Scientific Visualisation 3 1999
Project Report

Kevin Pulo
kev@zip.com.au
Student ID: 9719113

November 9, 1999

1 Introduction

1.1 Dataset background

The dataset used in this visualisation project is of Australian meteorological data. It was obtained from the Australian Bureau of Meteorology with the help of Wei Wang from Sydney Vislab. The data has been generated from a model developed at the Bureau of Meteorology, however, it is likely that the origin of at least some of the data is based in measurements of the real system.

1.2 Dataset particulars

The dataset contains data from around Australia, excluding the westernmost edge of Western Australia, the northernmost tips of the Northern Territory and Queensland, and the southern half of Tasmania. The dimensions are 82 in the x direction (West–East), 60 in the y direction (South–North) and 16 in the z direction (vertical slice planes). The dataset is spatially irregular, that is, the data points are at arbitrary positions in 3D space[1], rather than at regular intervals as is the case with regular and rectilinear data[2]. In addition, a separate file gives topographic information, that is, the surface's elevation above sea-level at each (x, y) point in the dataset.

The dataset covers the 47 hour period between 00:00 UTC[3] 29 August, 1996 (10:00 AEST[4] 29 August, 1996) and 23:00 UTC 30 August, 1996 (09:00 AEST 31 August, 1996), at 1 hour intervals. Thus the dataset can be thought of as being 4 dimensional, 3 in space and 1 in time.

At each point in the 4D dataset, three variables are available, being the wind velocity (a 3-vector in x, y and z in km/hr), the temperature (in degrees Kelvin) and the relative humidity (that is, the humidity relative to the dew-point absolute humidity).

The dataset suffers from a peculiarity, however, and that is a distinct lack of data for the z-component of the wind velocity. Whether this is due to the original wind velocity measurements lacking z-components or the model's possible inability to deal with z-components of wind velocity is unknown. For the purposes of the visualisations in this project, this meant that the winds appear to be "trapped" in

[1] As specified in a file of the dataset, `xyz.dat`.

[2] This can be clearly seen when performing a `Volume Bounds` on the dataset, as in Figure 3.

[3] Universal Coordinated Time, formerly Greenwich Mean Time (GMT), the time at the zero meridian.

[4] Australian Eastern Standard Time.

horizontal planes. This is unfortunate, but doesn't detract a great deal from the visualisations themselves.

1.3 Dataset investigations

Two main ideas shaped the investigation of this dataset — how the variables change as the altitude is increased, and the possible relationships between one or more variables.

The motivation for the first is that analysing the variables on the ground, while directly relevant to everyday experience, may not provide the whole story. As such, it is hoped that investigating the variables at higher altitudes may provide extra insights into the weather, specifically that at ground level.

The motivation for the second is that all three of the variables in the dataset are ultimately determined from one physical entity: the gaseous atmosphere. As such, it would be expected that meteorological phenomenon, for example, low pressure zones or cold fronts, would be manifest in more than one of the variables. As such, visualising more than one variable simultaneously might make these phenomenon easier to see and interpret.

2 Visualisations Performed

Four distinct types of visualisations of the dataset were produced, each examining a different variable or set of variables in a different way. All of the visualisations produced animations, some are animations of the system as it progresses with time, and the others are animations which use time as an extra dimension in the visualisation, that is, are animations of some phenomena at a fixed point in time.

2.1 Surface wind and clouds

The first visualisation aimed to be an introductory look at the dataset. It looks at the wind velocity at ground level and the cloud distribution through the dataset as time progresses through the 47 hours in the dataset. This gives a good indication of what to expect in the humidity and ground-level winds of the dataset's time period.

The visualisation involves a few simple stages. The first is a topographic map which provides a reference and context for the rest of the visualisation. This is achieved by generating a mesh showing the terrain, coloured by altitude, and a contour at sea-level to indicate the coast.

The next stage is to add the wind velocity visualisation. This was a straightforward hedgehog vector visualisation, in the lowest horizontal plane possible. The hedgehogs are simple and easy to interpret, showing direction and magnitude.

The final stage is to display the relative humidity data by creating an isosurface at relative humidity of 1.0, the dew-point. This is then made slightly transparent so that the clouds don't obscure any of the wind information. The dew-point is the most useful point, as it is approximately the point where clouds begin to form in reality. Thus, an isosurface at the dew-point gives the approximate surface where clouds would form in reality, making it a good choice.

2.2 Surface temperature and wind

The second visualisation aimed to extend the first visualisation while still remaining simple and uncomplicated. This visualisation is purely 2 dimensional, showing the wind velocity and temperature at ground level, and how these change over the time period of the dataset.

