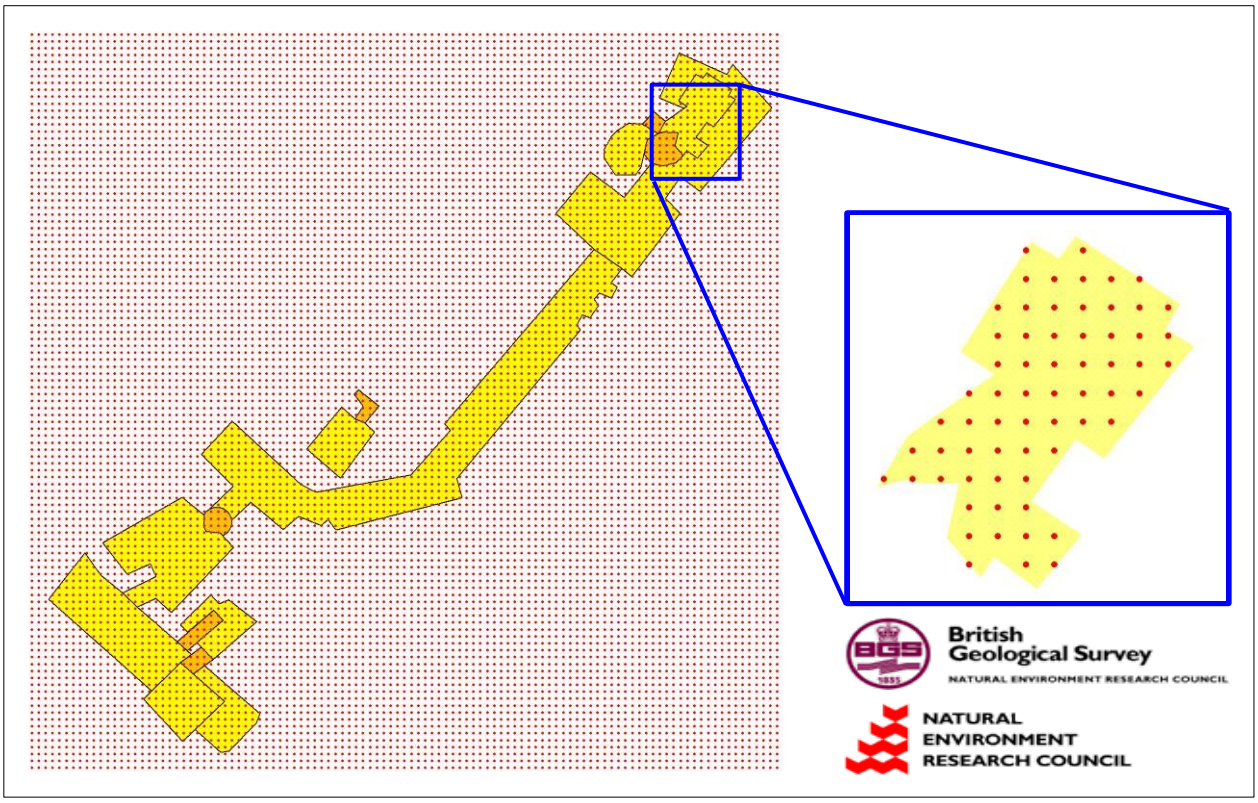


Figure 3. 0.5m Integral Grid Point Array covering St James Street Caves (left), with SQL selection of points for Location 9: St James Street Upper Cave (right).

system dimensions. Once the limits of the full 3D block are input, the program exported a text file containing a list of every voxel within that domain, ready for the encoded survey files to be input through the array. Voxel Creator also allowed the data to be visualised on a 2D plane for quick analysis and data-checking.

The main advantage of the voxel structure approach is its ability to successfully store attribute values for each unit. Based on the work of Carlbom [13], this feature can be utilised to store binary integers which describe whether a voxel is contained (1) or absent (0) from an object (in this case, the cave system). Voxel Creator takes each integral grid point from the input text files, and finds all corresponding voxel points within the blank array that lie under that point (by comparing the stored Easting and Northing coordinates). If the height (*z* value) of the voxel lies within the height range given for the particular object being read (the Maximum and Minimum Z values), then the voxel is prescribed the value of 1, otherwise it is overlooked and the default value of 0 is left unchanged.

A variety of additional options were provided to demonstrate the Boolean functionality available through the voxel structure. These included reduction of the data through the deletion of blank voxels (outside the cave system), and the identification and storage of "edge" voxels within the cave structure. These features aimed to minimize computer memory usage, as well as speed up visualisation, analysis and querying functionality.

Voxel Creator provided two different ways to view and explore the newly-created data structure. The first, and most simple approach allowed by the program is a 2D cross-sectional view of the data. Although relatively easy to program, this approach still took time to process as the voxel data had to be read for each cross-section selected, whether the chosen line ran along a grid row or column, or a user-defined diagonal slice of the array.

Nevertheless, a more interactive, 3D visualisation technique may be preferred in order to fully explore the potential of the data structure. Therefore, the program also allowed the option to export each location as a VRML (*.WRL) file. After full processing of all the locations was complete, the 15 outputted VRML files for the sample cave were combined using an *inline* file structure to display the entire cave system in an immersive 3D space which could be viewed from any angle within the virtual environment. As an open, extensible, industry-standard scene description language for 3D scenes, VRML is relatively universal and opens up a diverse potential range of possibilities and applications that the voxel-structured cave data may be used for in the future.

Reviewing the data structure

As suggested in previous research, the conversion of the BGS survey data to the voxel data structure was both relatively straightforward and successful [14]. Although processing time and memory storage were both key areas of concern, the program performed admirably and was able to combine all of the location files with few problems. A total of 5,818 voxels were created within the entire cave system, allowing the calculation of a direct volume approximation for the cavern of 727.25m^3 (a figure that could be made more accurate if the integral grid resolution was increased). The binary (0,1) method of encoding may be highlighted as a source of volume error as object occupancy within voxels may lead to under- or over-estimation of voxel numbers, determined heavily by the placement of the central voxel point grid array, but in future a higher resolution grid may once again be able to produce more accuracy.

The following section addresses the key issues and advantages of the voxel structure approach with regards to the Voxel Creator program as well as the related research.

Processing speed

As would be expected, processing time increased linearly with the number of voxels that had to be encoded. A different programming perspective could be adopted in order to reduce processing time in future. Larger caves and higher voxel resolutions would invariably result in large, unwanted periods for array creation, and therefore subsequent programming must concentrate on reducing the number of times that the entire array is read, by decreasing data cycles and perhaps extracting the necessary 2D region of integral grid points before processing.

Attribute linkage

The advantages of allowing attribute storage were clear to see during the programming and visualisation stages of the voxel methodology. Boolean operations were easily administered, allowing successful combination, reduction and visualisation of the 3D data array, and permitted characteristics within the model, such as locational independence, to occur effortlessly throughout the 3D array with initial data registration kept at a minimum. The additional storage of locational codes also allowed volume approximations to be made for each cave object. In future, additional attributes such as detailed object-relationships could be calculated and stored. Although this would increase the size of the dataset, hierarchical database structures provided by Object Oriented techniques could allow this information to be stored with minimal processing cost and allow quick retrieval and analysis.

