TunVis

Visualizing specific geologic features for tunnel planning and construction

Abstract

This paper describes a method and a system for visualizing a virtual tunnel through an existing geologic formation and for the generation of a camera animation along the tunnel axis.

The tunnel is built from geologic data coming from vertical sample drills from the mountain top along the choosen tunnel axis. Its wire frame model is then edited by a geologist and image information of the rock appearance is textured onto the tunnel wall, resulting in a photorealistic representation of the expected geologic occurrences.

Finally a camera animation and video options are defined to render a video sequence of the camera movement.

Introduction

Engineering Geology is a scientific technical discipline, which has gained interest due to conservation-consciousness and growing safety demands of the people, especially in huge tunnel building projects.

Usually a technical geologist is called in for planning and construction of ground buildings. He investigates areas of interest and provides the expert opinion, including all acquired data.

As the description of geologic information is typically a three-dimensional problem, graphical represenations are used to show the facts in an easily understandable way. These representations may still be difficult to interpret for non-trained persons. That is why increased effort is put into the development of three dimensional visualisation systems for geologic information.

Research is done mainly in the simulation area. Geologic visualization systems (like Geo3View, [KlRa91] and [LiUl95]) often model terrains as triangular surfaces. After running simulation processes the resulting surfaces are shaded and displayed using common computer graphics techniques. Thereby the resulting model may be arbitrarily clipped and viewed from any position in object space. As mentioned, these systems visualize simulated data, but do not focus on a photorealistic animation.

This paper focuses on two issues: Real data and photorealistic animation. Real geologic data (as opposed to data coming from simulations) is converted into an analytical representation and then used to model a static tunnel. The tunnel consists of many parts each with different rock characteristics. The parts of the resulting wire frame model are textured individually to show the varying visual characteristics. Finally a camera path is defined along the tunnel axis, a video sequence is calculated and recorded on a VCR.

The methods in this paper were implemented as TunVis, a software system that enables geologists to visualize certain tunnels or tunnel pieces of interest. It was developed in cooperation with a geologist and is in use to create video sequences of camera animations through planned tunnels. These video sequences are intended to explain the results from geologic investigations in a graphical way. They provide a better understanding among non-experienced persons.

The paper does not focus on implementation details, but instead is intended as motivation and reference to solve problems concerning the visualization of tunneling tasks. It starts with the acquisition of tunnel data, which are used to create a wire frame model of the tunnel. Then it describes the tunnel - strata intersection, and how it divides the tunnel into parts. These parts are textured and a camera animation is defined. After a section that focuses on algorithmic details of the used methods, some remarks follow on the implementation of TunVis as well as a results section.

Tunnel data acquisition

Vertical sample drills are done on the mountain top along the selected tunnel axis to investigate position and orientation of the underlying rock formations. The borders between formations of different characteristics are written down in a geologic notation. A cross section of sample drills is shown in figure 1.

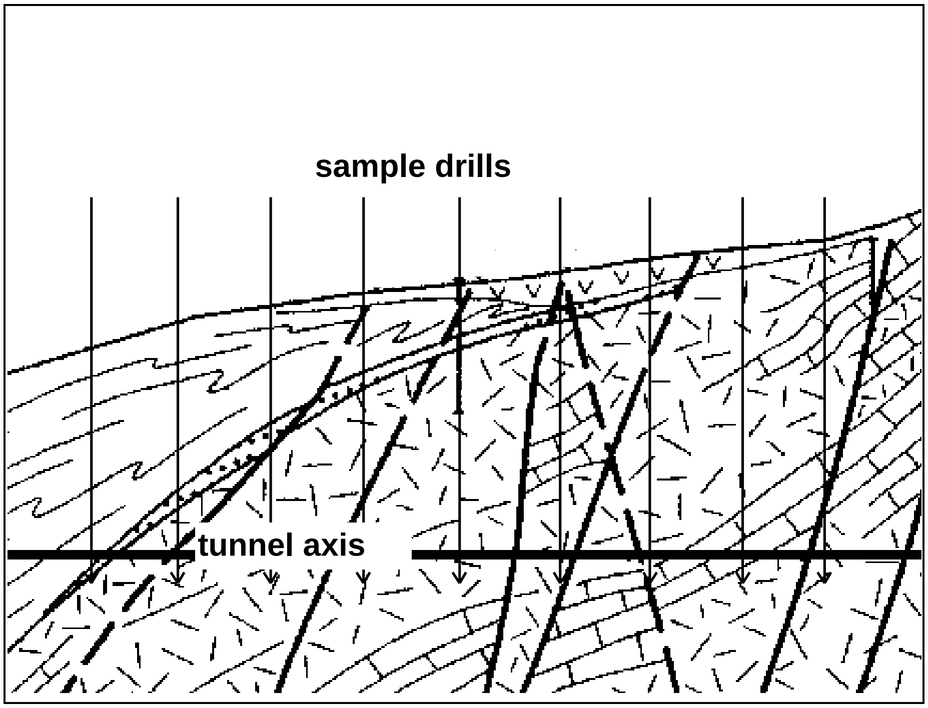

figure 1 - cross section of vertical sample drills

In figure 1 one can see very clearly the strata, that make up the mountain. The tunnel visualisation should exactly show the boundaries beween different strata and each stratum may have a specific visual appearance. The tunnel must therefore be partitioned into pieces, each of which represents the intersection between the tunnel and a certain stratum.