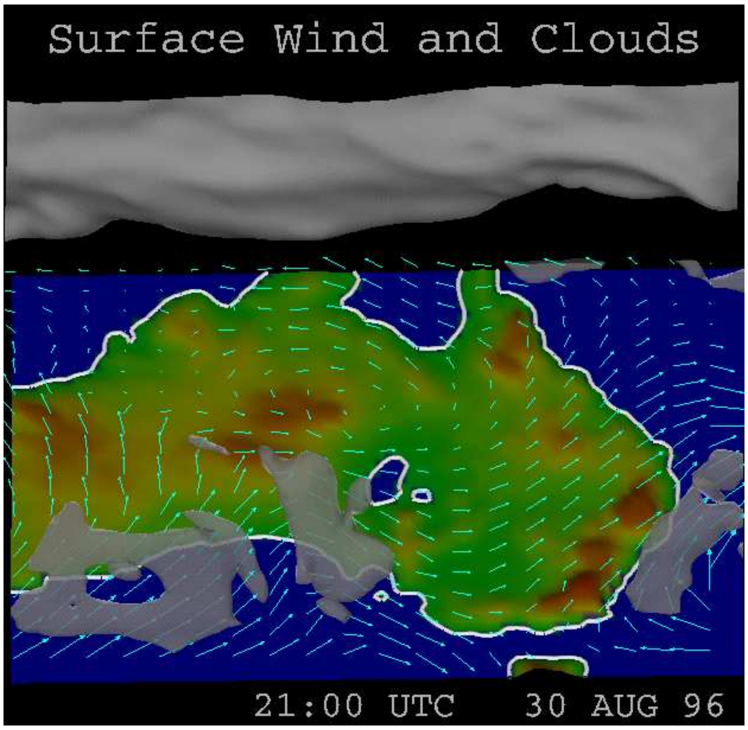


Figure 1: A sample frame from the animation of the first visualisation, which shows the wind velocity at ground level, and the approximate appearance of clouds throughout the dataset.

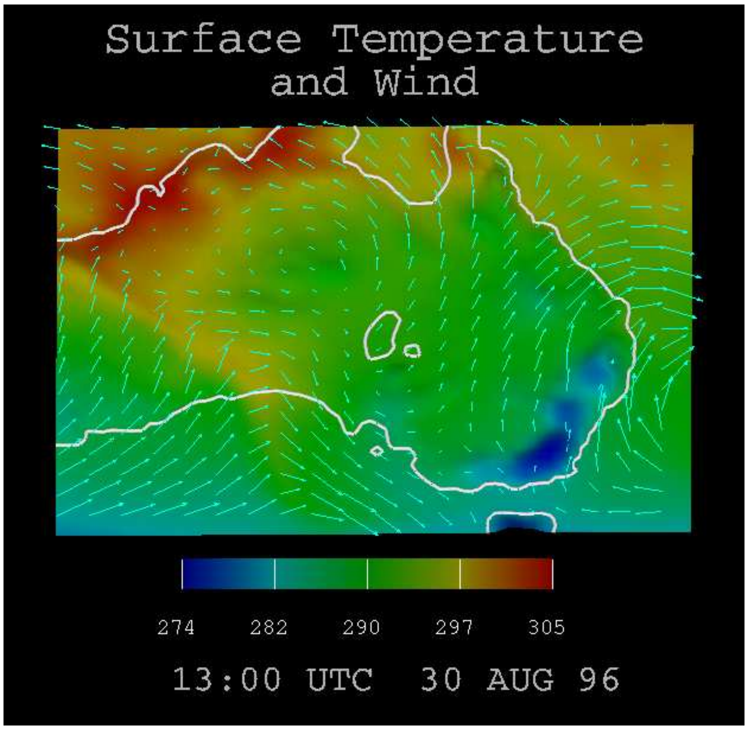


Figure 2: A sample frame from the animation of the second visualisation, which shows the wind velocity and temperature at ground level.

The wind velocity is visualised in the same way as in the first visualisation, described in Section 2.1. The temperature is visualised by colouring the terrain mesh by the temperature of the points in the mesh. This is an easily interpreted way to view temperature data, and also allows the temperature to be directly compared to its geographic location. For example, the blue in the lower right corner of Australia is not surprising when it's geographic location, the Great Dividing Range, is considered.

Another purpose of these first two visualisations is to look for possible correlations between variables, the first between humidity and surface wind velocity, and the second between surface temperature and wind velocity. The most notable example of how successful this technique of combining variables can be is a front progressing from the south western corner of Australia across to the center of South Australia. When analysing the wind vectors alone, this front is quite hard to see. However, when combined with the temperature visualisation, the front becomes very clearly visible, and the unique pattern of wind vectors can be seen above the yellow streak of temperature.

Further, if we then refer back to the first visualisation, clouds can be seen forming over and following the front as it progresses. This kind of clearly visible information and interpretations is the value and reason for combining several variables when visualising.

2.3 Surface temperature and wind speeds

The third visualisation aimed to take a different tack, by examining the wind at higher altitudes. It used the simpler streamlines rather than hedgehogs, as the overall direction of the wind (west to east) is now known and the emphasis is on the strength and shape of the winds at different altitudes, at a single point in time.

The visualisation takes streamlines in successive horizontal slice planes, starting at ground level and working their way up. This gives the viewer the impression of how the winds would vary if, for example, they were observing them[5] in a airplane which was rising.

The most notable features are that the winds at higher altitudes are stronger and more concentrated, and the jetstream is clearly visible.

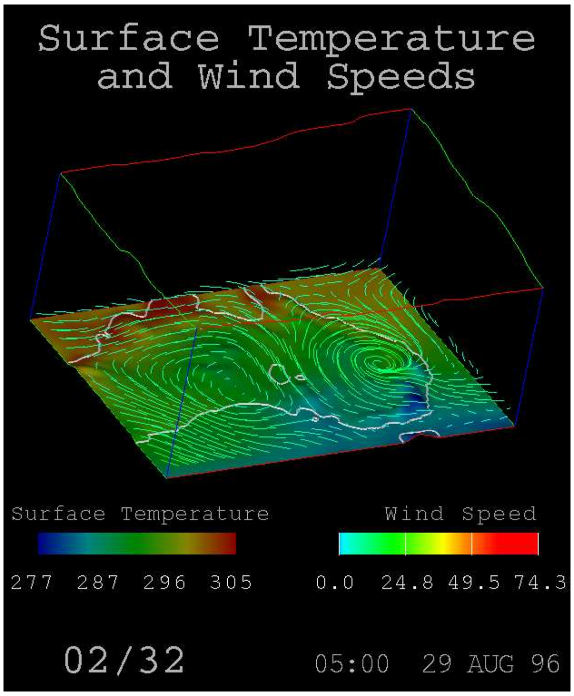
The surface temperature is also visualised in the same manner as in previous visualisations. This was done in the hope that some correlation between the high altitude winds and the surface temperature. None appear obvious, but a more trained eye may be able to distinguish some.