Reducing large data volumes

Due to improvements in resolution and accuracy of acquisition devices, such as laser range and medical scanners, the data volumes associated with information structures have become problematic, particularly for visualisation and querying applications. Large volumes and complicated information can make visualisation more challenging. Therefore, multiresolution techniques may be necessary to view volume data, creating initially coarse and simplified representations that can be magnified for finer resolution. However, the main concern with the voxel approach to data structuring is the vast storage cost [14]. Voxel Creator attempted to reduce the dataset size by eliminating voxels that did not contain the cave system, or were not at the edge of a cave object. Although this is successful in reducing large files and keeping only the information that is crucially required, a better way to reduce memory cost would be to impose a structural organisation on the basic voxel labelling scheme.

The most simple reduction algorithm accessible through Voxel Creator removed voxels that did not reside within one of the 15 cave system objects. Although rudimentary, this method reduced the overall file size for the 3D array by almost 99%. This highlights the key problem with the voxel approach, which is the amount of 'wasted' or unused space within the data structure, and emphasizes the need for the reduction and compression of information.

Although having a total voxel count for the entire system was advantageous for volume measurements, further reduction could be considered as voxels within the walls of caves also take up valuable storage space. Therefore, after implementing the Edge Detection algorithm, 29% of the voxels were shown to be entirely contained within other cave voxels and were deemed to be unnecessary. This significantly reduced the storage size of the output array, and retained voxels that were required for a successful and complete visualisation of the enclosed system. Obviously, however, these results are heavily dependent on the resolution of the voxel array and the size of the cave system.

Visualising the data model

Both attempted visualisations of the 3D array proved successful for the exploration and viewing of the dataset. The main advantage provided by the chosen caves was their man-made origins, making them flat-roofed and cubic in construction. This compliments the chosen data structure and modelling outputs, which favour the artificial, more rigid form provided by such structures. However, to apply this framework on all subterranean forms, particularly to naturally-weathered caves, may yield highly inaccurate and poorly-formed models, as these objects could possess erratic and unsymmetrical shape properties.

Once again, it would be understandable to see that an increased resolution for the 3D array would have produced further realism and accuracy in the visualisations. Nevertheless, the chosen 0.5m voxel size is still adequate for enabling the general form and relationships within the cave system to be viewed without jeopardizing the processing time and memory consumption.

The simple 2D interface allowed by Voxel Creator provided clear and acceptable cross-sections of the cave system to be viewed from any chosen transect. The limited interactivity and manipulation of viewpoints was a problem, and therefore the second visualisation attempt, harnessing exported VRML files, appeared to be a far more useful product with greater application potential. The main advantage of the VRML output was that once the 3D data had been read into Voxel Creator, the VRML could be created and would not need to continually use the voxel array for cross-referencing during each new visualisation. An entirely separate file system was created by the program for each location, allowing separate objects, or the entire system (see cover image), to be viewed at any time. The VRML approach also had a large potential for development. Texture changes and lighting can add additional realism to the model, while combination of other cave models and surface datasets could in future create a large, virtual cave environment of the entire central Nottingham area, with hot-links created within the models to open additional photographic, scanned survey and historical/educational information, producing a complete data archive for reference, storage and exploration [15].

One disadvantage of both the chosen methods and of the data structure itself is that we can only, at present, observe exterior views of the cave structure [16]. The VRML shown on the cover of this issue visualises the entire voxel set in a solid form. As previously mentioned, the coarse resolution of the study dataset dramatically impedes examination of any potential 'walk-through' function, although a higher level of accuracy, and a larger, more complex array, would provide far more authentic, hollow structures which could be textured in a realistic way (perhaps even with original survey photographs). This option would also allow supplementary historical information to be stored in the 3D environment, as several of Nottingham's caves contain wall drawings and artwork, which could be embedded into the virtual space of the related cavern. More intricate formations could also be modelled in VRML using photogrammetric techniques or LIDAR-based survey equipment to provide increased realism and dimensional preservation for these artefacts.

With regards to cave modelling, a common objective would be to render complex scenes as quickly as possible, but also store a great number of geometric primitives, attaining a level of accuracy higher than the one currently undertaken in this project. A multiresolution approach would therefore be an acceptable way to achieve this, by creating a rendering program that allows the user to choose the appropriate level of detail for the object's screen size so that less time is wasted drawing insignificant detail [14]. Creation of a multiresolution model is difficult, and was well beyond the scope of this study, as laborious manual database preparation is still frequently used in modern research [17]. Multiple resolution solutions may, however, provide an answer for fast and successful visualisation in subsequent efforts at higher resolution cave modelling.

Structural organisation

Although not hugely important for the less complex models, database design is still considered an important factor in 3D solid geometric modelling for the further development of geo-representations. Successful implementation can bring advantages such as data security and integrity, as well as support for the management of model versions, multiresolution representation and query-based model decomposition. Despite this, few databases have yet been developed in the growing field of geo-representation [9]. Approaches to the design depend on the chosen data model for the database, usually either relational or object-oriented, which have both received a great deal of research in the last decade. While most of the database approaches used in solid modelling are employed for boundary representations, Kavouras [18] showed that voxel models can also be linked to associated attribute tables through a voxel

indexing scheme (such as the unique voxel ID value used in this study), suggesting that a consideration of the approach would not be wasted.

Other issues to consider

Problems with uncertain data need to be addressed when secondary data is used for voxel data construction. As the survey data was created from a primary source this issue did not arise in this study, although it may be necessary to address this matter in future work, with the potential for introducing fuzzy theories of voxel occupation and uncertainty [9]. It may also be recommended to consider the addition of time geographies into the data structure. 4D models, which handle three dimensions of space and one of time, are frequently being adopted into solid geometric modelling practice [15]. These techniques are generally used to represent forms, properties and structures of geo-phenomena and dynamic physical processes, and could well be exploited in future cave modelling frameworks for the modelling of weathering, the development of the artificial cave network over time, or for keeping an accurate record of cave information storage development.

Conclusion

This study successfully achieved its objectives by demonstrating that the voxel-based approach to data structuring and 3D solid geometric modelling can be used effectively in order to serve the multiple purposes of storage, interpretation and graphical communication of cave information. The chosen method provided unique opportunities for attribute linkage and data organisation. Proposed enhancements to the methods undertaken in this project could add numerous improvements to both performance and future use, particularly if new, more accurate data sources could be acquired, and if structural organisation within the dataset could reduce memory cost and processing time, while also providing more efficient and informative visual representations of the cave model. Although the surface of this topic area has only just been scratched, the modelling accomplishments of the preliminary model, using only digitized survey plans, should not be under-rated, with the creation of a robust and functional model providing ample proof that the data structure can perform adequately as a framework for future cave geo-representation.

In summary, the voxel system, though simple in appearance, conceals numerous complexities in the way in which solid forms are constructed, classified, rendered and displayed, and offers great potential for future data archiving and application. This approach could lead to the development of versatile cave information systems, which combine the capabilities for storage and retrieval with those of analysis, interpretation and visualisation, providing functionality that would be of immense interest across the spectrum of education, research and industrial practice.