Wire frame model of the tunnel

The tunnel itself is represented as a tube, whose base lies in the xy-plane of the world coordinate system. Its cross section consists of a circle with a small segment cut off. The tunnel axis is aligned with the z-axis of the world coordinate system. The cross section is then swept along the z-axis as shown in figure 2.

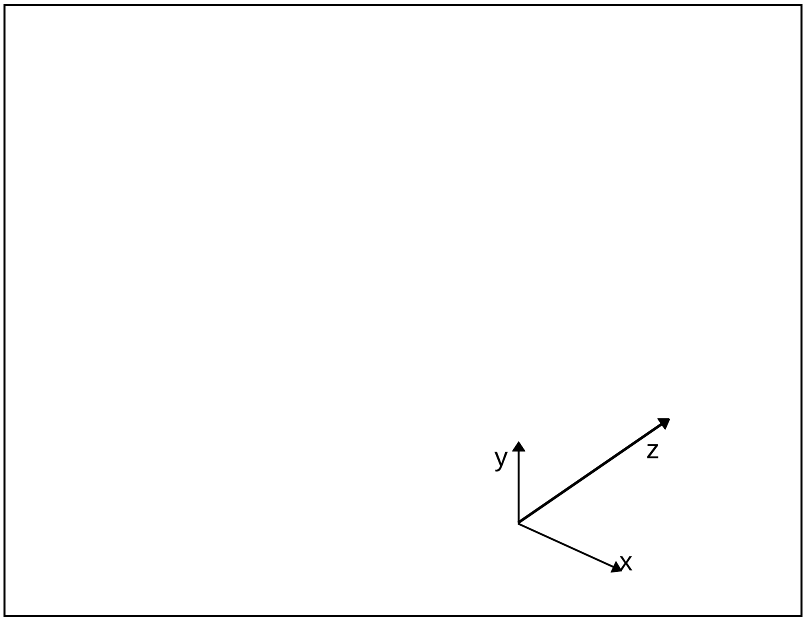

figure 2 - tunnel representation, sweeping a cross section along the z-axis

The tunnel pieces are modelled by intersecting the tunnel with all given strata. A stratum is assumed to be bounded by two bounding planes within the bounds of the tunnel. Position and orientation of these planes are taken from the geologic map of the sample drills.

The axis of a planned tunnel may have curves. Because the camera simply moves along the tunnel axis, curves of the axis are straightened. The planes must therefore be adjusted. This is done by rotating the plane by the negative angle, around which the axis is bent on that point.

Texturing individual tunnel pieces

Texturing ([FoDa90]) is used to give the tunnel pieces a realistic appearance. A stratum has a three-dimensional expansion and is often aligned with neighboring strata. This alignment takes place in space. Therefore it would be desirable to treat textures in a three-dimensional way.

However, texture images are two-dimensional and in most cases images of rock formations have anisotropic behaviour. A three-dimensional rotation of a stratum can not easily be represented by a transformation within its 2D texture image, if the texture has anisotropic characteristics.

If an anisotropic image is rotated only in 2D, spiral effects occur in the tunnel after texturing. For example a texture image representing a stone wall was rotated by 15° and then mapped onto the tunnel wall in figure 3.

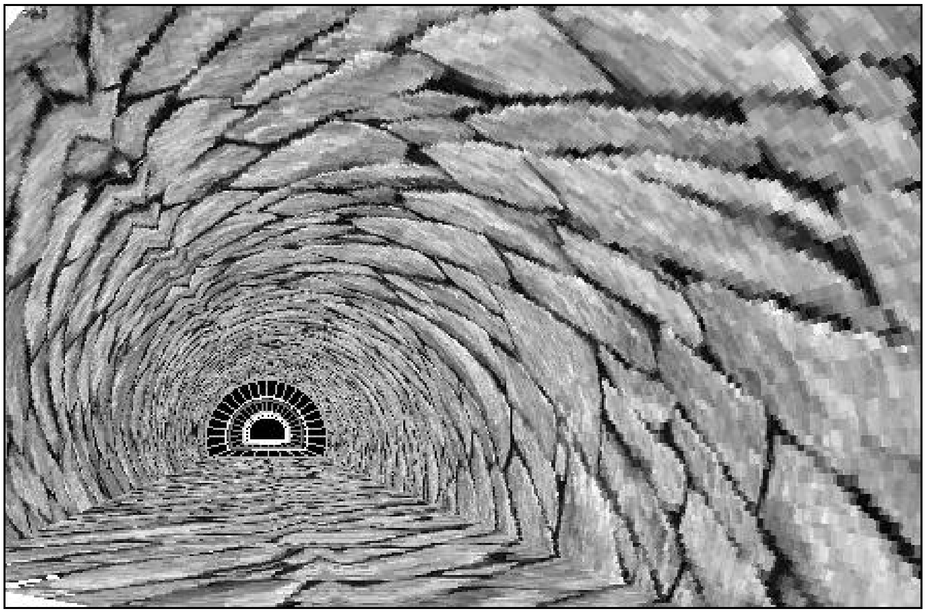

figure 3 - spiral effect after rotation of 2D texture

One way to solve the problem of rotating a stratum in the tunnel is to use solid texturing ([FoDa90]). The problem is, that 3D anisotropic texture data is not available for most of the stone materials. Its generation is a nontrivial problem. One possibity can be found in the generation of solid textures from 2D images. A good approach for the generation of solid textures from isotropic samples is described in [HeBe95], but it still fails to construct an anisotropic solid texture from a 2D sample.