In addition, 32 slice planes were taken in the animation, yet the data contains only 16 horizontal slice planes in the z direction. 16 slice planes in the animation were not enough to make it smoothly flowing, so to compensate for this the streamlines were interpolated between slice planes using a Runge-Kutta method. This is a good approximation technique and gives a smooth animation.

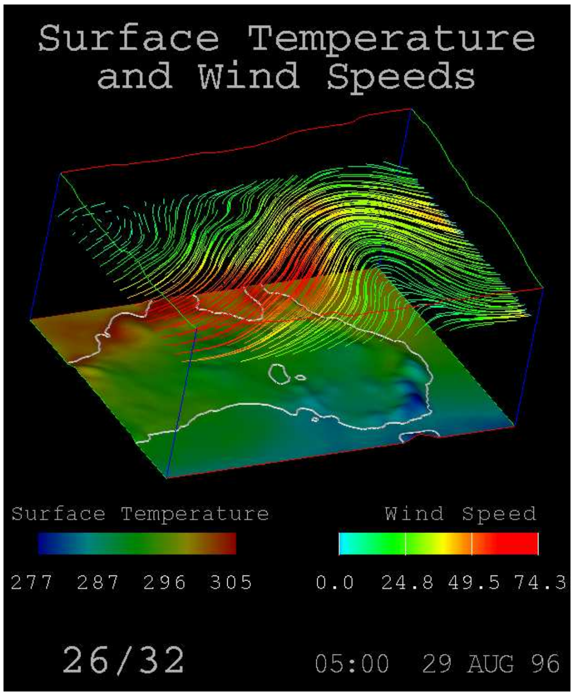
2.4 Air temperature gradient

The fourth visualisation aimed to further investigate the temperature, this time at higher altitudes. It does this by using particle advection, by injecting massless particles at ground level and allowing them to "float" upwards in the temperature gradient field. This gives the effect of the particles rising as if they were only affected by the temperature and convection currents, without any regard for the wind or other effects. Further, the particles and their streamline-style paths are coloured according their temperature. This makes the overall view very natural —

[5] In the sense of being able to see the streamlines as in the visualisation. Naturally, this isn't possible in reality, and a major driving force of scientific visualisation.



(a) Low altitude winds



(b) High altitude winds

Figure 3: Two sample frames from the animation of the third visualisation, which shows the temperature at ground level and wind velocity throughout the dataset.

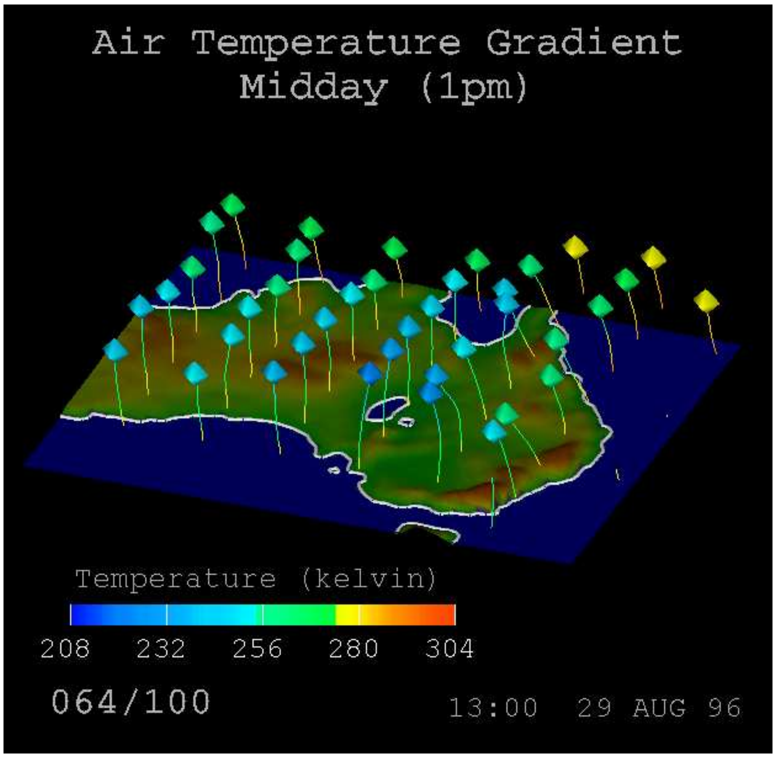


Figure 4: A sample frame from the animation of the fourth visualisation, which shows how the temperature evolves as the altitude increases.

the particles are coloured by temperature, and the paths they take the direction of the changing temperature.

The colours are selected such that that temperatures above $\sim 0°$C are orange and yellow, temperatures between $\sim 50°$C and $\sim 0°$C are green, and temperatures below $\sim 50°$C are blue in colour. This means that the particles "switch" colours when they pass these thresholds, giving good visual indication of what stage of cooling they are at.