Acknowledgements

This article is based on a dissertation entitled "*One foot in the cave: Developing a data structure framework for three-dimensional cave geo-representation*" presented for the degree of M.Sc. in GIS at the School of Geography, University of Nottingham, September 2005. The author would like to thank Dr Michael McCullagh and Dr Gary Priestnall of Nottingham University for their guidance and advice during the initial stages of this dissertation. Regards also to Jenny Walsby of the British Geological Survey for allowing full access to the Nottingham Cave Archives in Keyworth, providing, most importantly, the interesting central subject matter for this project.

References

[1] Stoddard, S. (1994). An Introduction to Cave Photography. [London: British Cave Research Association]

[2] Roe, M. (2000). The brighter the light the darker the shadows: how we perceive and represent underground spaces, Capra, 2. online at: `http://www.shef.ac.uk/~capra/2/roe.html` [accessed Jan. 2005]

[3] Sellers, W.I., Orton, R. & Chamberlain, A.T. (2001). Computer-aided visualisation of archaeological caves, *Capra*, 3, online at: `http://www.shef.ac.uk/~capra/3/sellers.html` [accessed Jan. 2005]

[4] Pflug, R. & Harbaugh, J. (1991). Computer Graphics in Geology. [Berlin: Springer-Verlag]

[5] Hamilton, A. (1977). Nottingham's Caves. [Nottingham Civic Society, England]

[6] Waltham, A.C. (1992). The Sandstone Caves of Nottingham, *Mercian Geologist*, 13(1).

[7] Edelsbrunner, H. & Mucke, E. (1994). Three-dimensional alpha shapes, *ACM Transactions on Graphics*, 13, 43-72.

[8] Requicha, A.A.G. & Rossignac, J.R. (1992). Solid modelling and beyond., *IEEE Computer Graphics and Applications*, 12(5), 31-44.

[9] Raper, J.F. (2000). Multidimensional Geographic Information Science. [London: Taylor and Francis]

[10] Kaufman, A., Cohen, D. & Yagel, R. (1993). Volume graphics, *Computer*, 26, 51-64.

[11] Karssenberg, D. & De Jong, K. (2005). Dynamic environmental modelling in GIS: 1. modelling in three spatial dimensions, *International Journal of Geographical Information Science*, 19(5), 559-579.

[12] Stoker, J. (2004). Voxels as a representation of multiple-return LIDAR data, *ASPRS Annual Conference Proceedings*, Denver, Colorado, May 2004.

[13] Carlbom, I. (1987). An algorithm for geometric set operations using cellular subdivision techniques, *IEEE Computer Graphics and Applications*, 7(5), 45-55.

[14] Watt, A. (2000). 3D Computer Graphics: Third Edition. [Harlow: Pearson Education Ltd.]

[15] Longley, P.A., Brooks, S.M., McDonnell, R. & MacMillan, B. (1998). Geocomputation: A Primer. [John Wiley and Sons Ltd.]

[16] Höhne, K.H., Bomans, M., Pommert, A., Riemer, M., Schiers, C., Tiede, U. & Wiebecke, G. (1990). 3D Visualization of Tomographic Volume Data using the Generalized Voxel-Model. [Hamburg: CH Volume Visualization Workshop]

[17] Funkhouser, T.A. (1993). Database and Display Algorithms for interactive Visualization of Architectural Models, PhD thesis, CS Division, UC Berkeley.

[18] Kavouras, M. (1987). A Spatial Information System for the Geosciences, PhD Thesis, University of New Brunswick, Canada.

Drawing up cave surveys by computer: the Tunnel software suite

Dave Loeffler

Once you are back on the surface after a surveying trip, there are two main parts to the job of drawing up your survey. The first is a purely mathematical one, that of calculating the most probable positions of your stations based on the tape, compass and clinometer data (or whatever other combination of instruments you are using). This job is ideally suited to computer calculation, and many software packages exist to solve the problem, whose merits have been extensively debated in past Compass Points articles. Once this first task is done, the next task is more artistic: to take the centreline you have calculated, and draw in the walls and other passage details around it. This is harder to automate to quite the same extent as calculating the centreline. Nonetheless a dedicated software package can make the task much easier. This article describes one such program, "Tunnel", which was written largely by Julian Todd with the collaboration of Martin Green and the author.

It is, of course, possible to draw up surveys using any of the standard computer drawing packages, without using any cave-specific software. This approach works well for certain caves - particularly small, simple caves that are fully surveyed within a short period. However, these packages become extremely frustrating for larger, more complex systems, or for long-term surveying projects where a major loop closure can require the whole survey to be redrawn. The difficulties encountered in correcting the first sheet of the Red Rose Cave and Pothole Club's Easegill survey when one of the entrance locations was found to be in error demonstrate this. The aim of Tunnel is to make this process easier, in several main ways. One of the most important is that the drawn-up survey can be warped if changes are made to the centreline (for example, due to a changes in the distribution of loop closure errors). In this article, I aim to describe Tunnel's main features, and to provide a quick-start guide to drawing up a cave using Tunnel.

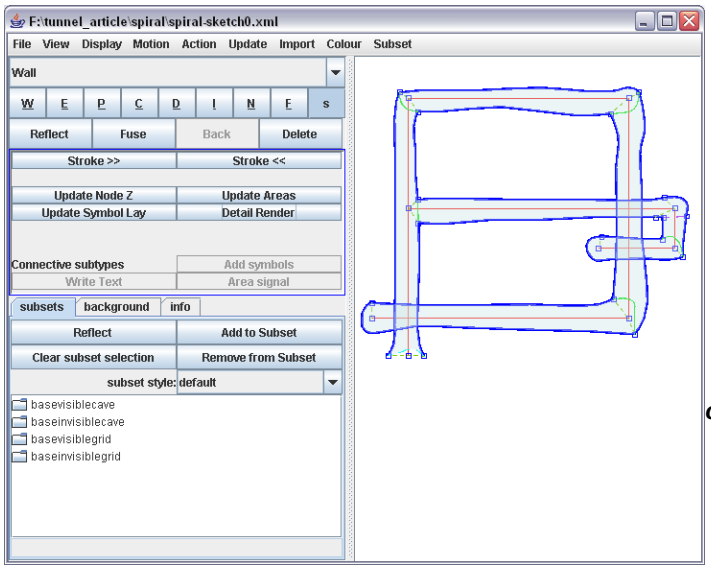
Importing your centreline data

Tunnel was designed to import centreline data in Survex's .svx format, but it can also import .TOP (Toporobot) and .PRJ (Walls) files; other import filters could be added if required. See [1] for a description of the Survex file format. When Tunnel imports a Survex file, it processes the data into its own format, which is based on XML. It also splits up the data into one file for each *begin/*end prefix block in the Survex file, which would typically represent one passage or survey trip (it is considered good practice to split your Survex files up in this way anyway) and puts each of these in a separate folder. Each of these folders will hold sketches and other files relating to this section of the cave. The advantage of using lots of small files is that it allows teams of people to draw up different parts of the same cave simultaneously, particularly when combined with versioning systems such as CVS [2].