Therefore the only possibility is to use 2D textures. Due to this restriction, rocks can not be aligned with the stratum they lie in. In addition, to avoid the spiral effect, anisotropic texture images have to be restricted to horizontal and vertical orientations.

Camera animation along the tunnel axis

Once the tunnel is modelled and textured, a camera is defined and the scene is rendered ([Watt93]). Animation is done by moving the camera incrementally along the axis through the tunnel and generating frames.

Algorithmic details of TunVis

To model the tunnel two kinds of data are necessary: the cross section of the tunnel and the intersecting strata, represented by their bounding planes.

Tunnel cross section

A tunnel in the system TunVis is based on a simple cross section. The cross section is created by linking the endpoints of a 220° circle segment with a base line (see figure 4).

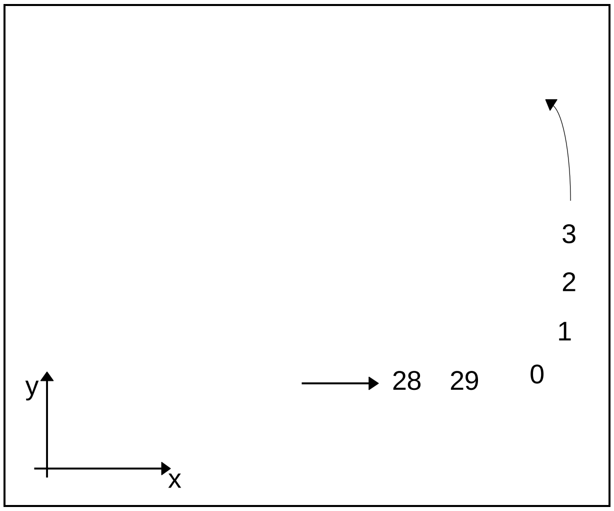

figure 4 - tunnels cross section

The cross section is partitioned into 30 pieces. The endpoints of these pieces are numbered from 0 to 29. When the tunnel is swept along the z-axis (as shown in figure 2) the cross section is positioned at the beginning and at the end of the tunnel. Corresponding points are connected to form strips. Strips are numbered with the numbers of the corresponding points. Therefore the raw tunnel consists of 30 strips, each numbered in the range from 0 to 29.

Intersecting strata

Strata are represented by two bounding planes each. These planes are taken from the geologic map of the sample drills. A plane is represented in the geologic notation by its **azimuth angle a** and its **dip angle j** ([Wall86]). For an explanation on these parameters, see figure 5.

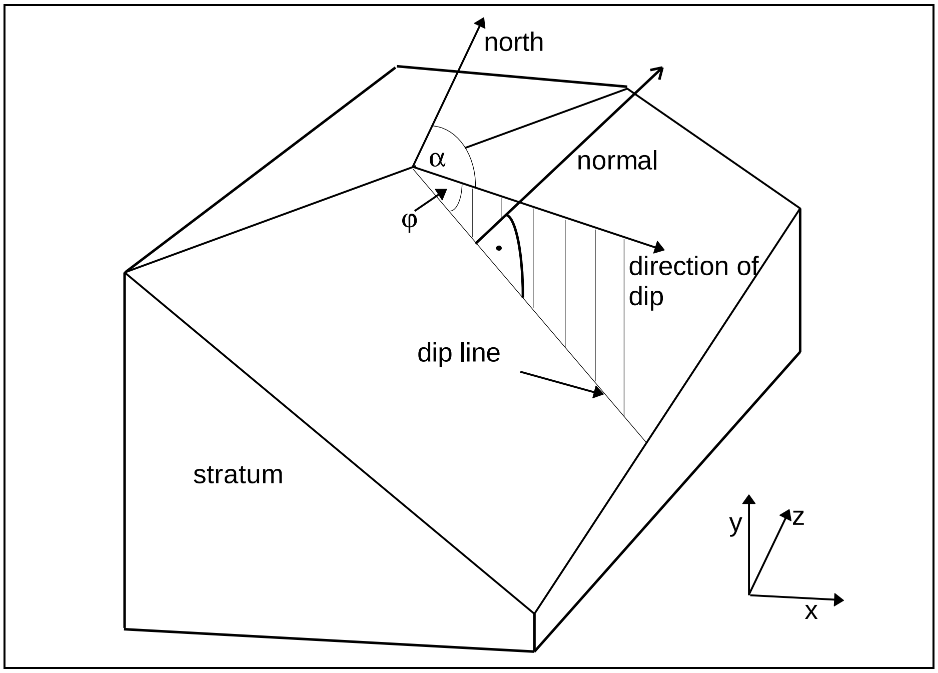

figure 5 - geologic notation for defining a plane

The straight line on the plane with the largest angle of inclination is called dip line. Its projection into the horizontal plane is called direction of dip. The angle between these two lines is the dip angle φ. The angle between the north direction and the direction of dip is the azimuth angle α.

The orientation of a stratum's bounding plane is defined by its azimuth angle α and its dip angle φ. For the transformation of the geologic notation into the analytical representation of planes, we calculate the plane normal. Therefore we assume, that the z-axis in the coordinate system is equivalent to the north direction. In figure 6 one can easily derive the x - and z - components of the plane normal.