Three separate points in time are considered, and an animation is generated for each. The three times are Midday (1pm AEST), Evening (7pm AEST) and Early Morning (4am AEST). It is hoped that these visualisations will exhibit differences since they are from different temperature conditions — the midday case should be the warmest (at least at ground level), followed by the evening and then the early morning. Unfortunately, this distinction is not as clear as was originally hoped. The midday case was warmer overall, but there is very little difference between the evening and early morning cases. A preliminary attribution for this is the possibility that the results for the evening and early morning cases are skewed because the sun is still shining in the upper atmosphere.

Another noticable feature is how many of the particles have a general drift to the west. This is attributed to the fact that the earth is spinning underneath, and the temperature is the temperature of the air at these points — hence, this may drift slightly with respect to the earth below.

3 Software Used

The main piece of software used in performing the visualisations was AVS 5, and the networks used are discussed and presented in detail in Section 3.1. All the animations were performed in AVS 5, which generated individual frame files for each animation. These frame files were then composited into SGI movie files and Quicktime movie files using the SGI `makemovie` utility. The image files created by AVS 5 were converted into the intermediate SGI RGB file format using the `mica` batch image conversion utility.

The specific settings found to be best for `makemovie` were either SGI movies using JPEG compression, or Quicktime movies using Compact Video (also known as Cinepak) compression. These movies have the best compression ratios (that is, produce the smallest files) whilst preserving good image quality and frame rates. The Quicktime Cinepak movies, in particular, are suitable for publication on the World Wide Web, as Quicktime Cinepak is a common format with wide support.

The time-based animations, presented in Sections 2.1 and 2.2, have frame rates of 2 frames per second. The animations with time as an additional visualisation dimension, presented in Sections 2.3 and 2.4, have frame rates of 5 frames per second. These rates were chosen as they show the animations with enough speed to appear smooth, yet slow enough to examine detail and thus are not too fast. However, fast, overview animations are useful for seeing the "bigger picture" of the animation, and frame rates of 10 frames per second for the time-based animations and 20 frames per second for the remaining animations have proved valuable.

The SGI utility `moviemaker` was found to be grossly too unstable for practical use, and so as a result all titles and other context was added in AVS 5 at the image generation stage, rather than at the movie creation stage.

In addition, small `awk` scripts were written to generate the many repetitive AVS field files required, as detailed in Section 3.1.

3.1 AVS networks

Several quite sophisticated AVS 5 networks were created to perform these visualisations. These networks will now be discussed in detail.

3.1.1 Surface wind and clouds

Figures 5 and 6 are the AVS 5 networks used to create the first visualisation, as discussed in Section 2.1.

The first section of the network to consider is the `topog` macro module, shown in Figure 6. This module reads in the topographic data and does two things with this data. It converts the topographic field into a mesh using the given colourmap, giving a mesh which shows the terrain of Australia, and also takes a contour of the topographic data at the 0.0 value, that is, sea-level, giving the outline of Australia[6]. The mesh's z scale is set to 1 to ensure that the terrain is on the same scale as the rest of the data.

The rest of the network shown in Figure 5 is in several large areas.

The right hand side simply provides context in the output in the form of labels, as well as housing the `topog` macro module.

The `extract scalar` and `isosurface` modules in the middle extract the relative humidity data from the main field, and then display an isosurface of it at 1.0, the dew-point — these appear as clouds in the output[7], as shown in Figure 1.

[6]The contour is passed through the `tube` module so that it may be easily seen.

[7]The clouds also have their opacity decreased to about 70%, to allow them to be partially transparent, and not obscure the surface wind vectors. This is achieved using controls in the AVS

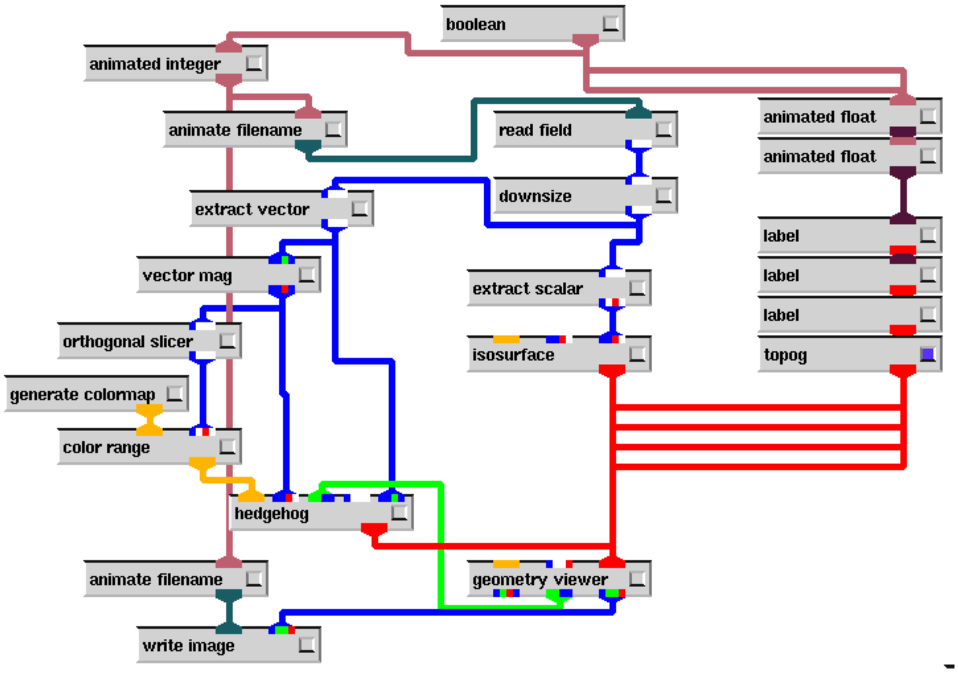


Figure 5: The AVS 5 network used to create the first visualisation, as discussed in Section 2.1.