Drawing a sketch

The core of Tunnel is its sketch drawing window, which allows you to draw in the walls of your cave using the mouse. We usually draw up each individual survey trip by hand around a print-out of the centreline. These sheets are then scanned and traced into individual Tunnel sketches. It is also possible to trace from scans of original underground survey notes, or even draw freehand; the latter requires some practice, but that is true for all computer drawing packages. What is shown in the sketch window differs from the final output, as different line types are highlighted in colour, but all labels and symbols are shown (unlike one of Tunnel's competitors, Therion, where labels are not visible until the final render, which makes accurate placement difficult – see [4] for a discussion of Therion's capabilities).

Tunnel supports a range of line types. Some of these are self-explanatory: a thick solid line for walls (dotted for estimated walls), a thinner line for passage detail, and the usual pitch and aven indicators with their tick marks. In addition, Tunnel calculates the two-dimensional shapes that these lines surround, so that it can work out which parts are inside the cave; by default, Tunnel shades in passages light grey in the final rendered output, to make them stand out against the white background (this is, of course, customisable). The two other main line types are invisible lines, which form the boundary of an area without having a line visible in the final output (so that, for example, areas of sand or mud can be defined), and connective lines, which are ignored in the area calculation and which serve to tie walls and other parts of the drawing to the centreline. Tunnel uses these connective lines to determine which areas should be drawn over or under others, based on the height of the centreline itself. They are also needed by Tunnel's warping algorithm used for loop closures. See Figures 1 and 2 for some examples of the area shading system.



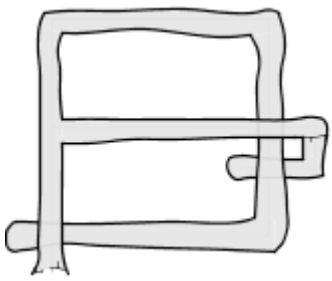


Figure 1: Here I have created a (rather contrived) example of a centreline that spirals downwards, with a side passage crossing over the main one then looping back under it. Tunnel automatically calculates which passages should be rendered above and below, based on the height of the centreline, to produce the output on the right. Note that the lower passages are still visible, as by default all passages are drawn slightly translucent; passages can be drawn opaque if desired, or made more transparent. The latter is useful for very complex surveys with many superimposed levels.

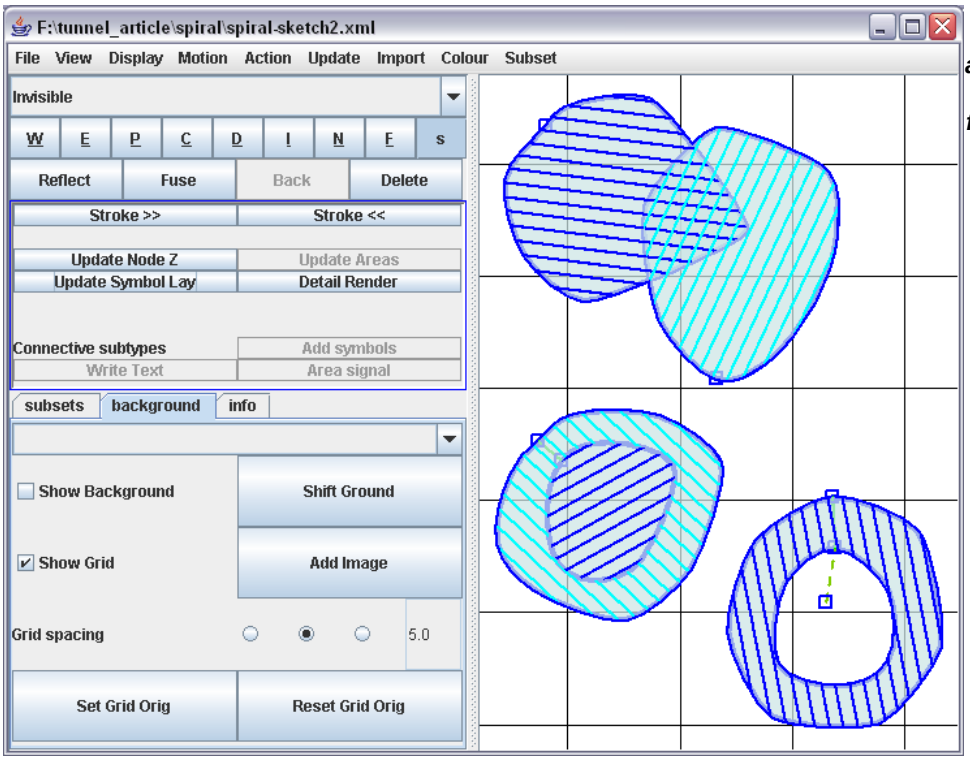


Figure 2: More examples of the area-detection algorithm. As Tunnel detects areas more or less by following walls, it assumes all areas are simply connected (that is, they have no "holes"). Areas may overlap, as in the first example, to allow for passages on multiple levels. Connecting two loops of wall with an invisible line, as in the second example, defines a ring-shaped area, but the central hole is also assumed to be passage; to draw a pillar, we also need to mark the inner area as rock, using a connective line with an "area signal".

Symbols

Tunnel allows a variety of types of symbols to be added to survey sketches. Some are added as single symbols (e.g. stream, slope and breeze arrows), but others are placed randomly to fill a specified area (e.g. sand, mud or boulders).

The algorithm for laying out random fields of symbols is a particular strength of Tunnel. It is actually a surprisingly difficult problem: how do you draw a realistic-looking field of random boulders? Placing them by hand is intolerably tedious. It is possible to draw a small section of reasonably random-looking boulders and tile this to cover the entire area, but this produces a rather odd effect, reminiscent of cheap wallpaper, as the eye recognises the similarities in the pattern. On the other hand, most genuinely random algorithms either produce lace-like patterns with numerous holes or perform poorly on areas of convoluted shape, or both. The algorithm Tunnel uses covers the area with a lattice, essentially approximating its shape with a bitmap, then places symbols at some random offset from each point of the lattice. This works remarkably well in practice (see Figure 3), especially when the size and alignment of the symbols is varied randomly at the same time.

The default set of symbols available within Tunnel covers most of those specified in the UIS standard [3], such as pools of water, various sizes of boulders and pebbles, stalactites and so on, but if this isn't enough, it is easy to design new ones: just create a new sketch in the "gsymbols" folder containing a drawing of the symbol, and add a line to the default style definition file specifying how it is to be laid out. The syntax of this file is quite complicated, but it is usually possible to just copy the code for an existing symbol that behaves in the same way.