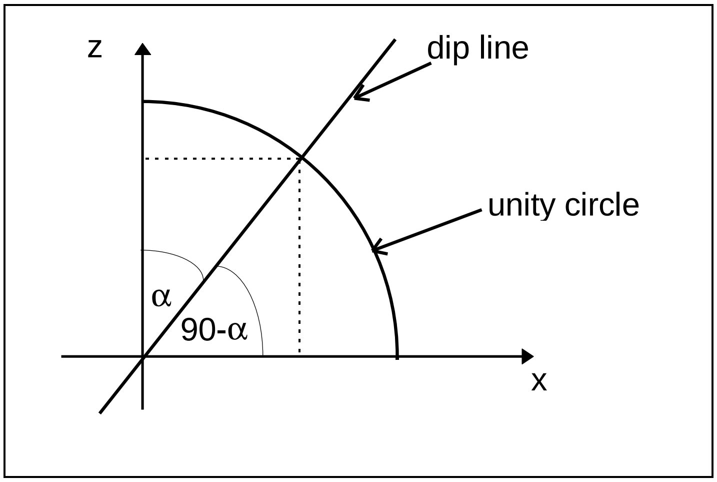

figure 6 - calcualtion of x- and z-component of plane normal

$$x = \cos(90° - \alpha) = \sin(\alpha) \tag{1}$$

$$z = \sin(90° - \alpha) = \cos(\alpha) \tag{2}$$

To determine the y - component we take a perpendicular look at the plane between the dip line and the normal (see figure 7).

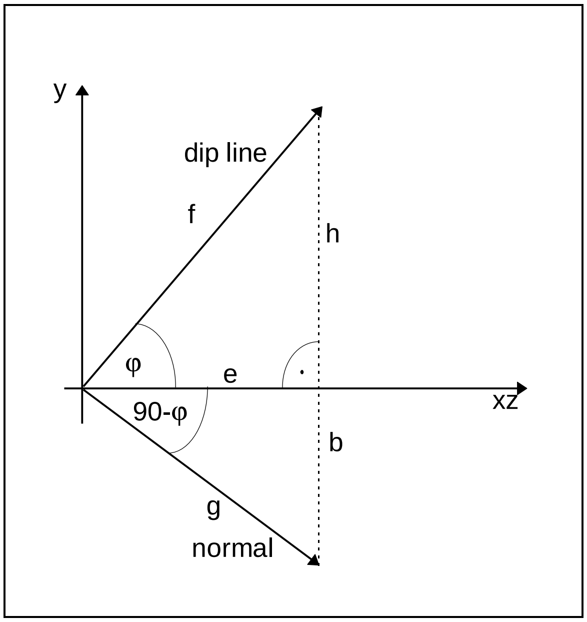

figure 7 - calcualtion of y-component of plane normal

One can see, that h:e = e:b. Since e equals 1 (radius of unity circle), h = 1/b and b=1/h. Now h = tan(φ), and thus:

(3) $b = 1 / \tan(\varphi)$ if $\varphi < 90°$
$b = 0$ if $\varphi = 90°$

To straighten the tunnel axis, we rotate it towards the north direction. The angel σ between the north direction and the direction of tunnel axis is taken from the geologic map. The axis is rotated by the angel σ. To adjust the planes correspondingly, the azimuth angle is replaced by the **relative azimuth g** In figure 8 one can see, that γ evaluates to:

(4) $\gamma = \alpha - \sigma$

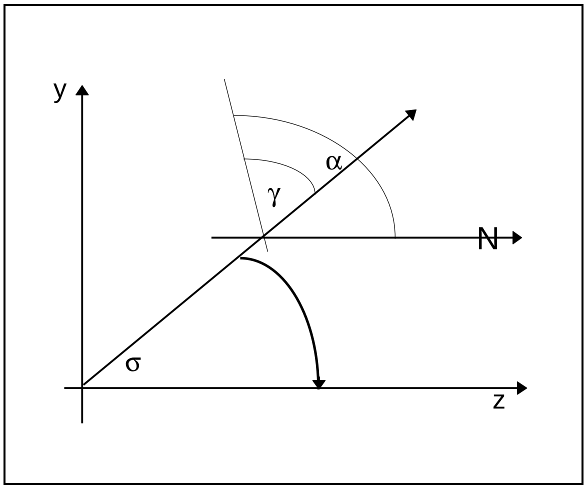

figure 8 - calculation of relative azimuth

A plane is represented by the analytical equation

(5) $ax + by + cz + d = 0.$

The coefficients a, b and c represent the plane normal.

(6) $a = \cos(90° - \gamma) = \sin(\gamma)$

(7) $b = 1 / \tan(\varphi)$

(8) $c = \sin(90° - \gamma) = \cos(\gamma)$

The **position s** of a stratum's plane denotes the intersection point with the tunnel axis. It is determined by the distance someone has to cover on the way along the axis from the tunnel entrance to the intersection point of the plane with the axis.

As the tunnel axis corresponds to the z - axis of the coordinate system and the tunnel has been straightened, the position s denotes the z-value of the axis - plane intersection. Thus the missing coefficient d can be calculated by substituting the point (0,0,s) into equation (5).

(9) $d = -(c * s)$

Tunnel - plane Intersection

To construct the tunnel pieces from the raw tunnel and the planes, each strip of the tunnel is intersected with all occuring planes and thus divided into strip-pieces, called scans. The scans are then used to model a piece of the tunnel lying between two planes of a stratum. A tunnel piece thus consists of scans.