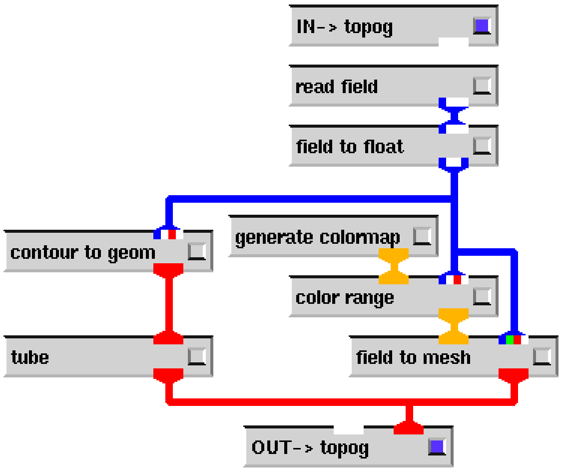


Figure 6: The `topog` macro module, as used in the first (and fourth) visualisation to provide the surface of Australia coloured by elevation, and the coastline of Australia.

The modules on the left between the `extract vector` and `hedgehog` modules provide the wind vector visualisation, in the form of hedgehog arrows. The arrows are coloured according to their magnitude (and the given colourmap)[8], this is the purpose of the `vector mag` module. The `orthogonal slicer` module is used to adjust the colourmap to the values of the wind in the lowest slice plane, which is where the hedgehog sample points are located. One problem which arose here is the inability to properly "thicken" the hedgehog lines, which will be discussed in Section 4.

Finally, there are several animated integers, floats and filenames to support the reading of the field files and the writing of the image files.

The field files for these networks are constructed in the form `mix.xxx.fld`, where `xxx` indicates the timestep sequence, that is, from `mix.000.fld` to `mix.047.fld` for this dataset. The field files skip ahead in the data file to wherever their time sequence begins[9], and are loaded with the aid of an `animated filename` module. This method of reading time sequences is easy to construct and is quite useful. The field files were constructed with the aid of a simple awk script. They have several advantages over reading the entire 4D data at once. They don't require as much memory, as they only require AVS to store a single timestep of data at a time, and they are easier to work with when only dealing with a single point in time. However, they have the disadvantage that they are slower when performing an animation, as the early timesteps of the data file must be re-read many times when it is being skipped to read the later timesteps.

3.1.2 Surface temperature and wind

Figures 7 and 8 are the AVS 5 networks used to create the second visualisation, as discussed in Section 2.2.

In this network, the `topog` macro module, shown in Figure 8 has been modified from the original `topog` module shown in Figure 6. Specifically, this `topog` module simply provides the contour for the coastline of Australia, without any terrain. This is because this network requires the terrain to be coloured by the (surface) temperature, and so is provided in the main network.

The main network shown in Figure 7 is quite similar in layout to that in Figure 5, however, there are differences in the central region.

The left and right hand edges, as well as the top and bottom regions are essentially the same, performing the same tasks. The main difference is in the central region. Here, the temperature is once again extracted with an `extract scalar` module, and the lowest slice (in the z direction) is taken with an `orthogonal slicer` module. The appropriate colour range for these temperatures is selected with the `color range` and `generate colormap` modules, and then the temperature field for the surface and this colourmap are fed into the `field to mesh` module. The `field to mesh` module has its z scale parameter set to 0, which means that the mesh constructed has the geometry of the underlying field. This field is irregular, and has the geometry of the terrain of Australia, and thus the `field to mesh` module outputs a mesh of the terrain of Australia coloured by the surface temperature.

5 Geometry Viewer.

[8]However, simply because the network contains the modules to colour the hedgehog arrows does not mean they will be coloured. Refer to Section 4 for more details on AVS nuances.

[9]The number of bytes to skip for a given timestep t is given by the formula

$$t \times xdim \times ydim \times zdim \times numvar \times bytespervar$$

where $xdim$, $ydim$, $zdim$ are the x, y, and z dimensions of the dataset, $numvar$ is the number of variables stored at each point (eg. 3 for the wind velocity, 1 for the relative humidity, etc), and $bytespervar$ is the number of bytes per variable, typically 4 for 32 bit values.

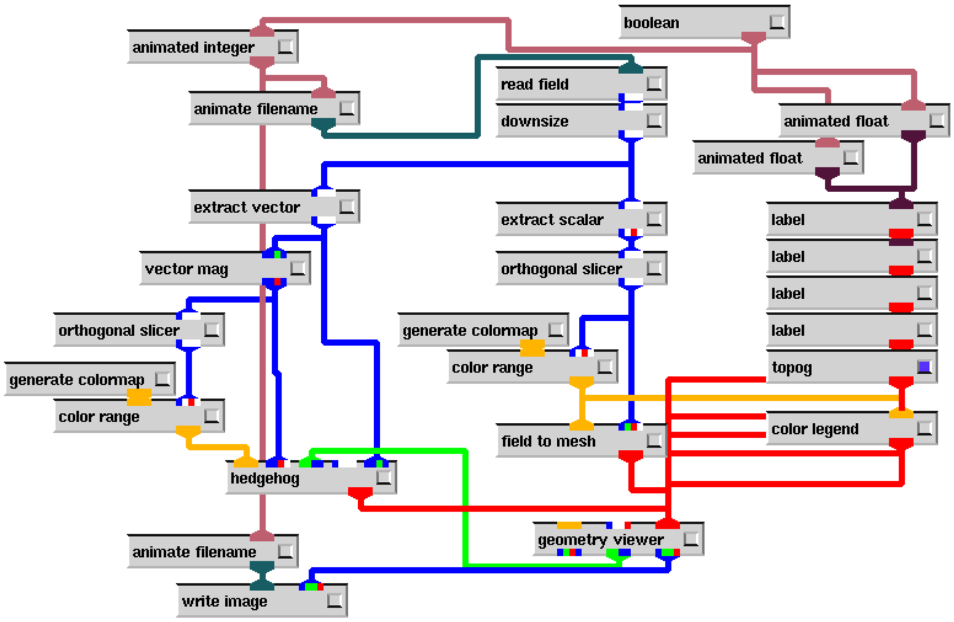


Figure 7: The AVS 5 network used to create the second visualisation, as discussed in Section 2.2.