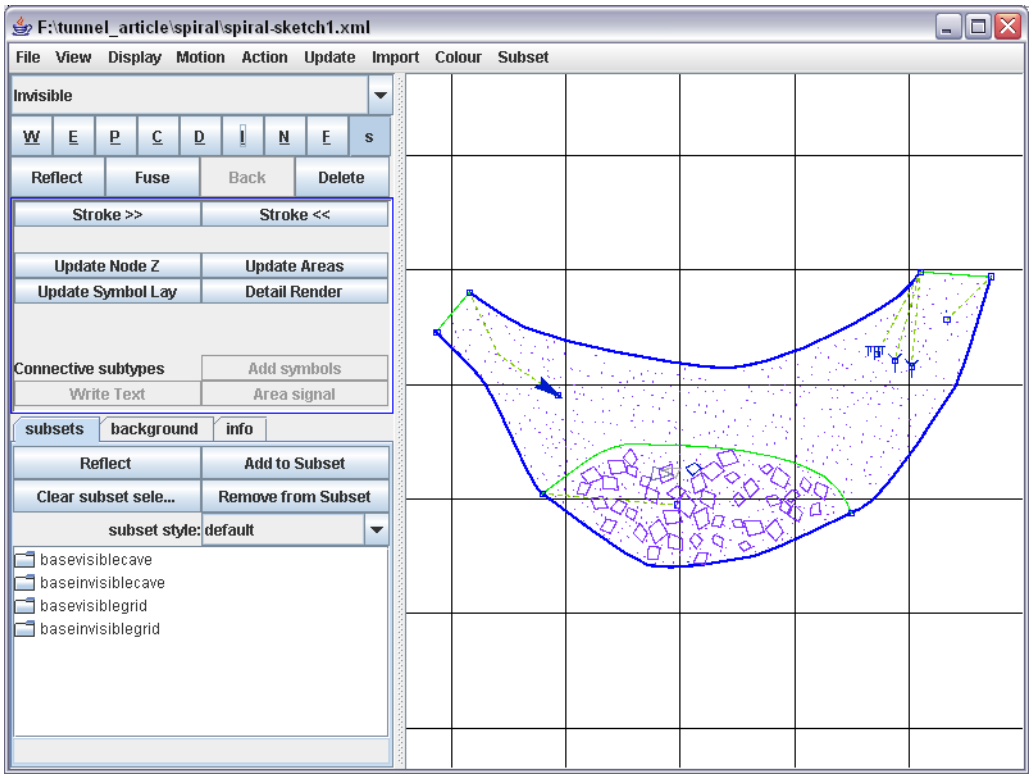


Figure 3: An example of laying out symbols in Tunnel. Notice the invisible line defining the area that contains the boulder symbols (this will not be shown in the final output).

A similar method based on connective lines is used to add text, although here the situation is simpler, as there is no need to spread text randomly across a specified area. Font sizes and colours are controlled by a system of font styles, so for example you might have a style for pitch depth labels, allowing you to easily change the appearance of the entire class of labels in one go (or even hide them entirely, if you want to produce a small scale version of your survey).

Joining it all together: The Import Sketch command

Having created sketches for the individual parts of the cave, the Import Sketch command allows these sketches to be combined into a single master sketch. This is somewhat analogous to the Therion concept of "scraps" [4]. However, unlike Therion, Tunnel has a multi-layer hierarchy: you can combine sketches A and B into a new sketch C, and then import C and another sketch D into a higher-level sketch E. This makes the task of joining your sketches together into a finished survey rather less intimidating, as you can combine bottom-level sketches of individual passages into a complete sketch of a particular area, then join this to other area sketches later.

When importing sketches, Tunnel automatically warps them to fit the centreline of the new sketch. This enables you to draw up sections of a large loop without the distortion induced by the loop closure. When you then import these sections into a sketch which includes the whole loop, it will bend them to take into account the loop closure error (see Figure 4). This figure simulates a case where a new loop closure somewhere else in the cave has forced the top right branch to be moved northwards, while the top left branch is fixed, perhaps as it is connected to something else. The left-hand side shows the original sketch; the right-hand side is the version warped to fit the new centreline. Note that the actual calculations of the station positions in the warped version were done with Survex, and imported into Tunnel; what Tunnel handles is bending the wall shapes to fit the new centreline. In practice, it is unlikely that a survey would need to be warped as grossly as this – this degree of misclosure would suggest a serious blunder in the surveying – but it serves to demonstrate the concept.

Outputting your finished survey

When you have finished drawing up your survey, there are various ways to export it. Tunnel can of course print directly to any printer supported by the operating system, at any scale; if the survey is larger than the paper size, it can be split between several pages with printed guides for cutting and pasting the result together. Alternatively, sketches can be exported to SVG (Scaleable Vector Graphics) format, which can be read directly by most drawing packages (and by the Firefox web browser; the Tunnel developers are working on the possibility of producing SVG-based interactive surveys with hyperlinks to text descriptions, photographs and so on). Finally, you can export to a bitmap file in TIFF, BMP or PNG format (although this is not ideal for print-resolution output, as the files tend to be unmanageably large). Most print shops should accept SVG files; the 600dpi A0 colour printout of the Cambridge University Caving Club (CUCC) Steinbrückenhöhle survey that was displayed at Hidden Earth 2005 was printed by this method, exporting an SVG from Tunnel and importing this file into Adobe Illustrator 10 on the print shop's computers.

Advanced formatting: the subset styles system

Over the last few years, much of Tunnel's development has been to support CUCC's survey of Steinbrückenhöhle in Austria. This cave system was discovered in 1999 and is now 11km long. Its complex layout includes six major planes of horizontal development corresponding to beds of rock sloping at differing angles and interconnected by numerous shaft series. As there are points where four or five passages in different levels cross over each other, the resulting survey is visually confusing if drawn up in the usual way. In drawing up the 2004 survey, Martin Green developed a system using multiple colours and transparency. This is implemented in Tunnel using a flexible system of "subset styles".

This works by grouping parts of the cave into subsets, which typically (but not necessarily) correspond to individual passages or survey trips (there is a command which automatically creates subsets based on the *begin/*end prefix hierarchy, which can then be fine-tuned by hand, or you can create your subsets from scratch if you prefer). A style definition file then organises these bottom-level subsets into a flexible hierarchy, which could (for example) correspond to the cave's main geographical areas or horizontal planes. It is then easy to specify how these areas are to be rendered, so one might choose to have the top level shown with a pale red background, the middle level in green, and the lowest level in blue with no symbols and pitch labels in italic; or any other choice of colours and styles that is needed. The style definition files are written in an XML-based format, which is immensely flexible but has the disadvantage of being difficult to edit; the Tunnel developers are currently working on ways to simplify the syntax without having to lose features.