The algorithm used in this work is called scanstrip algorithm. It works as follows:

Each strip on the raw tunnel is intersected by all planes. Since all points on a strip have the same x- and y-coordinate, only z-values are stored. On every strip these z-values are sorted in an ascending order. For an explanation of the next steps, see figure 9.

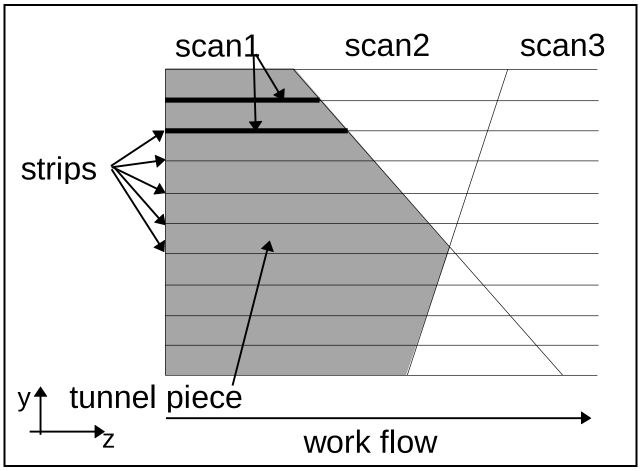

figure 9 - building tunnel pieces through scans on a strip

Starting from the beginning of the tunnel, the algorithm considers only the first scan on every strip around the tunnel. These scans are assigned to one tunnel piece. In the next step, the second scans on every strip are assigned to a tunnel piece and so on.

For the assignment of scans to tunnel pieces, the following considerations have to be taken into account:

- Two scans belong to the same tunnel piece, if there is no intersection among two different planes between them (see figure 10). In this case both scans are assigned to the same tunnel piece.

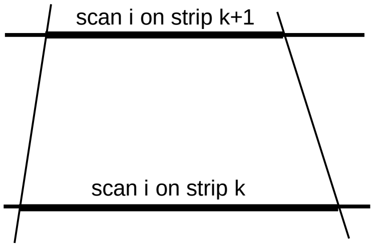

figure 10 - Scanstrip, no intersection

- Two scans belong to the same tunnel piece, if there is one intersection among two different planes between them, either on the left side, or on the right side. In this case a new scan has to be calculated at the intersection point S and assigned to the tunnel piece. See figure 11 and figure 12.

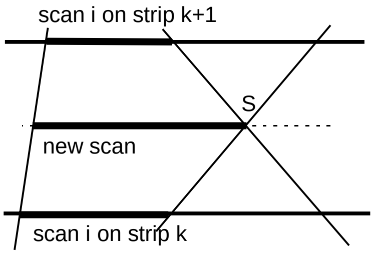

figure 11 - Scanstrip, intersection to the right

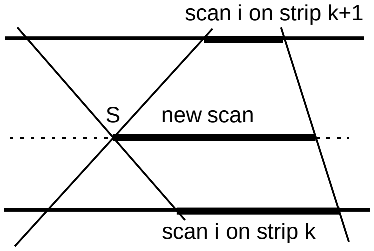

figure 12 - Scanstrip, intersection to the left

- Two scans belong to two different tunnel pieces, if two planes cross over between them (i.e. the first plane on the first scan is the second plane on the second scan and the second plane on the first scan is the first plane on the second scan, see figure 13). In this case the intersection point S is used to end one tunnel piece and to start another.

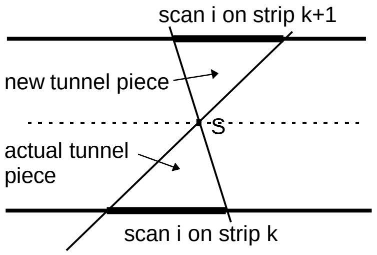

figure 13 - Scanstrip, crossover

- It is assumed that no other cases, like the one in figure 14, take place in practice. In such a case three planes would have to lie very close to each other. With the sparsly located sample drills this can not be the case.

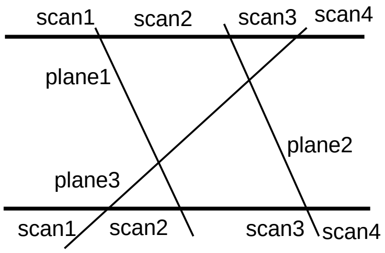

figure 14 - Scanstrip, planes too close to each other

The starting point of every step in the scanstrip algorithm is strip zero. At this point the scans are assigned to a temporary tunnel piece object, because the scans from strip 29 and below might belong to it (see figure 15). On the processing of all strips new tunnel pieces might be started. However, when the algorithm reaches strip 29, the temporary tunnel piece object is assigned to the current active piece.