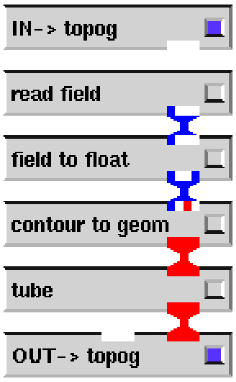


Figure 8: The `topog` macro module, as used in the second (and third) visualisation to provide just the coastline of Australia.

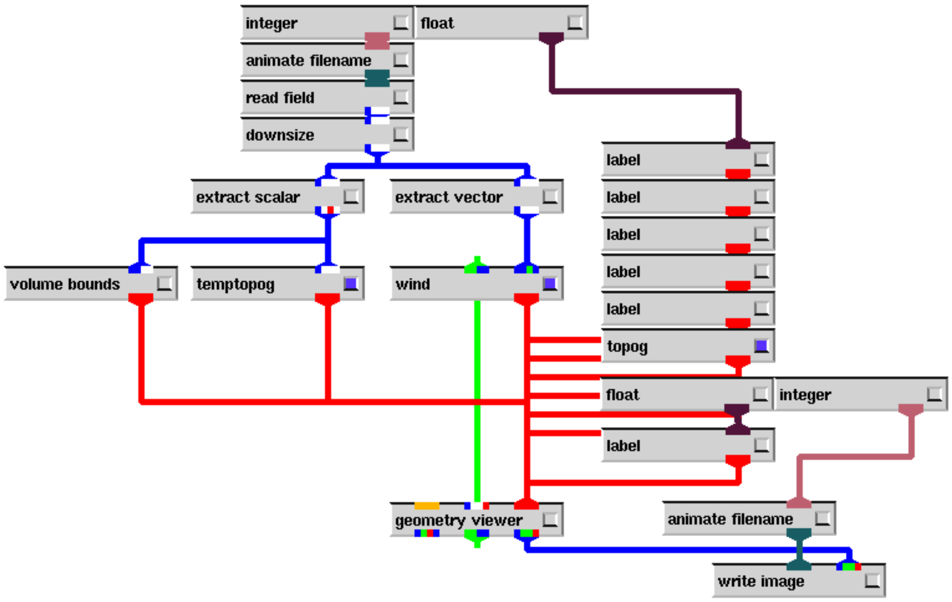


Figure 9: The AVS 5 network used to create the third visualisation, as discussed in Section 2.3.

3.1.3 Surface temperature and wind speeds

Figures 9 to 11 are the AVS 5 networks used to create the third visualisation, as discussed in Section 2.3.

The `topog` macro module used in Figure 9 is identical to that used in the second visualisation, shown in Figure 8. The `temptopog` macro module, shown in Figure 10, simply encapsulates the task of generating the terrain of Australia coloured according to surface temperature which was performed in the second visualisation, shown in its main network in Figure 7.

The `wind` macro module, as shown in Figure 11, visualises the wind velocity[10]

[10]Which is extracted from the main field with the `extract vector` module in the main network, shown in Figure 9.

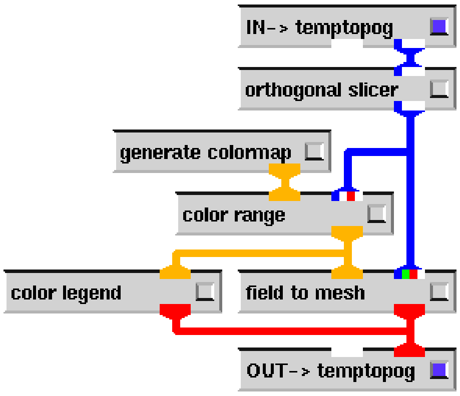


Figure 10: The `temptopog` macro module, as used in the third visualisation to provide the terrain of Australia coloured according to the surface temperature.

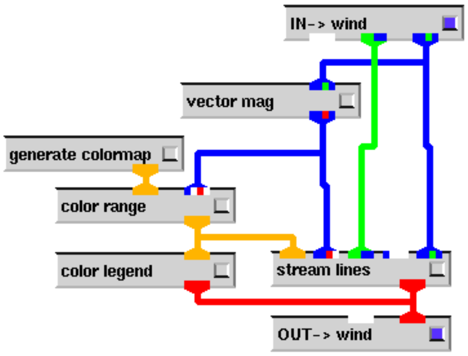


Figure 11: The `wind` macro module, as used in the third visualisation to provide the visualisation of the wind in a horizontal slice place of the dataset. The visualisation is a set of streamlines coloured by the speed of the wind at each point.

as a set of streamlines coloured according to the magnitude of the wind velocity (ie. the wind speed) at each point.

The `volume bounds` module provides context for where the current horizontal slice plane "fits" within the whole dataset. Notice in Figure 3 that the volume bounds are not a rectangular prism, due to the irregularity of the underlying dataset, as discussed in Section 1.2.