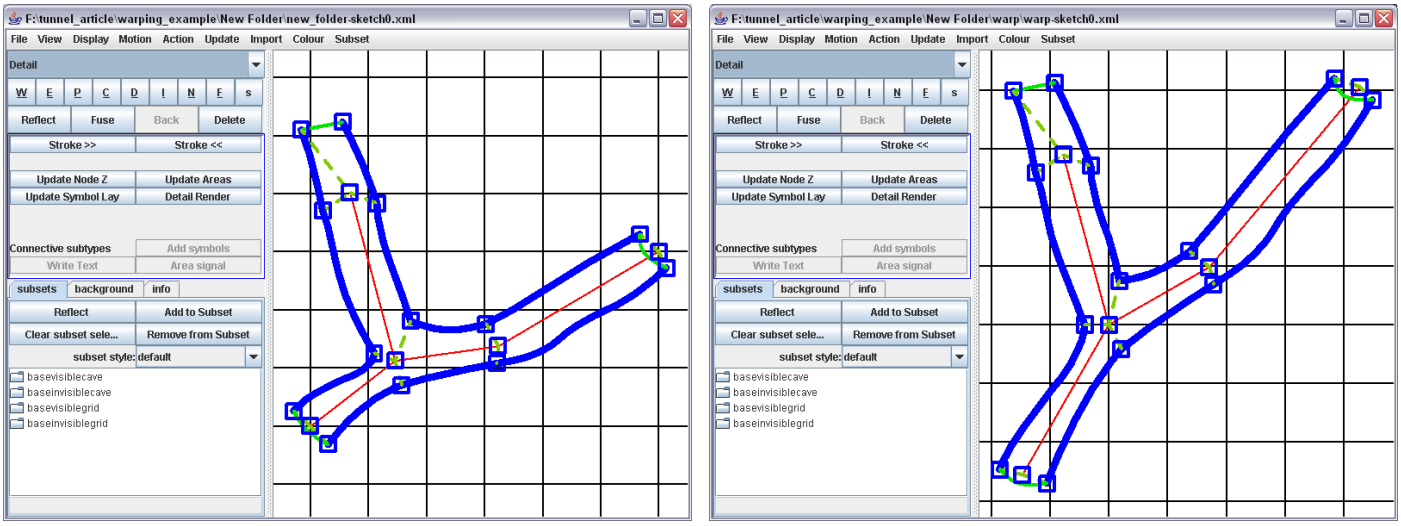


Figure 4: Example of warping a sketch following changes to the centreline due to loop closure adjustment (left is original, right is warped).

Future directions

At present, the main sketch-drawing interface of Tunnel has reached a reasonably stable state: further major changes in functionality are not contemplated at present, and current development is focussed on weeding out the few remaining bugs and making the interface somewhat easier to use – parts of it are rather haphazard, and the documentation is out of date following the substantial development effort that accompanied the drawing-up of over 30km of survey of Er Wang Dong and San Wang Dong for the Hong Meigui Cave Exploration Society's expedition to Houping last year. Features that are under development include a "necklace" algorithm for placing symbols along a line, allowing one to draw chains of slope or stream arrows, or a barrier of boulders terminating a passage.

However, we have some long-term plans. So far development has concentrated on drawing plans, as the convoluted nature of CUCC's Austrian caves means that elevations are formidably difficult to draw and, once drawn, are frequently incomprehensible. But we believe that it may be possible to use Tunnel to intelligently combine small sketches of extended elevation survey to produce larger extended elevations of whole caves or parts of them, or even to produce an approximation to a projected elevation by foreshortening the elevation sketches.

How to draw up a simple cave in Tunnel

This walkthrough describes the process of drawing up part of a real cave explored and surveyed by CUCC: Hauchhöhle, number 1623/234 in the Austrian Geological Survey catalogue. The starting point for the drawing-up process is a set of .svx files giving the centreline data, and scans of the original surveyors' drawings.

Step 1: Installing Tunnel

Tunnel can be downloaded from the SourceForge website, at `http://sourceforge.net/projects/tunnelx/`. Download a copy of the latest build from the tunnelx CVS repository. (The name Tunnel-X is something of an anachronism, as it was introduced to distinguish the main graphical Tunnel project from a very old DOS-based ancestor just called Tunnel, which no longer exists). Tunnel is written in Java, so it should work on any Java-supported platform (it has been tested under Windows, Macintosh, and various Unix variants). You will also need to download the Java Development Kit (JDK), available from the Sun website at `http://java.sun.com/`. The Tunnel directory contains a README.txt file which gives instructions on how to compile and run Tunnel.

Step 2: Importing Survex data

The first step in drawing up a survey is to import the centreline data and process it into Tunnel's XML-based format. To do this, use the "Open svx" command, then "Set XML dir" and "Save XML dir" to create the XML files. If the cave contains complex loops you will also need to tell Tunnel the final station positions calculated by Survex. To do this, process your survey with Survex in the normal way, use 3dtopos to create a .pos file, and place this .pos file in the top level of the new XML folder hierarchy; you need to close Tunnel and re-open it after this step.

Step 3: Creating a new sketch

The left-hand panel in the Tunnel window now shows a tree-style view of the *begin/*end prefix hierarchy of your .svx file (Figure 5). In our example, this is only one layer deep, but it can be divided up into as many levels of hierarchy as you need. Let's have a look at Left-Hand Passage, which is marked with the prefix "lefthand".

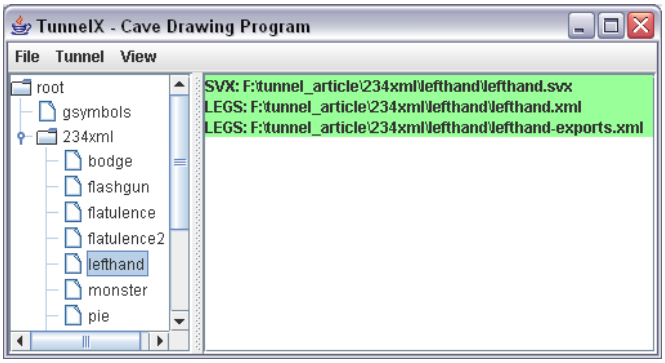


Figure 5: Tunnel window showing the survey hierarchy.

We want to add a sketch of this passage, which can be done by selecting "New Empty Sketch" from the Tunnel menu. This creates a new file called "sketch0.xml", and opens up the sketch editing window. Before we can draw anything else, we need some centreline data: select "Import Centreline" from the Import menu. The result is shown in Figure 6.

To get the hang of the interface, try drawing a few lines. The controls are based entirely on combinations of mouse clicks and modifier keys: if you click anywhere in the picture, it will start a line; you can add more points onto the line by clicking again, or Shift-click to finish the line. If you Ctrl-click on an existing point, the line you are drawing will snap to that point, or if you're not currently drawing a line it will start a new line based there. To select a line, right-click on it; you can then change the line type (the menu of line types is at the top left; "Detail" is the default line type), or press Ctrl + Delete to get rid of it. You can add new points in the middle of existing lines by selecting them and then Ctrl + Shift + clicking somewhere along the line. To scroll around your image, press Shift and drag with the middle mouse button (the wheel); Ctrl + middle mouse button zooms in and out. This interface based entirely on modifier keys and mouse clicks is a bit counterintuitive at first, but in practice it is far faster than having to pick different tools off a toolbar. If you don't have a middle mouse button, pressing the two buttons simultaneously should have the same effect. The only serious problem with this interface seems to be that most Macintosh computers have single-button mice; the developers are prepared to implement a workaround for this if anyone asks for one.

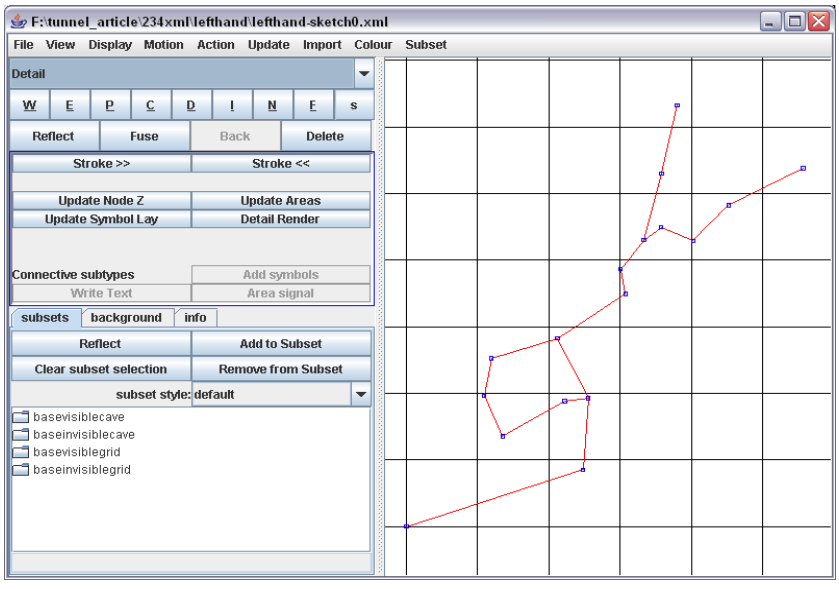


Figure 6: Imported centreline in the Tunnel window.