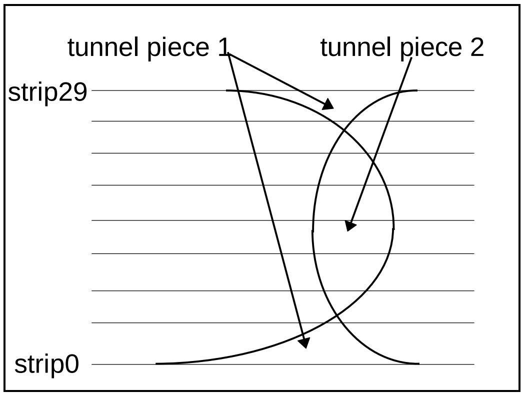

figure 15 - begin and end of tunnel pieces

The calculation of intersection points of planes between two scans is done by linear interpolation:

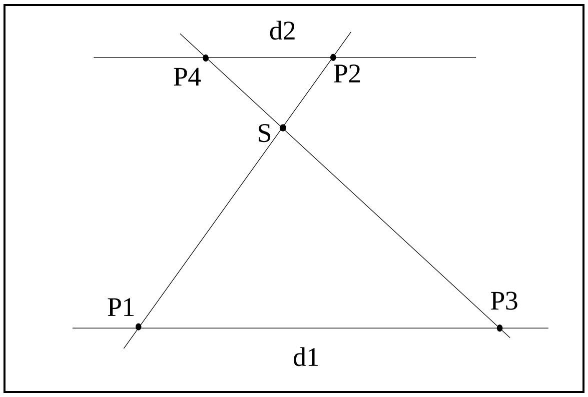

figure 16 - linear interpolation between two scans

With figure 16, the intersection point S can easily be calculated:

$$S = P1 * \frac{1}{(1+\frac{d1}{d2})} + P2 * \frac{1}{(1+\frac{d2}{d1})}$$

Textures

To give the tunnel pieces a realistic appearance, they are textured, using photographs of existing rock formations.

When mapping these images onto the tunnel wall, two different methods can be applied: mapping exactly one image on one tunnel piece, or mapping multiple instances (or parts) of an image periodically onto one tunnel piece (see figure 17).

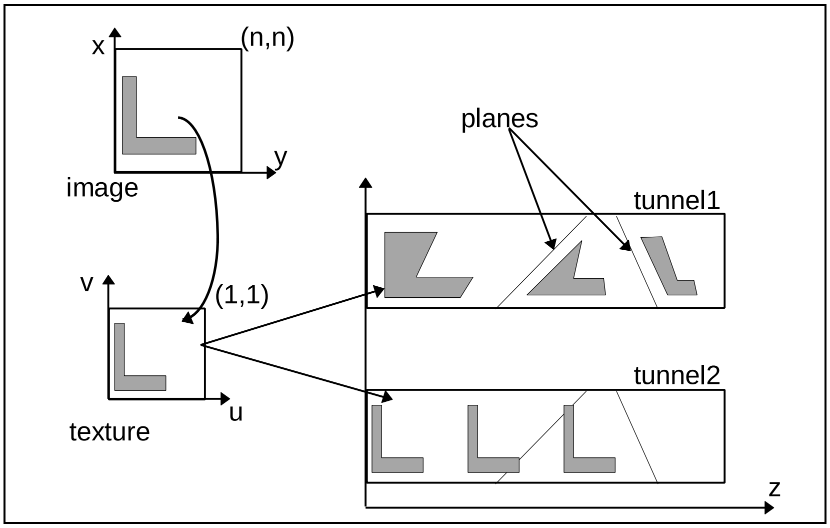

figure 17 - different types of texture mapping for tunnel pieces

The photographic image shows an n x n meter extract of the real world. The texture coordinates range from (0,0) in the lower left corner to (1,1) in the upper right corner.

In the first case (see figure 18) texture coordinate (0,0) is mapped to the front border of the tunnel piece at strip 0. Coordinate (1,0) is mapped at the corresponding point at the back border (at strip0).

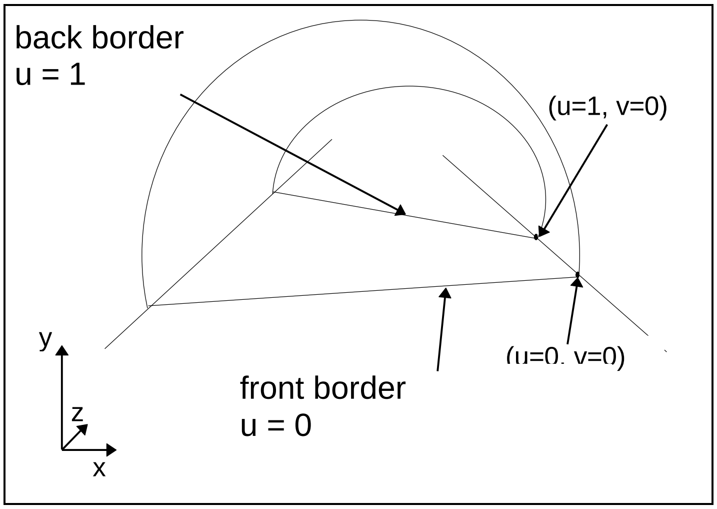

figure 18 - front and back border of a tunnel piece

Correct scaling between the n x n resolution of the photography and the length of the tunnel can not be applied in this case. Instead, the image is aligned with the tunnel pieces' borders. In some cases the three-dimensional alignment of a stratum's orientation with its bounding planes can be approached with this effect. The drawback of this method becomes apparent, when two neighboring pieces are textured with the same image. Because of the dissimilar image distortion on both pieces, a sharp border between them occours (see figure 19).

In the second case texture coordinate u = 0 is mapped to the beginning of the entire tunnel, and images are mapped periodically onto the tunnel pieces. Thus a correct scaling between the n x n resolution of the photography and the length of the tunnel can be applied. This is especially advantageous when linking together two neighboring tunnel pieces, i.e. mapping the same texture on them. The difference between the two methods is shown in figure 19.