There are no animated integers or animated floats in the main network[11]. This is because there is no easy way to select the plane used by the `streamlines` module. Thus the animation for this network was performed by hand, using the Geometry Viewer's `Transform Object` control to move the streamlines plane up by 0.5 units each frame. The integer controlling the output filename and the float controlling the frame number display label module were then updated and the process repeated for each frame.

3.1.4 Air temperature gradient

Figures 12 and 13 are the AVS 5 networks used to create the fourth visualisation, as discussed in Section 2.4.

The `topog` macro module used in Figure 12 is identical to the `topog` module used in the first visualisation, shown in Figure 6, used to generate the terrain of Australia coloured by elevation.

The main network for this visualisation, shown in Figure 12, has delegated most of the real work to the `advect` macro module. The main network in this case simply reads in a single field file (at a particular timestep, specified by the `integer` at the top left), displays the time of this timestep (using the `float` at the top right) and various other context information (the `labels` and `topog` along the right), and outputs the frames to image files using the `write image` module. The field file read has the temperature extracted from it with the `extract scalar` module, which is then fed to the `advect` module.

The `advect` macro module, shown in Figure 13, accepts as input a 3D scalar field (the temperature) and an `upstream_transform` struct, used to tie the Geometry Viewer to the `particle advector` module within the `advect` module. The `advect` module outputs the geometry of the scene and an `integer` indicating the current frame number, suitable for using with an `animated filename` when saving frame image files.

[11]There are animated filenames, but these are simply for convenience, to be able to specify simply a timestep number or frame number, rather than editing a filename string.

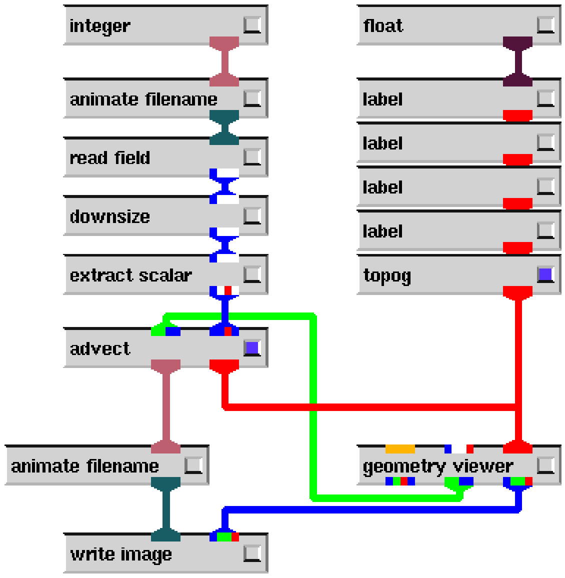


Figure 12: The AVS 5 network used to create the fourth visualisation, as discussed in Section 2.4.

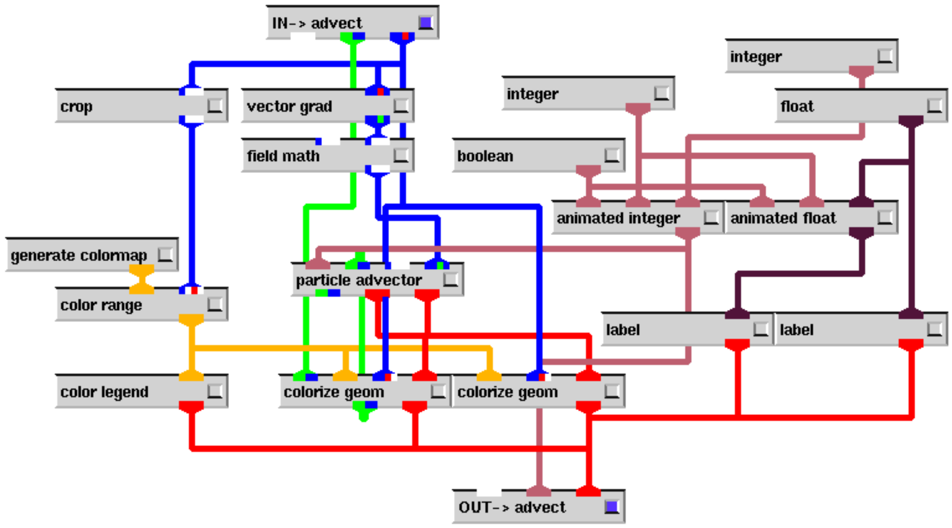


Figure 13: The `advect` macro module, as used in the fourth visualisation to provide the visualisation of the temperature gradients and temperatures along these gradients in the dataset. The visualisation itself uses the method of particle advection.

The `advect` module creates its colourmap by cropping out the highest horizontal slice plane in the z direction, which has some erroneous temperature values in it, and then colouring accordingly. The colourmap has been created so that temperatures above $\sim 0°C$ are orange and yellow, temperatures between $\sim 50°C$ and $\sim 0°C$ are green, and temperatures below $\sim 50°C$ are blue in colour. The temperature's gradient is computed with the `vector grad` module, and since it normally flows from low values (ie. cold) to high values (ie. warm) it is then passed through the `field math` module to multiply all the field's values by -1, making the field flow from warm values to cold values, as desired.

This gradient is then passed into the `particle advector` module, which has been synchronised to the `animated integer` on the right. This means that the `particle advector` module will only advance the positions of the particles when the value of the `animated integer` changes. This allows the individual frames of the particle advection to be saved as separate image files.