Step 4: Importing scanned drawings

Before we do any serious drawing, we need to import the existing drawn version of the survey, in order to trace over it. Click on the "Background" tab at the bottom left of the screen, make sure "Show Background" is ticked, and click on "Add Image". This will display the scan behind the image. Now you can move it around to fit the centreline: just draw a line from a point on the background image to the corresponding point on the centreline and click the "Shift Ground" button to move the first point onto the second one. To rotate and scale the background, draw a line with three points; this will move the second point to the third one while keeping the first one fixed. This should enable you to line up the red centreline with the centreline from the imported sketch. The result should look something like Figure 7.

Step 5: Drawing in the walls

The next stage is to draw in the walls. Select the "Wall" line type and draw a line around the wall of the part you want to draw. When you've drawn in the walls themselves, "seal up" the ends (where exploration stopped, or bits that are on a separate survey) with invisible lines. Click on "Update Areas", and it will work out what is cave and what is rock. Unfortunately, in my example "Update Areas" gets it wrong: it mistakes the loop in the passage for two chambers, one above the other. This can be corrected with an area signal, as shown in Figure 2 above.

You can just leave it at that if your survey is a simple one, but if the survey is complicated, you need to tell Tunnel a bit more about how the walls relate to the centreline, so that it knows which bits of wall to draw above and below others if they cross over, and so it can warp the drawing intelligently if there is a loop closure. To do this, add points along the walls every now and again, and join these to the centreline stations with connective lines. (There is an example of how to do this in the main article, in the discussion of warping). How conscientious you need to be at this stage depends on how much you think you are likely to need the layering and warping features, but at the very least you should have one connective line joining each segment of wall to the centreline.

Step 6: Adding labels and symbols

The original drawing shows a floor of boulders for this passage, so let's draw that in. First, use invisible lines to divide up the passage so sections with different symbols in them are in different areas. Then, for each area you want to add a symbol to, select a point on the boundary of the area and draw a connective line leading into the area, click "Add symbols", and select a symbol from the list. In the example in Figure 8, I have added boulders and pebbles to most of the survey, except the loop on the left, which has a bare rock floor, and the passage at the top, which is floored with mud; and I have put in some arrows for slopes and draughts. (The line attached to the draught is somewhat convoluted, as the symbol is actually not inside any area, but the connective line always has to start off heading *into* an area, so Tunnel knows which area to associate it to for layering purposes). To add text, we add a connective line in the same way, but use the "Write Text" button; there are various predefined text styles for question marks, pitches, passage names and so on, which can be customised if required. (Notice that I've turned off "Show Background" in the figure to make it easier to see what's going on.)

Step 7: Combining your sketch with the rest of the survey

Let's suppose we have another sketch into which the rest of the survey has been already imported, and we want to join our Left-Hand Passage sketch onto this. Open this master sketch in the sketch editing window, then flip back to the file list window and select the other sketch we've just drawn. Now go back to the sketch editing window and select "Import Down Sketch" from the Import menu. This will bring in the Left-Hand Passage sketch, and you can zoom in and do the fine work of stitching together the two (Figure 9).

This can be done semi-automatically: if you have two wall lines, one from each sketch, you can fuse them together – just draw a line from one to the other and click "Fuse".

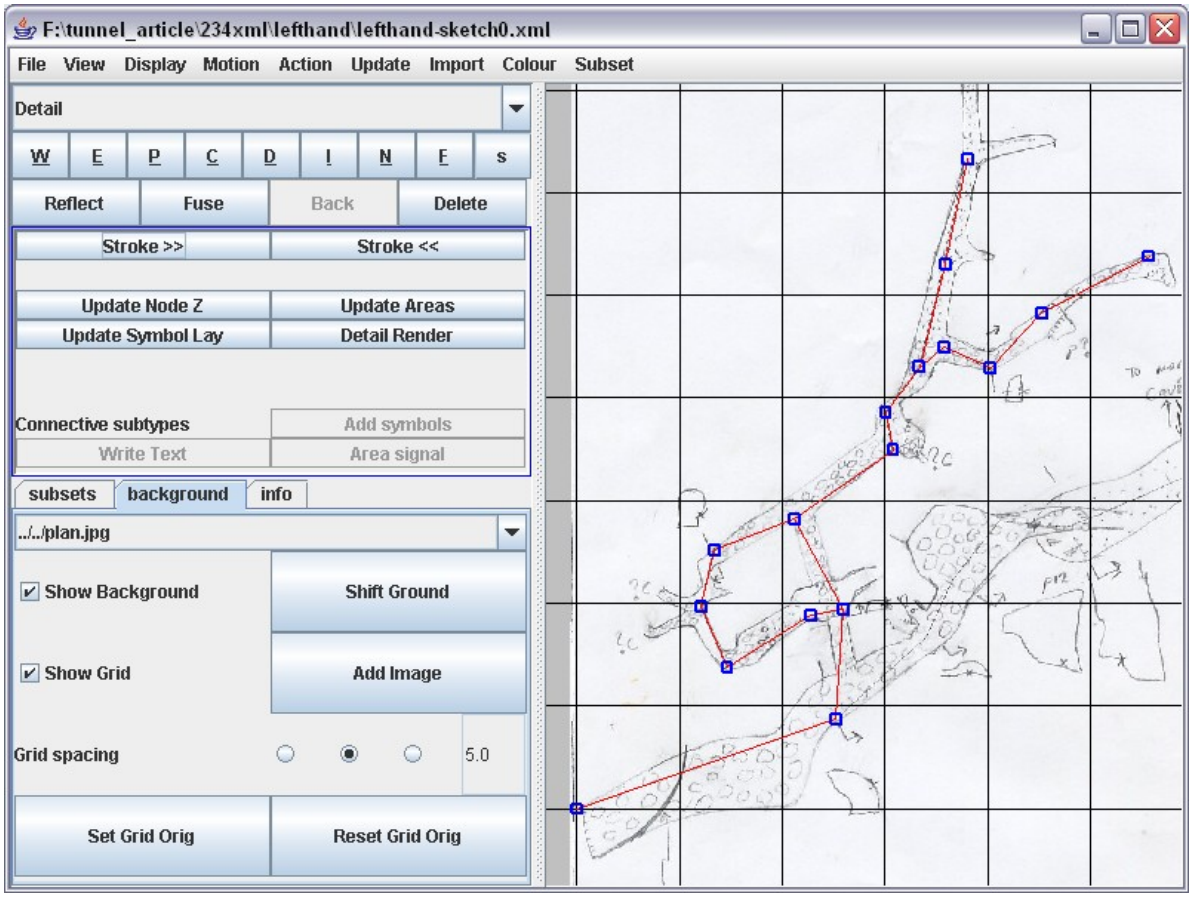


Figure 7: A survey sketch aligned with the centreline.