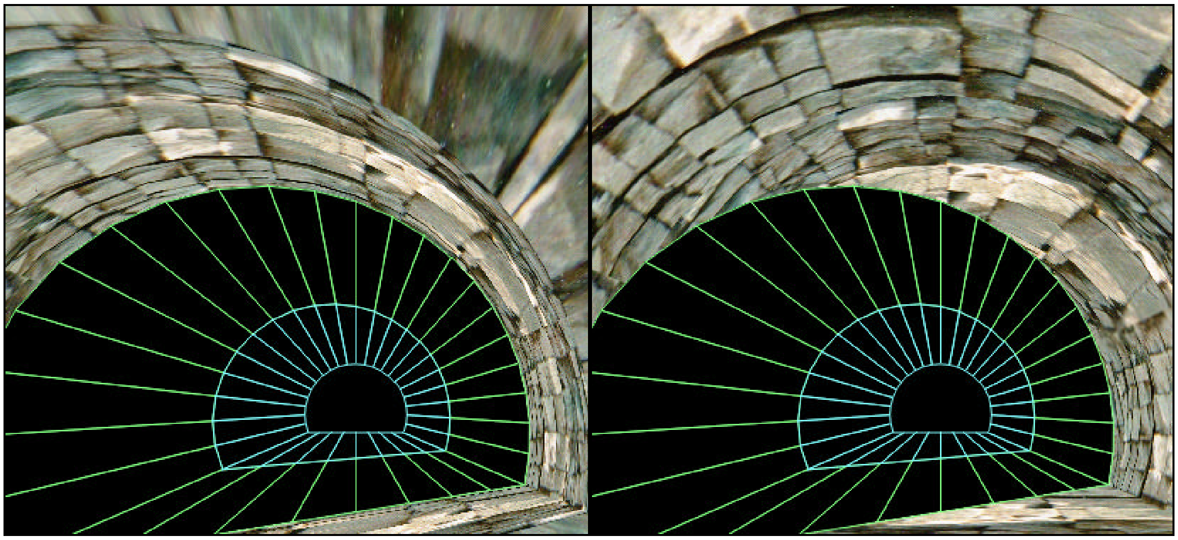

figure 19 - linking tunnel pieces with periodic and non periodic textures

In the v direction (on the circumference of the tunnel), always periodic texturing is applied. See figure 20 for an explanation of the used mapping method.

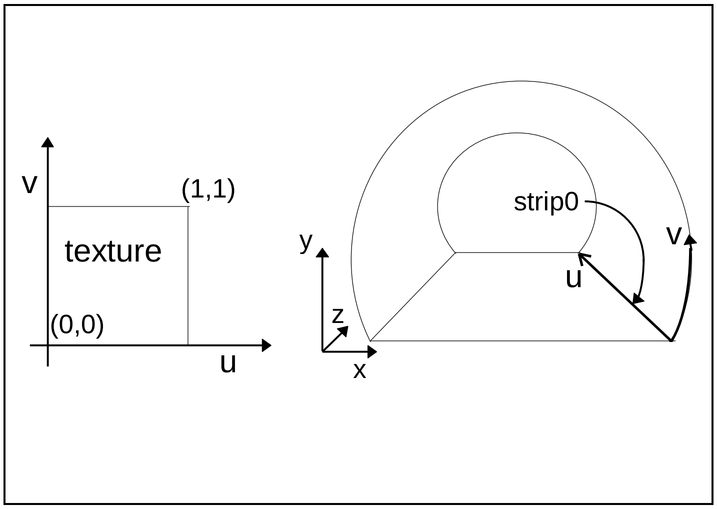

figure 20 - texture mapping on tunnel

Output

Video creation is done by moving the camera along the z-axis and capturing single frames from the frame buffer. These images are then coded into JPEG files ([PeMi93] and [WaGr91]) and put together to a digital video sequence.

Implementation

The methods in this paper were implemented in TunVis, a system that enables geologists to visualize certain tunnels or tunnel pieces of interest. It was developed as a comissioned work and is in use to create video sequences of camera animations through planned tunnels.

TunVis was implemented under Windows95 on a 133MHz Pentium PC, using Microsoft Visual C and its MFC ([Chri92]) for the Windows specific programming. The graphical programming was done with the MS Windows implementation of the OpenGL language ([DaWN93] and [Open92]).

In TunVis a tunnel is created by entering geologic data into a table. The wire frame model of the tunnel pieces can then be interactively edited by selecting certain pieces and assigning textures to them. The texture images are taken from a database of geologic photographies, which has been designed for this purpose.

Afterwards the camera animation is defined, frames are rendered and compressed to JPEGs. Finally the JPEGs are used to create a digital video sequence, which may then be edited and recorded on a VCR.

Results

The creation of a video sequence was tested by a technical geologist. He visualized a 1.350 meter tunnel with the intersection of 13 strata. The sequence was recorded at a frame rate of 25 FPS and had the duration of 60 seconds.

The time effort for creating the video sequence amounted to 9 hours and 50 minutes, divided into 15 minutes for entering geologic data, 30 minutes for texturing the tunnel and 9 hours and 5 minutes for the system to calculate the 1.500 frames. The storage effort for the 1.500 resulting images was 576 MBytes.