The `particle advector` module has input ports to support colouring the output geometries, however, it always colours the output geometries according to their speed through the vector field. This is undesirable, what is wanted is to colour the output geometries according to their actual temperatures. This is achieved through the use of the `colorize geom` modules, which colour their input geometries according to the scalar field they are given — the actual temperature at each point, which works quite well. The `particle advector` module outputs two geometries, the particles themselves and their streamline "tails". Both are colourised in exactly the same way.

4 Issues Encountered

During the construction of these visualisations, many issues and limitations with AVS 5 were encountered. Most could be resolved satisfactorily, but some were simply had to be endured. The most notable will now be discussed.

4.1 Inserting modules between hedgehogs and the geometry viewer

AVS 5's `tube` module is a very useful module. It converts lines in a geometry into eight-sided cylinders of a given radius, and is thus excellent for "thickening" lines to be more clearly visible.

However, it has a major drawback. When it is inserted into a network between a `hedgehog` or `streamlines` module and the `geometry viewer` module, it breaks the normally invisible, automatic `upstream_transform` link between these modules. This link is used to tell the `hedgehog` or `streamlines` module that its sample points have changed, and the hedgehogs or streamlines must be recalculated.

Worst, though, is that reconnecting the `upstream_transform` link does not restore things to a sane condition as it should. To work properly, the output of the `hedgehog` or `streamlines` module must be directly connected to the `geometry viewer`, and the `geometry viewer`'s `upstream_transform` output must be directly connected to the `hedgehog` or `streamlines` module. If this is not the case, it is impossible to change the `hedgehog` or `streamlines` module's set of sample points from their default, which is a plane in the middle of the dataset. For most of the visualisations in this project, the hedgehogs or streamlines were desired at other planes, such as ground level.

Unfortunately, this is a hard limitation of AVS 5, and there is no real solution other than to live with the limitation. The best solution, however, would be to modify the `tube` module (in source code) to allow it to properly propagate the `upstream_transform` module upwards in the network.

The `particle advector` module also suffers from similar problems, but is more well behaved when the `upstream_transform` links are reconnected.

4.2 Colouring hedgehogs

As seen in the wind vector hedgehogs in the first and second visualisations, sometimes the AVS 5 `hedgehog` module will, for unknown reasons, not properly colour the vector field, despite having all the appropriate parts of the network required for this. This problem was unresolved, as the length of the arrows show their magnitude, and while it would have been nice to have the arrows coloured according to their magnitude, the length is (barely) good enough.

4.3 Constant colour scales over time

In AVS 5, it is sometimes particularly hard to get colour scales which are remain constant over an animation. This is because the colourmap is generated for each timestep of the dataset individually, as a direct result of the `color scale` module and the way different field files are used for each timestep. The result is a colour scale which changes for each timestep in an animation, which, if severe, can seriously impede the visualisation any scientific meaning it may have.

The second visualisation suffered from this problem, fortunately, though, the problem was minor. The colour scale never changed by more than about 5% overall, and was always in the same regime of values. However, the correct way to solve this problem is to construct a colour map which is valid for all the timesteps of the animation, and then feed this directly into the appropriate module without the use of the `color range` module (though the `color range` module may be useful in obtaining the colourmap).

4.4 Automatically altering sample plane of hedgehog or streamlines module

There is no easy way to automatically (that is, using controls in the network such as `animated integer`) alter the sample plane used by the `hedgehog` or `streamlines` modules. This is desirable for an animation of hedgehogs or streamlines through many planes in a 3D dataset, as in the third visualisation.

Sadly, other modules such as `orthogonal slicer` and `crop` have their limitations as well, making them unsuitable for preprocessing input of the `hedgehog` or `streamlines` modules. The `orthogonal slicer`'s output is not compatible with `hedgehog` or `streamlines` (different dimensionality) and causes an error when used, and `crop` merely throws away data, rather than limiting where the `hedgehog` or `streamlines` modules may place their sample plane.

The solution for this was to manually generate each frame, by using the Geometry Viewer's `Transform Object` feature to move the sample plane, and then using the usual AVS 5 modules to save the frame.

4.5 Video related issues

From its creation, the animations of the project were not intended for output on video. Thus, they are less attractive when displayed on video, because the video considerations were not taken into account.

The thin lines of the hedgehogs, streamlines and particle advection tails are the biggest problem. They are too thin for video to properly display, and further, colour bleeding of the background makes the thin lines in the foreground even harder to see. The ability to use the AVS 5 `tube` module on this output would have helped in this case.

Further, some of the colourmaps had reds which were too saturated for video. When displayed on video, these colours couldn't be displayed and were shown in black, rather than red. The modification of the colourmaps, of the passing of the output through the AVS 5 `prepare video` module would have avoided this problem.

Lastly, none of the animations were created in PAL or NTSC resolution, indeed, the third visualisation is not even approximately of the correct aspect ratio. In addition, most of the animations extend outside the video safe region of approximately the central 80% of the screen. This is simply because the animations were never intended for video, they could easily be letterboxed to fit, or regenerated with better layouts and the correct resolution.

5 Acknowledgements

The author would like to express sincere thanks to Dr. Wei Wang from Sydney Vislab for her help and assistance in obtaining the dataset and the visualisations performed on it. She assisted in the reading of the data into AVS and many technical issues which occured in AVS 5, as well as helping find the exact particulars of the dataset when no supporting documentation was available.

In addition, Daniel Mitchell, also from Sydney Vislab, assisted with many technical issues in AVS 5 and helped bring several ideas to fruition.

This work was produced with facilities provided by the Sydney Regional Scientific Visualisation Laboratory, 1999, as part of the Scientific Visualisation 3 third-year Physics course (PHYS 3303).