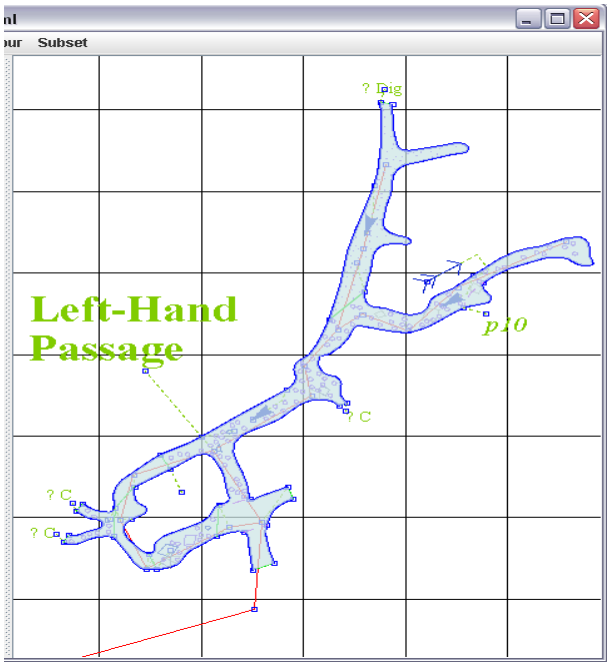


Figure 8: Survey segment with walls traced from the original sketch and passage details added.

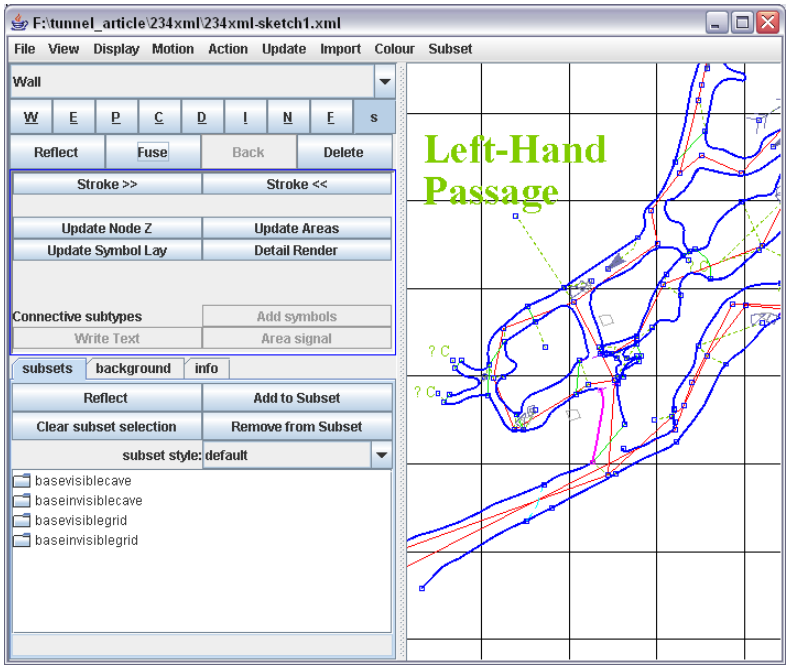


Figure 9: Joining two survey segments together.

Step 8: Rendering and exporting your sketch

When you are happy with the join between the imported sketch and the master, select "Update Everything" from the Update menu, which calculates the node Z coordinates, updates the areas and updates the symbol layout. Now your survey is ready to be printed, or exported as SVG or bitmap, using the File menu. The entrance area of Hauchhöhle ends up as shown in Figure 10.

Conclusion

The developers hope that Tunnel has achieved at least in part its aims of providing a system that allows complex cave surveys to be drawn up on a computer, and that it overcomes some of the disadvantages inherent in drawing up surveys using general-purpose drawing packages. If you have a survey to draw up, we encourage you to give Tunnel a try. More information about how to obtain and use Tunnel is available from the Tunnel website [5]; alternatively, please feel free to contact Julian (`julian@goatchurch.org.uk`), Martin (`mjg54@cam.ac.uk`) or myself (`dave@cucc.survex.com`).

References

[1] Betts, O. & Wookey. Survex 1.0.39 Manual: Survex Data Files, online at: `http://www.survex.com/docs/manual/datafile.htm`

[2] Ximbiot CVS Wiki, online at: `http://ximbiot.com/cvs/wiki/`

[3] Survey and mapping working groups of UIS informatics commission, online at: `http://www.uisic.uis-speleo.org/wgsurmap.html`

[4] Wookey (2004). Therion – state of the art cave-drawing software, *Compass Points*, 33, 5-12.

[5] TunnelX – Index to information, online at: `http://www.goatchurch.org.uk/tunnelx/`

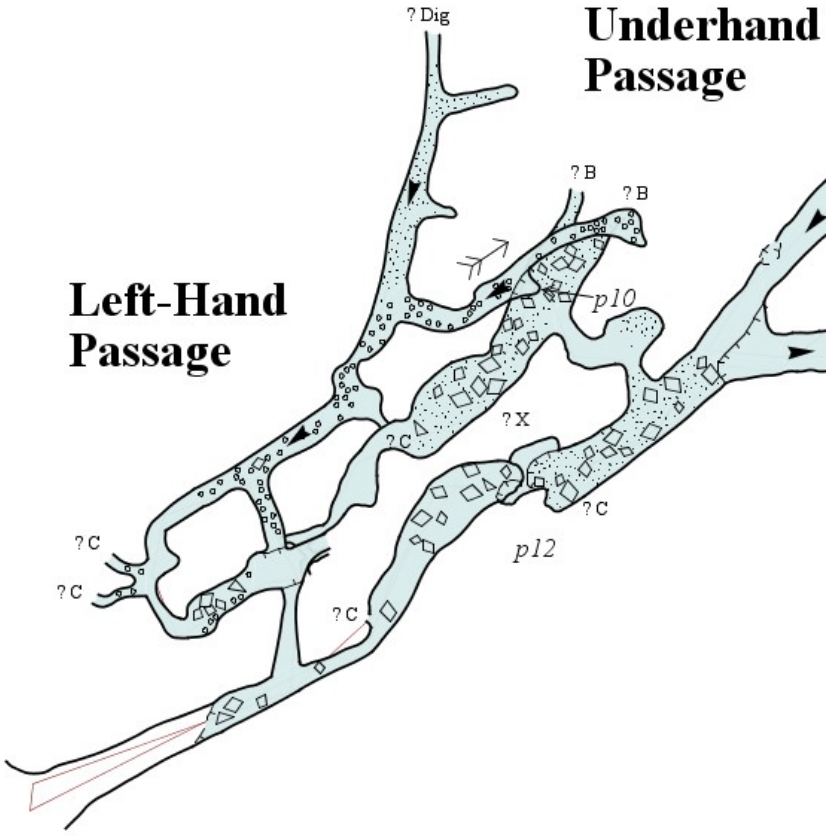


Figure 10: Final version of the Hauchhöhle entrance survey.