A snapshot of the resulting video is shown in figure 21. A short sequence of it (as well as a short description of TunVis) can be found in the World Wide Web at *http://www.cg.tuwien.ac.at/research/vis/index.html*

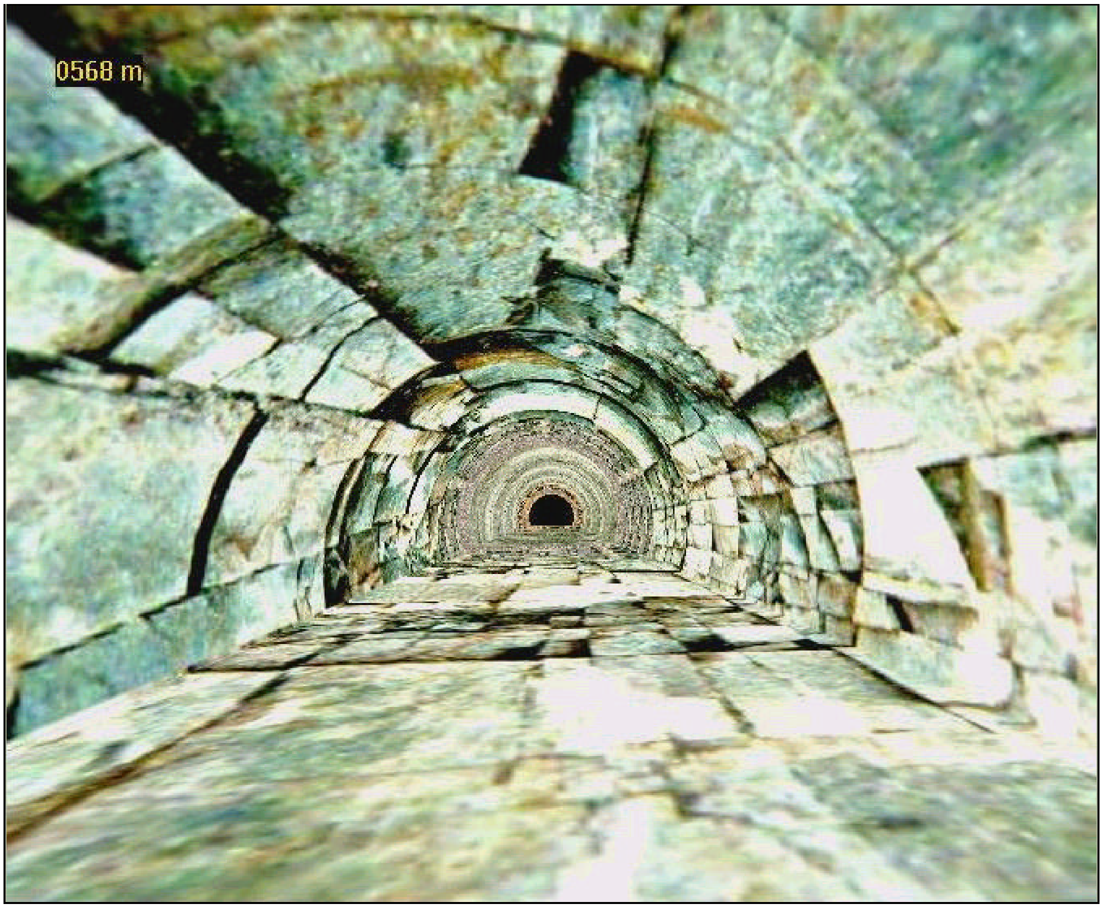

figure 21 - snapshot from TunVis video sequence

List of figures

References

[Bone97] De Bonet, **„Multiresolution Sampling Procedure for Analysis and Synthesis of Texture Images“**, 1997, Computer Graphic Proceedings, Annual Conference Series, pp. 361-368

[Chri92] Christian, **„The Microsoft Guide to C++ Programming“**, 1992, Microsoft Press

[DaWN93] Davis, Woo, Neider, **„OpenGL Programming Guide: The Official Guide to Learning OpenGL, Release 1“**, 1993, Addison Wesley Publishing Company

[FoDa90] Foley, Van Dam, Feiner, Hughes, **„Computer Graphics: Principle and Practice“**, 2nd Edition 1990, Addison Wesley Publishing Company

[HeBe95] Heeger, Bergen, **„Pyramid-Based Texture Analysis/Synthesis“**, 1995, Computer Graphic Proceedings, Annual Conference Series, pp. 229-234

[KlRa91] Klein H. & Ramshorn Ch., **„3D-Computergraphik zum Visualisieren von geologischen Strukturen und simulierten geologischen Prozessen“**, 1991, Feiburger Geowissenschaftliche Beiträge, Universität Freiburg

[LiUl95] Lindenbeck Ch. & Ulmer H. **„Entwicklung und Anwendung von Computerprogrammen zur Visualisierung geologischer Strukturen und Prozesse“**, 1995, Anhang, Dissertation Geowiss. Fakultät, Universität Freiburg.

[Open92] OpenGL Architecture Review Board, **„OpenGL Reference Manual: The Official Reference Document for OpenGL, Release 1“**, 1992, Addison Wesley Publishing Company

[PeMi93] Pennebaker, Mitchell, **„JPEG Still Image Data Compression Standard“**, 1993, Van Nostrand Reinhold, pp. 638

[WaGr91] Wallace, Gregory K. **„The JPEG Still Picture Compression Standard“**, Communications of the ACM, April 1991 (vol. 34 no. 4), pp. 30-44

[Wall86] Wallbrecher**, „Gefügestatistische und Analytische Arbeitsweisen in der Geologie“**, 1986, Enke Verlag

[Watt93] Watt A., **„3D Computer Graphics“**, 1993, 2nd Edition, Addison Wesley Publishing Company